# Subscription Model Based Business

## PCI Compliance

PCI DSS 🡪 Payment Card Industry Data Security Standard. We do not want to store the user’s credit card information

Requirements

* Build and maintain a secure network
* Protect Cardholder Data
* Maintain a Vulnerable Management Program
* Implement Strong Access Control Measures
* Regularly Monitor and Test Networks
* Maintain an Information Security Policy

The technical difference between dedicated and collocated hosting is that the host typically owns a dedicated server whereas we typically own a collocated one.

### Payment System

Payment Processors 🡪 It is a delayed payment system that normally goes through a third-party site.

Payment Gateway 🡪 It is a real-time payment system that can be directly integrated into our own site.

Thumb Rule for Database

* Public: SELECT
* Customer: SELECT, INSERT, UPDATE
* Admin: SELECT, INSERT, UPDATE, DELETE

## Secure Transactions

Secure Sockets Layer (SSL) defines a protocol protecting data transmitted over public networks. SSL is an absolute must for e-commerce site. SSL provides encryption, decryption of data passed back and forth between the server and the client.

## Types of Attacks

### Protecting the User

**Cross-Site-Scripting**

It can be injecting JavaScript into the components of user inputs and therefore, steal the user cookies or execute code to steal the personal data. This can be easily avoided using PHP function **strip\_tags()**, this removes any HTML, JavaScript or PHP from any string

## Protecting the Site

**Denial of Service**

It attempts to connect to our simultaneously from all around the world and causing the server to be busy

**SQL Injections**

Tricking MySQL with comments. A double hyphen (--) is used in SQL to comment out the remainder of a line of SQL code.

We can use the PHP and SQL function combined function **mysqli\_real\_escape\_strings()**, it **e**scapes special characters in a string for use in an SQL statement, taking into account the current charset of the connection.

We should typecast all values that should be numeric to force them to be numbers.

**$id = (int)$\_GET[‘id]**

**Remote File Inclusion (RFI)**

The hacker attempts to get a site to include a file found on another server, PHP, when it calls fopen(), require(), include(), **LFI (Local File Inclusion)** attack is similar to RFI allows sensitive data like password to be read and displayed🡪 this can be prevented if the server do not use system() and exec() PHP functions.

**Cross-Site Request Forgeries**

A cross-site request forgery(CSRF) is an attack that attempts to cause a victim to unknowingly send arbitrary HTTP requests, usually to URLs requiring privileged access and using the existing session of the victim to determine access. We can prevent this by creating unique tokens

**$token = md5(uniqid(rand(),true))**

**$\_SERVER['HTTP\_USER\_AGENT']**

### Summary

|  |  |
| --- | --- |
| **Functions** | **Usage Reason** |
| **strip\_tags()** | Removes any HTML, JavaScript or PHP from any string |
| **mysqli\_real\_escape\_string()** | Escapes special characters from the string |
| **(int)$\_GET[‘id’]** | Typecast the numeric values |
| **$token = md5(uniqid(rand(), true))**  **$\_SERVER['HTTP\_USER\_AGENT']** | Halts unwanted arbitrary HTTP requests and prevents use of existing session |

1. <? php session\_start();
2. $token = md5(uniqid(rand(), true));
3. $agent = $\_SERVER['HTTP\_USER\_AGENT'];
4. $\_SESSION['token'] = $token;
5. $\_SESSION['agent'] = $agent;
6. setcookie('token', $token, time() + (86400 \* 30), "/");
7. setcookie('agent', $agent, time() + (86400 \* 30), "/");
8. echo "<!DOCTYPE html>";
9. echo "  <html lang = 'en'>";
10. echo "      <head>";
11. echo "          <title> Form </title>";
12. echo "          <meta charset = 'UTF-8'/>";
13. echo "          <meta name = 'viewport' content = 'width=device-width; initial-scale=1'/>";
14. echo "      </head>";
15. echo "      <body>";
16. echo "          <form method = 'post' action = 'handleForm.php'>";
17. echo "              <input type = 'hidden' name = 'token' value = '".$token."'/>";
18. echo "              <input type = 'hidden' name = 'agent' value = '".$agent."'/>";
19. echo "              <label for = 'data'> Data </label><br/>";
20. echo "              <input type = 'text' name = 'data' /><br/><br/>";
21. echo "              <input type = 'submit' name = 'submit' value = 'Confirm'/>";
22. echo "          </form>";
23. echo "      </body>";
24. echo "  </html>"; ?>
25. <? php session\_start();
26. define('DB\_USER', 'root');
27. define('DB\_PASS', '');
28. define('DB\_HOST', 'localhost');
29. define('DB\_NAME', 'ecommerce1');
30. if ($\_SERVER['REQUEST\_METHOD'] == 'POST'): if (isset($\_SESSION['token']) && isset($\_SESSION['agent'])): if (isset($\_COOKIE['token']) && isset($\_COOKIE['agent'])): if ($\_POST['submit']): if ($\_POST['token'] == $\_SESSION['token']): if ($\_POST['agent'] == $\_SESSION['agent']): if ($\_POST['token'] == $\_COOKIE['token']): if ($\_POST['agent'] == $\_COOKIE['agent']): $connection = mysqli\_connect(DB\_HOST, DB\_USER, DB\_PASS, DB\_NAME);
31. $strip = $\_POST['data'];
32. $escape = mysqli\_real\_escape\_string($connection, $strip);
33. echo "You entered: ".$escape.
34. "<br/>";
35. echo "Your token: ".$\_POST['token'].
36. "<br/>";
37. echo "Your agent: ".$\_POST['agent'].
38. "<br/>";
39. echo "Current agent: ".$\_SERVER['HTTP\_USER\_AGENT'].
40. "<br/>";
41. else :echo "Cookie Agent do not match";
42. endif;
43. else :echo "Cookie Token do not match";
44. endif;
45. else :echo "Session Agent do not match";
46. endif;
47. else :echo "Session Token do not match";
48. endif;
49. else :echo "Form did not get submitted";
50. endif;
51. else :echo "Token and Agent Cookie is not set";
52. endif;
53. else :echo "Token and Agent Session is not set!";
54. endif;
55. else :echo "It is not a post request!";
56. endif; ?>

## Database Design

The **categories**, **pages**, and **pdfs** tables 🡪 **Product Side**

**categories** table 🡪 Lists the categories into which the HTML content will be organized.

**pages** table 🡪 stores the actual HTML content 🡪 **title, description and content**

**Pages**

**title 🡪** It will be used as a **link to each page** and will also be used as the **browser’s title**

**description 🡪** Describes the page content

**content** 🡪 Stores the actual html content

**PDF**

It lists the particulars for each PDF file the site has

**Users**

This table stores a minimum amount of information about the customers. Each customer can create a username, stores their email address, a password, first name and their last name.

Enum, It represents the list of options whether administrator or member. The date\_expires

column stores the data through which their account is active.

**Orders**

This table swill stroe every that goes through PayPal, associated with the ID. Three pieces of information will be recorded: the transaction\_id, payment status, payment amount

**Primary Key**

* There can only be one primary key in a table
* In some DBMS it cannot be NULL - e.g. MySQL adds NOT NULL
* Primary Key is a unique key identifier of the record

**Unique Key**

* Can be more than one unique key in one table
* Unique key can have NULL values
* It can be a candidate key
* Unique key can be NULL and may not be unique

Primary key makes the table row unique (i.e, there cannot be 2 rows with the exact same key). We can only have 1 primary key in a database table.

Unique key makes the table column in a table row unique (i.e., no 2 table rows may have the same exact value). You can have more than 1 unique key table column (unlike primary key which means only 1 table column in the table is unique).

`backtick` allows us to use ridiculously column names in tables. Ex: **bro,so,cool,man**

**CATEGORIES**

1. CREATE TABLE categories(id SMALLINT NOT NULL AUTO\_INCREMENT, category VARCHAR(30) NOT NULL, PRIMARY KEY(id), UNIQUE KEY category(category))

**ORDERS**

1. CREATE  TABLE  orders(id  INT  UNSIGNED  NOT  NULL  AUTO\_INCREMENT,  user\_id  INT  UNSIGNED  NOT  NULL,  transaction\_id  VARCHAR(30)  NOT  NULL,  payment\_status  VARCHAR(30)  NOT  NULL,  payment\_amont  DECIMAL(6.2)  UNSIGNED  NOT  NULL,  payment\_data\_time  TIMESTAMP  NOT  NULL  DEFAULT  CURRENT  TIMESTAMP  PRIMARY  KEY(id),  KEY  user\_id(user\_id))

**PAGES**

1. CREATE TABLE pages(id MEDIUMINT UNSIGNED NOT NULL AUTO\_INCREMENT, category\_id SMALLINT UNSIGNED NOT NULL, title VARCHAR(100) NOT NULL, description TINYTEXT NOT NULL, content LONGTEXT NOT NULL, date\_created TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP, PRIMARY KEY(id), KEY category\_id(category\_id), KEY creation\_date(date\_created))

**PDFS**

1. CREATE TABLE pdfs(id SMALLINT UNSIGNED NOT NULL AUTO\_INCREMENT, tmp\_name CHAR(40) NOT NULL, title VARCHAR(100) NOT NULL, description TINYTEXT NOT NULL, file\_name VARCHAR(40) NOT NULL, size MEDIUMINT UNSIGNED NOT NULL, date\_created TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP, PRIMARY KEY(id), UNIQUE KEY tmp\_name(tmp\_name), KEY dat\_created(date\_created))

**USERS**

1. CREATE TABLE users(id INT UNSIGNED NOT NULL AUTO\_INCREMENT, type ENUM('memeber, admin') NOT NULL, username VARCHAR(30) NOT NULL, email VARCHAR(80) NOT NULL, pas VARBINARY(32) NOT NULL, first\_name VARCHAR(20) NOT NULL, last\_name VARCHAR(40) NOT NULL, date\_expires DATE NOT NULL, date\_created TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP, date\_modified TIMESTAMP NOT NULL DEFAULT ‘0000-00-00 00:00:00’, PRIMARY KEY(id), UNIQUE KEY username(username),
2. UNIQUE KEY email(email)))

## Server Organization

🡪 Administration Files

🡪 CSS

🡪 Vendor

🡪 Images

🡪 Media

🡪 PHP includes

🡪 JavaScript

🡪 Test

### PHP Includes

**config.inc.php 🡪** It is a script that defines the sites’ general behavior and various constants.

**footer.php 🡪** It is the half of the HTML template

**form\_functions.inc.php 🡪** It defines a function used by every form

**header.php 🡪** It is the other half of the HTML template

**login.inc.php 🡪** It handles the login process

**login\_form.inc.php 🡪** It is the login form

**mysql.inc.php 🡪** Database connection

## Database Connection

**mysqli\_set\_charset($dbc, 'utf8')**

It indicates which character set should be used for communications between PHP and the database.

### PHP Closing Tag Omitting

* Functionality Loss 🡪 Redirecting User 🡪 User payment not processed properly
* Excess Line Endings 🡪 Error in Ajax response/JSON
* Downloads in our web app may causing the app to break in the long run

Therefore, leaving an extra space or blank line after the closing tag will cause problem. When other scripts will have this file included, it will result in headers being sent to the Web browser. If the including (parent) script later attempts to send a cookie, start a session, or redirect the browser, we will get “headers already sent” error message.

## Config File

**$\_SERVER['DOCUMENT\_ROOT']**

Get the root of each site

**Custom Error Handler**

It is useful to automatically notify the coder where the problem lies.

**set\_error\_handler("err\_handler")**

This script can declare a catch-all function that is called by PHP when an error condition occurs

**err\_handler($e\_level, $e\_message, $e\_file, $e\_line, $e\_vars)**

Error Levels

|  |  |
| --- | --- |
| 2 | Warnings |
| 8 | Notice |
| 256 | User Error |
| 512 | User Warning |
| 1024 | User Notice |
| 4096 | Recoverable Error |
| 8191 | All |

The **debug\_backtrace()** function generates a PHP back trace.

This function displays data from the code that led up to the **debug\_backtrace()** function.

Returns an array of associative arrays. The possible returned elements are:

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Description** |
| function | string | The current function name |
| line | integer | The current line number |
| file | string | The current file name |
| class | string | The current class name |
| object | object | The current object |
| type | string | The current call type. Possible calls:  Returns: "->"  - Method call  Returns: "::"  - Static method call  Returns nothing - Function call |
|  |  |
|  |  |
|  |  |
| args | array | If inside a function, it lists the functions arguments. If inside an included file, it lists the included file names |

**error\_log()**

This function sends an error message to a log, to a file, or to a mail account.

**basename($\_SERVER[‘PHP\_SELF’])**

This function returns the filename from a path.

|  |  |
| --- | --- |
| **$\_SERVER[‘DOCUMENT\_ROOT’]** | **$\_SERVER[‘HTTP\_HOST’]** |
| The root directory of this site defined by the 'DocumentRoot' directive in the General Section or a section e.g.  **var/www/example**  C:/xampp/htdocs | The base URL of the host e.g.  [**www.example.com**](http://www.example.com)  localhost |

**For PHP**

If we want to include a file we cannot use **HTTP\_HOST**, we need to use **DOCUMENT\_ROOT**

include**(**$\_SERVER['DOCUMENT\_ROOT'].'/E-Commerce/includes/mysql.inc.php')

**For HTML**

**../** 🡪 means parent of the current folder

**/ 🡪** means the root folder

**./ 🡪** means the current folder

In <script> or <link> tag, we need to use

<script src = "/E-Commerce/vendor/jquery.min.js"></script>

**MAGIC Constants**

|  |  |
| --- | --- |
| \_\_LINE\_\_ | The current line number of the file. |
| \_\_FILE\_\_ | The full path and filename of the file with symlinks resolved. If used inside an include, the name of the included file is returned. |
| \_\_DIR\_\_ | The directory of the file. If used inside an include, the directory of the included file is returned. This is equivalent to dirname(\_\_FILE\_\_). This directory name does not have a trailing slash unless it is the root directory. |
| \_\_FUNCTION\_\_ | The function name. |
| \_\_CLASS\_\_ | The class name. The class name includes the namespace it was declared in (e.g. Foo\Bar). Note that as of PHP 5.4 \_\_CLASS\_\_ works also in traits. When used in a trait method, \_\_CLASS\_\_ is the name of the class the trait is used in. |
| \_\_TRAIT\_\_ | The trait name. The trait name includes the namespace it was declared in (e.g. Foo\Bar). |
| \_\_METHOD\_\_ | The class method name. |
| \_\_NAMESPACE\_\_ | The name of the current namespace. |

**get\_magic\_quotes\_gpc()**

It is a function that checks the configuration (php.ini) and returns 0 if magic\_quotes\_gpc is off (otherwise it returns 1).

When magic\_quotes are on, all ' (single-quote), " (double quote), \ (backslash) and NULLs are escaped with a backslash automatically. This is to prevent all sorts of injection security issues.

**MD5, SHA1, and SHA256 Avoid it**

Hashing algorithms such as MD5, SHA1 and SHA256 are designed to be very fast and efficient. With modern techniques and computer equipment, it has become trivial to "brute force" the output of these algorithms, in order to determine the original input.

Because of how quickly a modern computer can "reverse" these hashing algorithms, many security professionals strongly suggest against their use for password hashing.

**password\_hash($string, PASSWORD\_DEFAULT)**

Creates a password hash

**password\_verify(Sstring, $hash)**

Verifies that a password matches a hash

**DATE\_ADD(NOW(), INTERVAL 1 MONTH) 🡪 MySQL**

Adds 1 month to the current date and time

**mysqli\_affected\_rows($dbc) == 1 🡪 MySQL**

Checks whether any rows got affected after updating

## Activating Accounts

* When the user registers, a random code is stored in the users table
* An email is sent to the registered email address, which includes a link to an activation page on the site. The link passes the user’s email address and the specific code to the PHP page.

Ex: <https://www.example.com/activate.php?x=email@example.com&y=CODE>

* The PHP page confirms that there is a record in a table with that combination of email address and code, then activates that account (normally by setting the code column to NULL)
* When the user logs in, the query must confirm that t the email and password combination is correct, and that the code column in the table has a NULL value

This is called a “closed-loop” confirmation process and prevents fake registrations.

**$\_SERVER[‘HTTP\_REFERRER’]**

Refer to the page where it got directed from

**<meta http-equiv='refresh' content='1'>**

This refereshes the page and it is a lot better option than the header function of PHP

**session\_regenerate\_id()**

Update the current session id with a newly generated one

## Interpreter (Hyperlink Reference Links)

**No words** 🡪 It will append it to the current page

**Add a slash / to the front of it** 🡪 It appends to our base URL

**Add a # sign** 🡪 it is an in-page anchor or a valid URL

## Hypertext Access

**RewriteEngine On**

# Turn on the rewriting engine

**RewriteRule ^pet-care/?$ pet\_care\_info\_01\_02\_2008.php [NC,L]**

# Handle requests for "pet-care"

**RewriteRule** 🡪 tells Apache that this like refers to a single RewriteRule.

^/pet-care/?$ 🡪 The "pattern"

pet\_care\_info\_01\_02\_2003.php 🡪 The "substitution".

**[NC,L] - "Flags"** 🡪 tells Apache how to apply the rule

**"NC"** 🡪 tells Apache that this rule should be case-insensitive

**"L"** 🡪 tells Apache not to process any more rules if this one is used.

The server will check the URL of every request to the site to see if this pattern matches. If it does, then Apache will swap the URL of the request for the "substitution" section that follows.

If the pattern above matches the request, Apache uses this URL instead of the requested URL.

**RewriteRule ^products/([0-9]+)/?$ show\_product.php?product\_id=$1 [NC,L]**

# Handle product requests

This allows us to "back-reference" (refer back to) that section of the URL in the following "substitution" section. The "$1" in the substitution tells Apache to put whatever matched the earlier bracketed pattern into the URL at this point. We can have lots of backreferences, and they are numbered in the order they appear.

## Paypal

* No setup fees
* No monthly costs for the basic payment options
* Fraud protections
* Shipping calculators and shipping label service
* Tax calculators
* Availability in 190 countries and with 21 currencies
* Currency Conversion
* International Tax and shipping calculators
* Ability to send invoices
* Inventory management
* A virtual terminal to manually process transactions
* Integration with popular third-party e-commerce systems

What PayPal doesn’t do is transfer money directly into our bank account. The funds received from transactions get applied to our PayPal account. The funds received from transactions get applied to our PayPal account. We can then go into our PayPal account and transfer money to our registered back account, which takes couple of days to go through.

PayPal offers two base solutions, named Website Payment Standard and Website Payment Pro. Both solutions accept payment from PayPal users or via credit cards. The pro solution provides extra security measurement and fraud application. The pro system is $30 system per month and we have to complete a credit application to qualify. Transaction fee 30 cents per transaction.

PayPal Fees

|  |  |
| --- | --- |
| **Monthly Sales** | **Per Transaction Fee** |
| $0 to $3,000 | 2.9% + $0.3 |
| $3,000.01 to $10,000 | 2.5% + $0.3 |
| $10,000.01 to $100.000 | 2.2% + $0.3 |
| over $100,000 | 1.9% + $0.3 |

There are different types of default buttons for different situations

* Buy Now, for selling single items
* Add to Cart, for selling multiple items
* Subscribe, for selling subscriptions
* Donate, for accepting donations

### Subscription

The subscribe feature comes in two formats

* Recurring payments via a PayPal accounts
* Recurring payments without a PayPal account

The latter allows anyone to use a credit card to make the purchase; there is a $19.99/month fee (to us, the merchant) for this service.

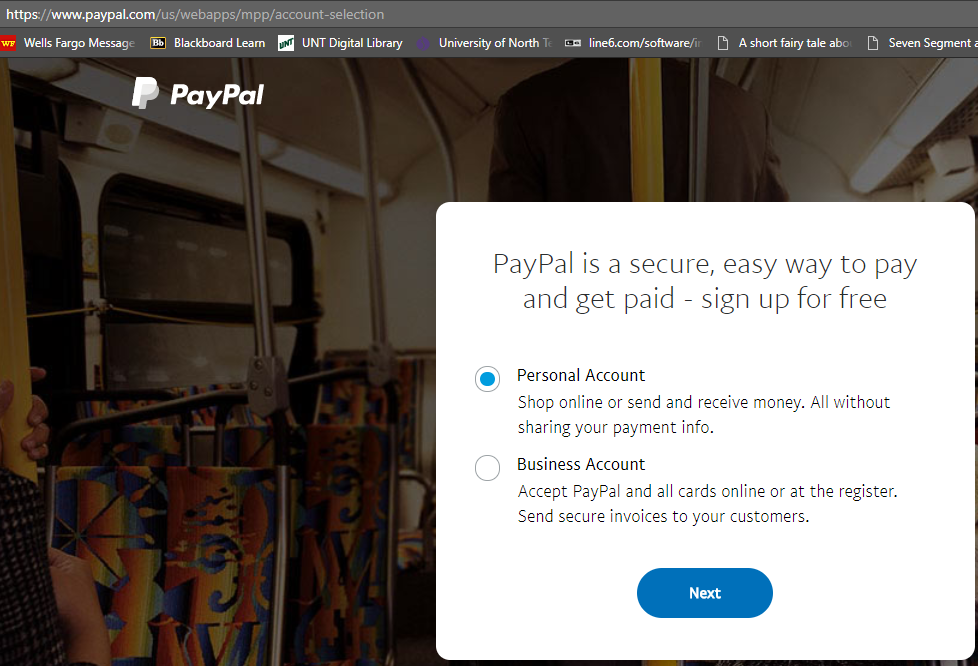
Three different sites of PayPal

* [www.paypal.com](http://www.paypal.com) 🡪 Need a PayPal account there representing the site, see transactions
* [www.developer.paypal.com](http://www.developer.paypal.com) 🡪 Test and use test accounts, register with email address
* [www.sandbox.paypal.com](http://www.sandbox.paypal.com) 🡪 Exactly like real PayPal for developers

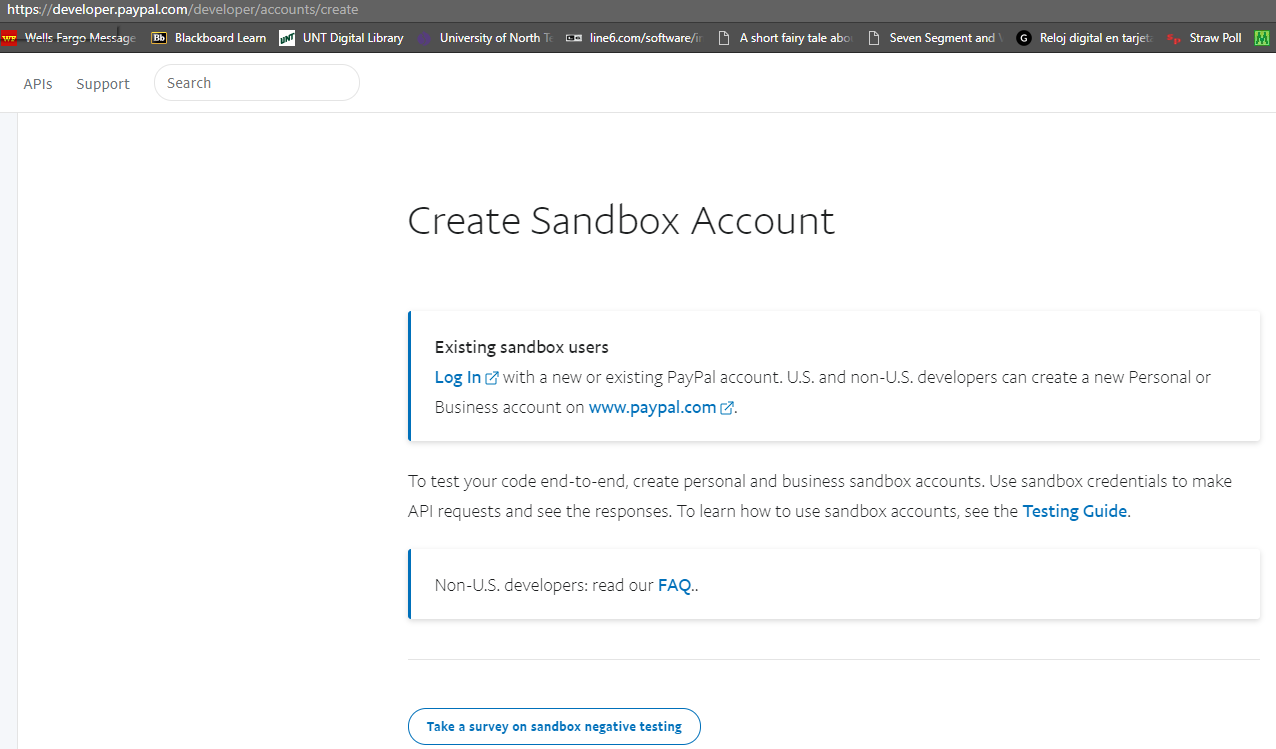
## Sandbox Account and Developer

**Steps**

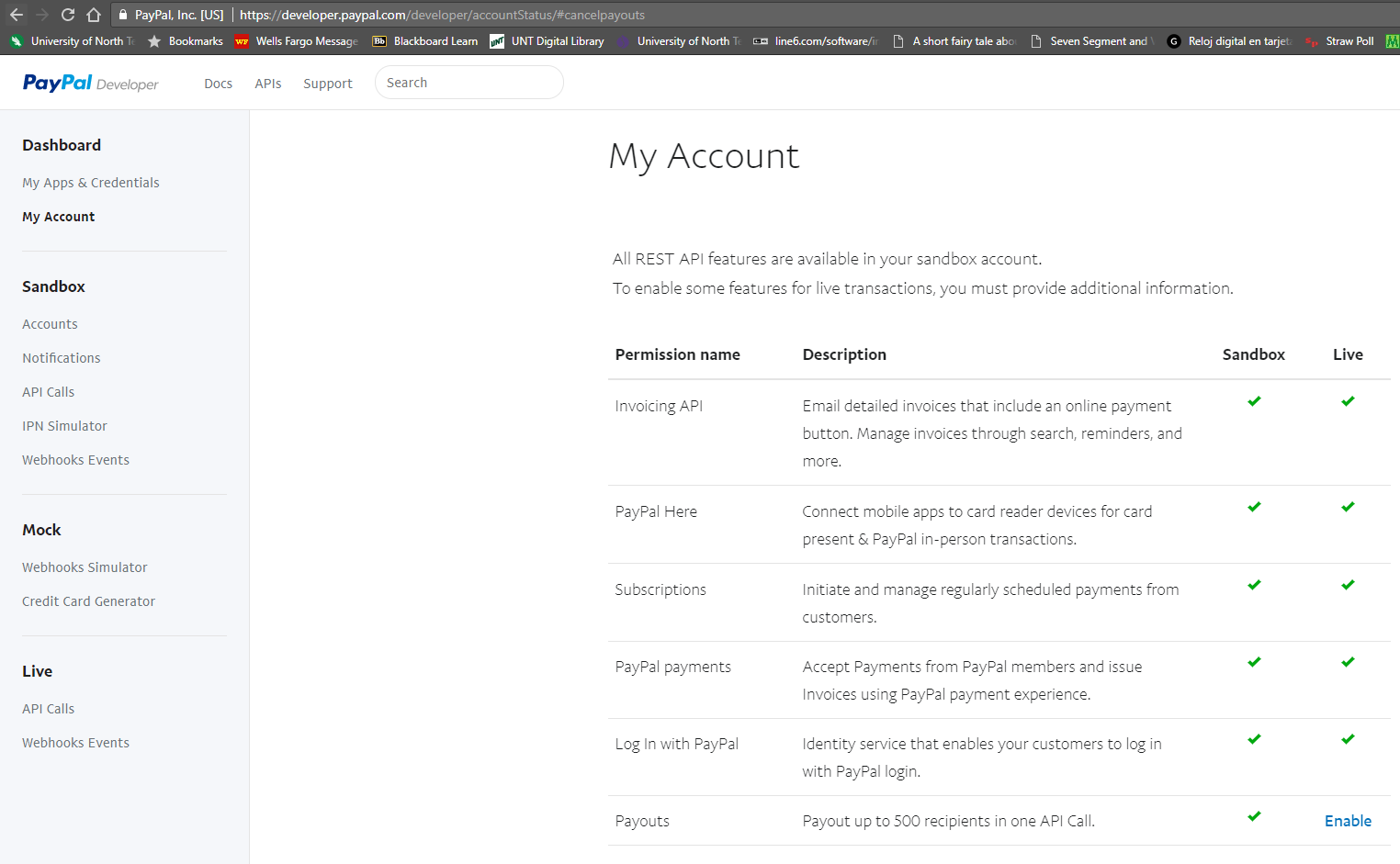
* Create a normal **PayPal**(we can use fake ids) account (Normal)



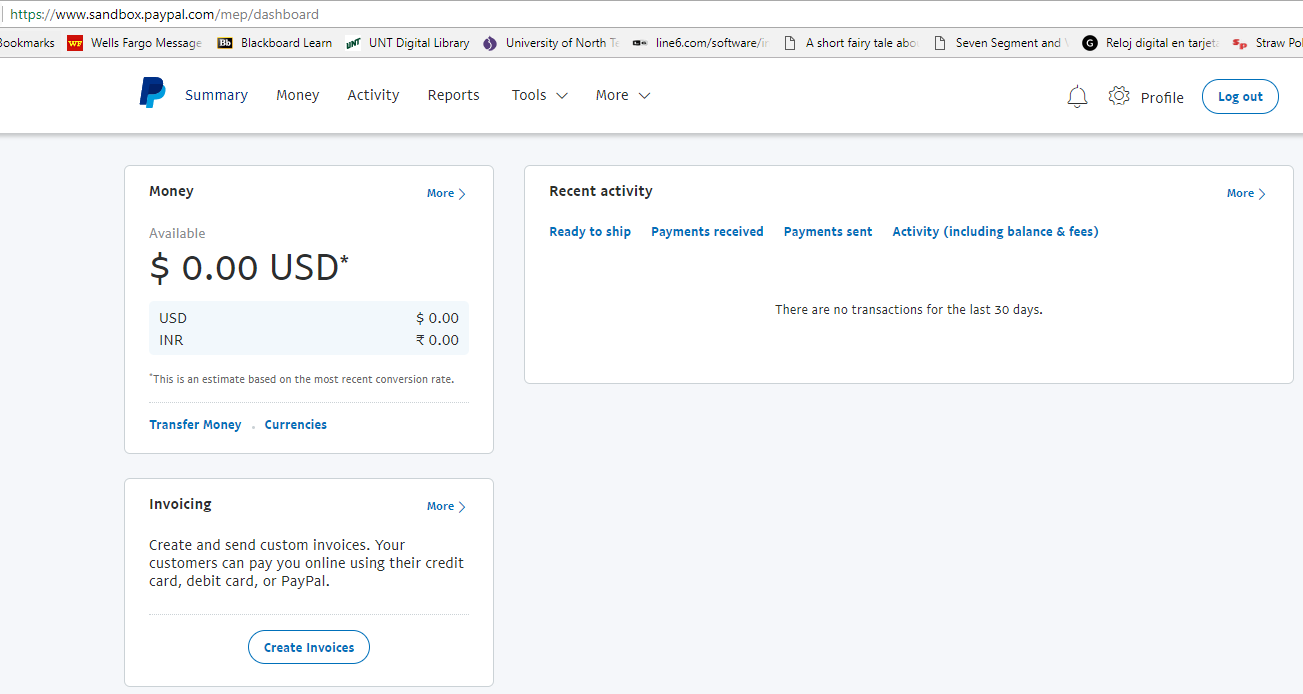
* Go to <https://developer.paypal.com/developer/accounts/create> (Developer)



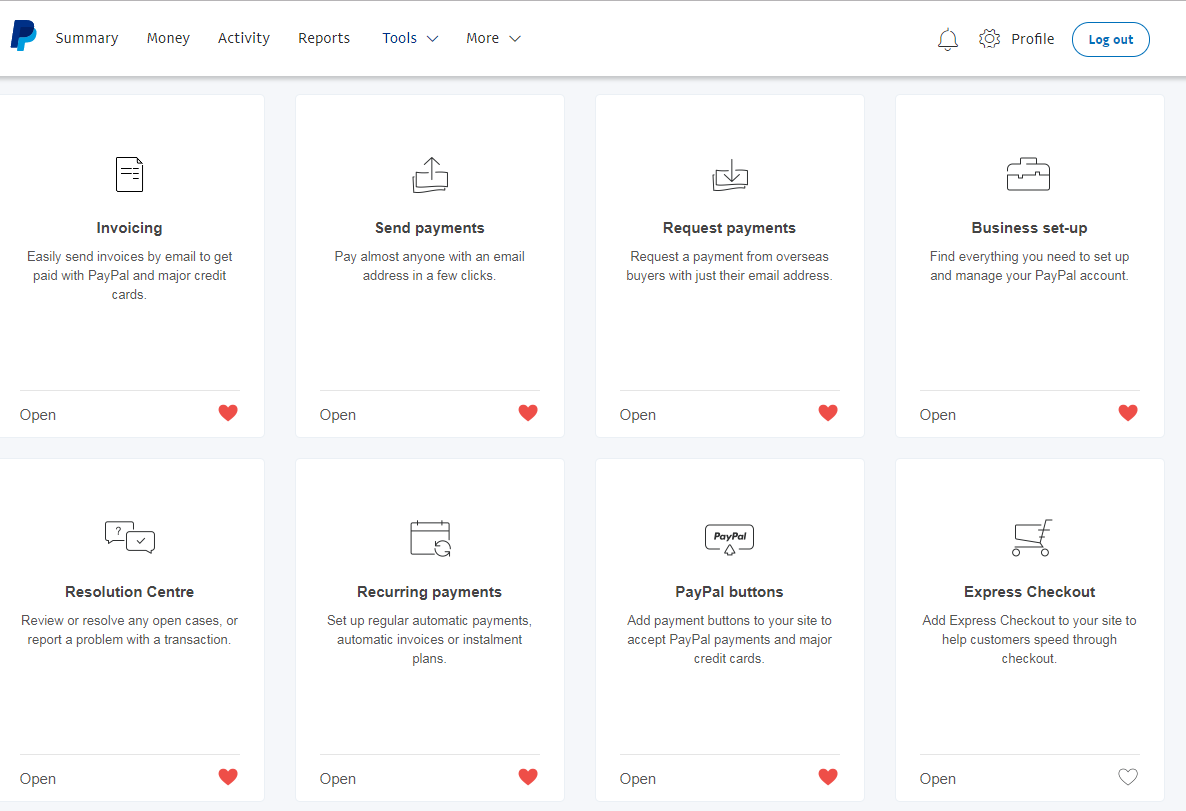
* Go to log in and type in the password and we would able to see our account (Developer)



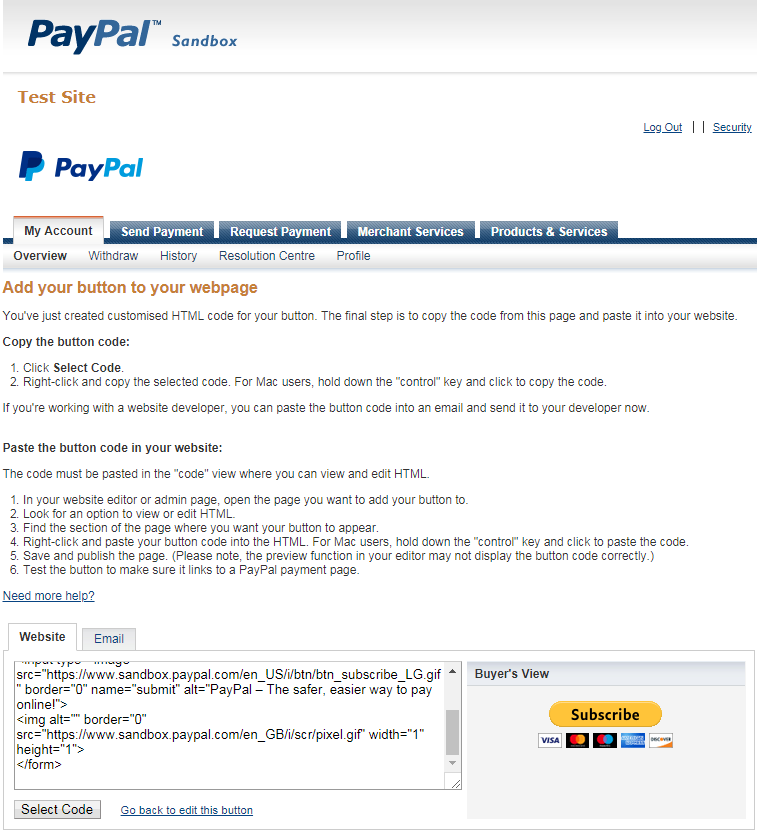
* Now log in to the sandbox account (Sandbox)



* Go to Tools 🡪 All Tools (Sandbox)



* Go to any section 🡪 right now, we are going to PayPal button (Sandbox)



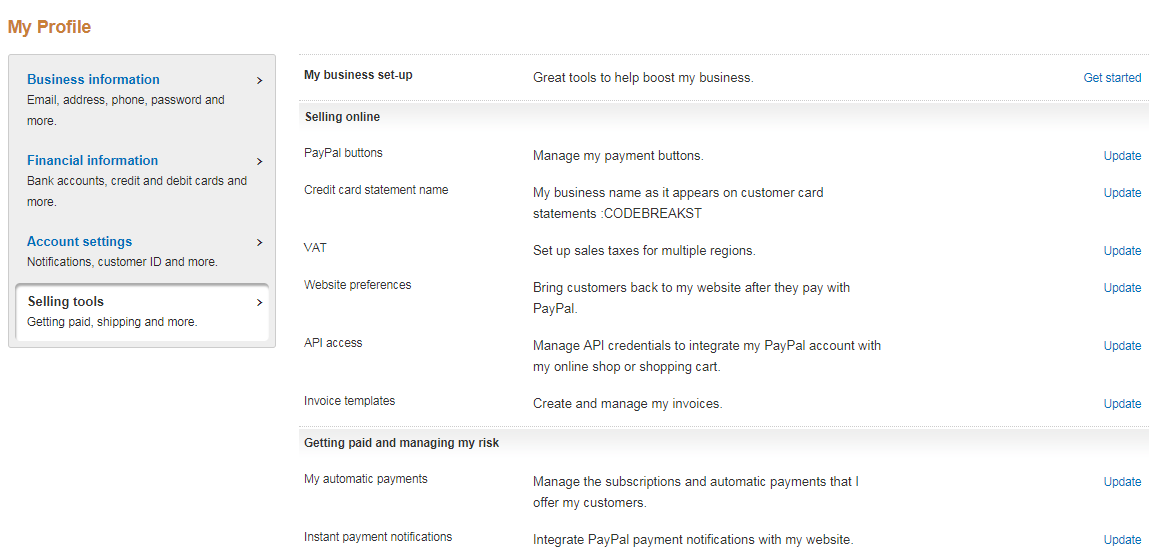
### Using IPN (Instant Payment Notification)

The site only reacts to a user action: If the user doesn’t return to our site before the session expires, they won’t be created with a year of access. This is pretty bad form, because the user’s action of sending us money via PayPal should be sufficient. **IPN** will notify a Web Site when payment has been processed. It will notify us with an email and have a rather a server-to-server communication that neither the customer nor the site’s administrator will witness. Through this e-commerce site can verify the transaction and update the database accordingly.

Integrating IPN is a two-part process. enabling it on a PayPal account and creating the listening script on the server.

### Enabling IPN

* Go to Profile 🡪 Profile Settings 🡪 Selling tools 🡪 Integrating payment notifications



### NOTE

If you are not receiving the correct response from Paypal ensure that you are using the main test account (Verified Business Account) from your Paypal Sandbox account.

Also ensure that we are testing the Paypal IPN Script on an online webserver (Not MAMP, Xampp etc..) as Paypal requires a reachable 'return url', 'cancel url' and 'notify url'.

### Account Name

**Email:** khan.photon2@gmail.com

**Merchant Fake Email:** [photonkhan@gmail.com](mailto:photonkhan@gmail.com)

**Shopper Fake Email:** rob.marley@gmail.com

### How it works

We will send the request through a form (with all the nitty-gritty details of the customers information) 🡪 We will link up with PayPal server with our IPN script which will keep track of what is being sent to the PayPal server 🡪 IPN scripts verifies whether the post that is being sent matches the intention or not 🡪 Then it sends whether the transaction happened or not

### Use Header

**$header = "POST /cgi-bin/webscr HTTP/1.1\r\n";**

**$header .= "Host:** [**www.sandbox.paypal.com\r\n**](http://www.sandbox.paypal.com\r\n)**";**

## Terms Summary

**mysqli\_set\_charset($dbc, 'utf8')**

It indicates which character set should be used for communications between PHP and the database.

**$\_SERVER['DOCUMENT\_ROOT']**

Get the root of each site

**debug\_backtrace()**

This function generates a PHP back trace.

**error\_log()**

This function sends an error message to a log, to a file, or to a mail account.

**basename($\_SERVER[‘PHP\_SELF’])**

This function returns the filename from a path.

**password\_hash($string, PASSWORD\_DEFAULT)**

Creates a password hash

**password\_verify(Sstring, $hash)**

Verifies that a password matches a hash

**$\_SERVER[‘HTTP\_REFERRER’]**

Refer to the page where it got directed from

**<meta http-equiv='refresh' content='1'>**

This refereshes the page and it is a lot better option than the header function of PHP

**session\_regenerate\_id()**

Update the current session id with a newly generated one

## Code

### About

1. <? php require\_once($\_SERVER['DOCUMENT\_ROOT'].
2. '/E-Commerce/includes/config.inc.php');
3. $page\_title = "About";
4. require\_once($\_SERVER['DOCUMENT\_ROOT'].
5. '/E-Commerce/includes/header.php');
6. require\_once(MYSQL);
7. require($\_SERVER['DOCUMENT\_ROOT'].
8. '/E-Commerce/includes/form\_functions.inc.php'); // $\_SESSION['user\_id'] = 1; // $\_SESSION['user\_type'] = 'admin';
9. $body = '<h3 class = "pl-2"> About </h3> < p class = "pl-2" > Welcome to Knowledge is Power, a site dedicated to keeping you up to date on the Web security and programming information you need know < /p>';
10. require\_once($\_SERVER['DOCUMENT\_ROOT'].
11. "/E-Commerce/includes/body.php");
12. require\_once($\_SERVER['DOCUMENT\_ROOT'].
13. "/E-Commerce/includes/footer.php"); ?>

### Add Page

1. <? php require\_once($\_SERVER['DOCUMENT\_ROOT'].
2. "/E-Commerce/includes/config.inc.php");
3. $page\_title = "Add a Site Content Page";
4. require\_once($\_SERVER['DOCUMENT\_ROOT'].
5. "/E-Commerce/includes/header.php");
6. require(MYSQL);
7. $add\_page\_errors = array();
8. $add\_page\_success = false;
9. if (isset($\_SESSION['user\_admin'])): if ($\_SERVER['REQUEST\_METHOD'] == 'POST' && isset($\_POST['add\_page'])): if (!empty(trim($\_POST['title']))): $p\_title = mysqli\_real\_escape\_string($dbc, strip\_tags($\_POST['title']));
10. else :$add\_page\_errors['title'] = "Please enter the title!";
11. endif;
12. if (filter\_var($\_POST['category'], FILTER\_VALIDATE\_INT, ['min\_range' => 1])): $category = $\_POST['category'];
13. else :$add\_page\_errors['category'] = "Please select a category";
14. endif;
15. if (!empty(trim($\_POST['description']))): $description = mysqli\_real\_escape\_string($dbc, strip\_tags($\_POST['description']));
16. else :$add\_page\_errors['description'] = "Please enter the description!";
17. endif;
18. if (!empty(trim($\_POST['content']))): $allowed = '<div><p><span><br><a><img><h1><h2><h3><h3><h4><ul><li><blockquote><code><strong>';
19. $content = mysqli\_real\_escape\_string($dbc, strip\_tags($\_POST['content'], $allowed));
20. else :$add\_page\_errors['content'] = "Please enter the content";
21. endif;
22. if (empty($add\_page\_errors)): $query = "INSERT INTO pages(category\_id, title, description, content)
23. VALUES('$category', '$p\_title', '$description', '$content')
24. ";
25. $result = mysqli\_query($dbc, $query);
26. if (mysqli\_affected\_rows($dbc) == 1): $add\_page\_success = true;
27. $\_POST = array();
28. else :trigger\_error("The page could not be added due to a system error. We apologize for the inconvenience!");
29. endif;
30. endif;
31. endif;
32. $body = " < h3 class = 'pl-2' > Add a Site Content Page < /h3> < form class = 'pl-2'
33. method = 'post'
34. action = '".str\_replace('
35. \_ ',' - ',basename($\_SERVER['
36. PHP\_SELF '],'.php '))."' > < div class = 'form-group' > < label
37. for = 'title' > < strong > Title < /strong></label > < input type = 'text'
38. name = 'title'
39. id = 'title'
40. class = 'form-control'
41. value = '";
42. $body. = isset($p\_title) ? $p\_title : '';
43. $body. = "'/>";
44. $body. = " < span class = 'text-danger' > ";
45. $body. = isset($add\_page\_errors['title']) ? $add\_page\_errors['title'] : '';
46. $body. = " < /span> < /div> < div class = 'form-group' > < label
47. for = 'category' > < strong > Category < /strong></label > < select name = 'category'
48. class = 'form-control' > ";
49. $query = "SELECT id, category from categories ORDER BY category ASC";
50. $result = mysqli\_query($dbc, $query);
51. while ($row = mysqli\_fetch\_array($result)): $body. = "<option value = '".$row['id'].
52. "' ";
53. $body. = (isset($\_POST['category']) && $\_POST['category'] == $row['id']) ? "selected='selected'>" : '>';
54. $body. = $row['category'].
55. "</option>";
56. endwhile;
57. $body. = "</select> < span class = 'text-danger' > ";
58. $body. = isset($add\_page\_errors['category']) ? $add\_page\_errors['category'] : "";
59. $body. = " < /span> < /div> < div class = 'form-group' > < label
60. for = 'description' > < strong > Description < /strong></label > < textarea name = 'description'
61. class = 'form-control'
62. id = 'description' > ";
63. $body. = isset($description) ? $description : '';
64. $body. = "</textarea>";
65. $body. = (isset($add\_page\_errors['description'])) ? "<span class = 'text-danger'>".$add\_page\_errors['description'].
66. "</span>": "";
67. $body. = " < /div> < div class = 'form-group' > < label
68. for = 'content' > < strong > Content < /strong></label > < textarea name = 'content'
69. class = 'form-control'
70. id = 'content' > ";
71. $body. = isset($content) ? $content : '';
72. $body. = "</textarea>";
73. $body. = (isset($add\_page\_errors['content'])) ? "<span class = 'text-danger'>".$add\_page\_errors['content'].
74. "</span>": "";
75. $body. = " < /div> < input type = 'submit'
76. class = 'btn btn-success'
77. name = 'add\_page'
78. value = 'Add this Page' / > < /form>";
79. $body. = "<script src = '/E-Commerce/vendor/tinymce/tinymce.min.js'></script>";
80. $body. = "<script src = '/E-Commerce/js/add\_page.js'></script>";
81. if ($add\_page\_success): $body. = "<code class = 'text-success bg-dark'> The page has been added</code>";
82. endif;
83. else :$body = "<h1 class = 'text-danger'> You are not authorized to enter this page!</h1>";
84. endif;
85. require\_once($\_SERVER['DOCUMENT\_ROOT'].
86. "/E-Commerce/includes/body.php");
87. require\_once($\_SERVER['DOCUMENT\_ROOT'].
88. "/E-Commerce/includes/footer.php"); ?>

### Add Pdf

1. <? php require\_once($\_SERVER['DOCUMENT\_ROOT'].
2. '/E-Commerce/includes/config.inc.php');
3. $page\_title = "Add a PDF";
4. require\_once($\_SERVER['DOCUMENT\_ROOT'].
5. '/E-Commerce/includes/header.php');
6. require\_once(MYSQL);
7. if (isset($\_SESSION['user\_admin'])): $add\_pdf\_errors = array();
8. if ($\_SERVER['REQUEST\_METHOD'] == 'POST' && isset($\_POST['upload'])): if (empty($\_POST['title'])): $add\_pdf\_errors['title'] = "Please enter the title!";
9. else :$t = mysqli\_real\_escape\_string($dbc, strip\_tags($\_POST['title']));
10. endif;
11. if (empty($\_POST['description'])): $add\_pdf\_errors['description'] = "Please enter the description";
12. else :$d = mysqli\_real\_escape\_string($dbc, trim(strip\_tags($\_POST['description'])));
13. endif;
14. if (is\_uploaded\_file($\_FILES['pdf']['tmp\_name']) && $\_FILES['pdf']['error'] == UPLOAD\_ERR\_OK): $file = $\_FILES['pdf'];
15. $size = round($file['size'] / 1024);
16. if ($size > 1024): $add\_pdf\_errors['pdf'] = "The uploaded file was too large";
17. else :if (($file['type'] != 'application/pdf') && substr($file['name'], -4) != ".pdf"): $add\_pdf\_errors['pdf'] = "The uploaded file was not a PDF";
18. else :if (array\_key\_exists('pdf', $add\_pdf\_errors)): unlink($file['tmp\_name']);
19. else :if (empty($add\_pdf\_errors)): $file\_name = $file['name'];
20. $tmp\_name = sha1($file['name'].uniqid("", true));
21. $query = "INSERT INTO pdfs (tmp\_name, title, description, file\_name, size)
22. VALUES('$tmp\_name', '$t', '$d', '$file\_name', '$size')
23. ";
24. $result = mysqli\_query($dbc, $query);
25. if (mysqli\_affected\_rows($dbc) == 1): move\_uploaded\_file($file['tmp\_name'], $\_SERVER['DOCUMENT\_ROOT'].
26. '/E-Commerce/includes/pdfs/'.$file\_name);
27. $success\_message = "<p class = 'mt-2 ml-2 text-success'>Successfully PDF Uploaded!</p>";
28. endif;
29. endif;
30. endif;
31. endif;
32. endif;
33. else :switch ($\_FILES['pdf']['error']):
34. case 1:
35. break;
36. case 2:
37. $add\_pdf\_errors['pdf'] = "The uploaded file was too large";
38. break;
39. case 3:
40. $add\_pdf\_errors['pdf'] = "The file was partially uploaded";
41. break;
42. case 4:
43. break;
44. case 5:
45. break;
46. case 6:
47. break;
48. case 7:
49. break;
50. case 8:
51. $add\_pdf\_errors['pdf'] = "No file was uploaded!";
52. break;
53. default:
54. $add\_pdf\_errors['pdf'] = "No file was uploaded!";
55. break;
56. endswitch;
57. endif;
58. endif;
59. $body = "<h3 class = 'pl-2'> Add a PDF </h3>";
60. $body. = " < form enctype = 'multipart/form-data'
61. action = '".str\_replace('
62. \_ ',' - ',basename($\_SERVER['
63. PHP\_SELF '],'.php '))."'
64. method = 'post'
65. class = 'pl-2' > ";
66. $body. = "<input type = 'hidden' class = 'form-control' name = 'MAX\_FILE\_SIZE' value = '1048576'/>";
67. $body. = "<div class = 'form-group'> < label
68. for = 'title' > < strong > Title < /strong></label > < input type = 'text'
69. name = 'title'
70. id = 'title'
71. class = 'form-control'
72. value = '";
73. $body. = isset($t) ? $t : '';
74. $body. = "'/>";
75. $body. = " < span class = 'text-danger' > ";
76. $body. = isset($add\_pdf\_errors['title']) ? $add\_pdf\_errors['title'] : '';
77. $body. = " < /span> < /div>";
78. $body. = "<div class = 'form-group'> < label
79. for = 'title' > < strong > Description < /strong></label > < textarea = 'text'
80. name = 'description'
81. id = 'description'
82. class = 'form-control' > ";
83. $body. = isset($d) ? $d : '';
84. $body. = "</textarea>";
85. $body. = " < span class = 'text-danger' > ";
86. $body. = isset($add\_pdf\_errors['description']) ? $add\_pdf\_errors['description'] : '';
87. $body. = " < /span> < /div>";
88. $body. = "<div class = 'form-group'> < label
89. for = 'pdf' > PDF < /label> < input type = 'file'
90. class = 'form-control'
91. id = 'pdf'
92. name = 'pdf' / > < span class = 'text-danger' > ";
93. $body. = isset($add\_pdf\_errors['pdf']) ? $add\_pdf\_errors['pdf'] : '';
94. $body. = " < /span> < /div>";
95. $body. = "<input type = 'submit' class = 'btn btn-success' name = 'upload' value = 'Add a PDF'/>";
96. $body. = "</form>";
97. $body. = isset($success\_message) ? $success\_message : '';
98. require\_once($\_SERVER['DOCUMENT\_ROOT'].
99. "/E-Commerce/includes/body.php");
100. require\_once($\_SERVER['DOCUMENT\_ROOT'].
101. "/E-Commerce/includes/footer.php");
102. endif; ?>

### Body

1. < div class = "row page" > < div class = "col-10 content my-2" > <? php
2. if (isset($body)) echo $body; ?> < /div> < div class = "sidebar col-2 d-none d-sm-block" > < div class = "my-2" > < h4 class = "text-success bg-dark card title" > Manage Your Accounts < /h4> < ul class = "list-unstyled card" > < li > < a href = "/E-Commerce/renew-account"
3. class = "nounderline"
4. title = "Renew Your Account" > < code class = "text-secondary" > Renew Account < /code> < /a>   < /li> < li > < a href = "/E-Commerce/change-password"
5. class = "nounderline"
6. title = "Change Your Password" > < code class = "text-secondary" > Change Password < /code> < /a> < /li> <!--     <li>
7. <a href = "/E-Commerce/favorites" class = "nounderline" title = "View Your Favorites">
8. <code class = "text-secondary"> Favorites </code>
9. </a>
10. </li>
11. <li>
12. <a href = "/E-Commerce/recommendations" class = "nounderline" title = "View Your Recommendations">
13. <code class = "text-secondary"> Recommendations </code>
14. </a>
15. </li> -->
16. <li>
17. <a href = "/E-Commerce/pdf" class = "nounderline" title = "View Our Popular PDFs">
18. <?php
19. $query = "SELECT tmp\_name, title, description, size, file\_name FROM pdfs ORDER BY date\_created DESC";
20. $result = mysqli\_query($dbc, $query);
21. $number = mysqli\_num\_rows($result);
22. ?>
23. <code>
24. Popular
25. <span class = "badge badge-dark"><?php echo $number; ?></span>
26. </code>
27. </a>
28. </li>
29. <li>
30. <a href = "/E-Commerce/logout" class = "nounderline" title = "Logout">
31. <code class = "text-secondary"> Logout </code>
32. </a>
33. </li>
34. </ul>
35. </div>
36. <div class = "my-2">
37. <h4 class = "text-info bg-dark card title"> Content</h4>
38. <ul class = "list-unstyled card">
39. <?php
40. $dbc = mysqli\_connect("localhost", "root", "", "ecommerce1")
41. or die("Server Denied");
42. $query = "SELECT \* FROM categories ORDER BY category";
43. $result = mysqli\_query($dbc, $query) or die ("Query Denied");
44. while($row = mysqli\_fetch\_array($result)):
45. echo '<li>';
46. echo '  <a href = "/E-Commerce/category/'.$row['id'].'" class = "nounderline" title = "'.$row['category'].'">';
47. echo '      <code class = "text-secondary"> '.$row['category'].' </code>';
48. echo '  </a>  ';
49. echo '</li>';
50. endwhile;
51. ?>
52. </ul>
53. </div>
55. <?php if(isset($\_SESSION['user\_admin'])): ?>
56. <div class = "my-2">
57. <h4 class = "text-danger bg-dark card title"> Administration</h4>
58. <ul class = "list-unstyled card">
59. <li>
60. <a href = "/E-commerce/add-page" class = "nounderline" title = "Add a Page">
61. <code class = "text-secondary"> Add a Page </code>
62. </a>
63. </li>
64. <li>
65. <a href = "/E-Commerce/add-pdf" class = "nounderline" title = "Add a PDF">
66. <code class = "text-secondary"> Add a PDF </code>
67. </a>
68. </li>
69. </ul>
70. </div>
71. <?php endif; ?>
72. <?php
73. $login\_access\_pages = ['index', 'about', 'contact', 'register'];
74. $login\_page = basename($\_SERVER['PHP\_SELF'],".php");
75. if(!isset($\_SESSION['user\_id'])):
76. if(in\_array($login\_page, $login\_access\_pages)):
77. include('login\_form.inc.php');
78. endif;
79. endif;
80. ?>
81. </div>
82. </div>

### Category

1. <? php require($\_SERVER['DOCUMENT\_ROOT'].
2. "/E-Commerce/includes/config.inc.php");
3. require(MYSQL);
4. if (isset($\_GET['id']) && filter\_var($\_GET['id'], FILTER\_VALIDATE\_INT, ['min\_range' => 1])): $query = "SELECT category FROM categories WHERE id = '".$\_GET['id'].
5. "'";
6. $result = mysqli\_query($dbc, $query);
7. if (mysqli\_num\_rows($result) != 1): $page\_title = "Error!";
8. require($\_SERVER['DOCUMENT\_ROOT'].
9. "/E-Commerce/includes/header.php");
10. $body = "<h1 class = 'ml-2 mt-2 text-danger'> This page has been accessed in error </h1>";
11. require($\_SERVER['DOCUMENT\_ROOT'].
12. "/E-Commerce/includes/body.php");
13. require($\_SERVER['DOCUMENT\_ROOT'].
14. "/E-Commerce/includes/footer.php");
15. exit();
16. endif;
17. $row = mysqli\_fetch\_array($result);
18. $page\_title = $row['category'];
19. include($\_SERVER['DOCUMENT\_ROOT'].
20. "/E-Commerce/includes/header.php");
21. if (isset($\_SESSION['user\_id']) && !isset($\_SESSION['user\_not\_expired'])): $body = "<h1 class = 'ml-2 mt-2 text-success'> Thank you for you interest in this content </h1>";
22. $body. = "<p class = 'ml-2 mt-2 text-danger'> Unfortunately you account has been expired. </p>";
23. $body. = "<p class = 'ml-2 mt-2 text-danger'> <a href = 'renew' class = 'nounderline'>Renew Your Account</a></p>";
24. $body. = "<p class = 'ml-2 mt-2 text-danger'>In order to access site content.</p>";
25. elseif(!isset($\_SESSION['user\_id'])): $body = "<h1 class = 'ml-2 mt-2 text-success'> Thank you for you interest in this content </h1>";
26. $body. = "<p class = 'ml-2 mt-2 text-danger'>You must be logged in as a registered user to view site content. </p>";
27. endif;
28. if (isset($\_SESSION['user\_id']) && isset($\_SESSION['user\_not\_expired'])): $query = "SELECT id, title, description FROM pages WHERE category\_id = '".$\_GET['id'].
29. "' ORDER BY date\_created DESC";
30. $result = mysqli\_query($dbc, $query);
31. if (mysqli\_num\_rows($result) > 0): $count = 0;
32. while ($row = mysqli\_fetch\_array($result)): if ($count == 0): $body = "<h3 class = 'ml-2'>";
33. endif;
34. $body. = "<h3 class = 'ml-2'>";
35. $body. = "<a href = '/E-Commerce/category/page/".$row['id'].
36. "' class = 'nounderline'>";
37. $body. = $row['title'];
38. $body. = "</a>";
39. $body. = "</h3>";
40. $body. = "<p class = 'ml-2'><code>";
41. $body. = $row['description'];
42. $body. = "</code></p'>";
43. $count++;
44. endwhile;
45. endif;
46. else :$page\_title = "Error!";
47. $body = "<h1 class = 'ml-2 mt-2 text-danger'> This page has been accesssed in error</h1>";
48. endif;
49. include($\_SERVER['DOCUMENT\_ROOT'].
50. "/E-Commerce/includes/body.php");
51. include($\_SERVER['DOCUMENT\_ROOT'].
52. "/E-Commerce/includes/footer.php");
53. endif; ?>

### Change Password

1. <? php require($\_SERVER['DOCUMENT\_ROOT'].
2. '/E-Commerce/includes/config.inc.php');
3. $page\_title = "Change your password";
4. include($\_SERVER['DOCUMENT\_ROOT'].
5. '/E-Commerce/includes/header.php');
6. require(MYSQL);
7. require($\_SERVER['DOCUMENT\_ROOT'].
8. '/E-Commerce/includes/form\_functions.inc.php');
9. if (isset($\_SESSION['user\_id'])): $change\_errors['pass'] = '';
10. $change\_errors['pass1'] = '';
11. $change\_errors['pass2'] = '';
12. $password\_change = false;
13. if ($\_SERVER['REQUEST\_METHOD'] == 'POST' && isset($\_POST['changepass'])): if (empty($\_POST['pass'])): $change\_errors['pass'] = "Field cannot be empty";
14. endif;
15. if (empty($\_POST['pass1']) || empty($\_POST['pass2'])): if (empty($\_POST['pass1'])): $change\_errors['pass1'] = "Field cannot be empty";
16. endif;
17. if (empty($\_POST['pass2'])): $change\_errors['pass2'] = "Field cannot be empty";
18. endif;
19. else :if ($\_POST['pass1'] != $\_POST['pass2']): $change\_errors['pass2'] = "Passwords do not match";
20. else :$pattern = "/^(?=.\*\d)(?=.\*[a-zA-Z])(?=.\*[\\_!-\.@#%]).{6,16}$/"; if (!preg\_match($pattern, $\_POST['pass1'])): $change\_errors['pass2'] = "Minimum 6 characters, Needs to have least 1 number and 1 special character";
21. else :$p = password\_hash(mysqli\_real\_escape\_string($dbc, $\_POST['pass1']), PASSWORD\_DEFAULT);
22. endif;
23. endif;
24. endif;
25. $change = 0;
26. foreach($change\_errors as $change\_error): if ($change\_error != ''): $change += 1;
27. endif;
28. endforeach;
29. if ($change == 0): $user\_id = $\_SESSION['user\_id'];
30. $query = "UPDATE users SET pass = '$p' WHERE id = '$user\_id'";
31. $result = mysqli\_query($dbc, $query);
32. if (mysqli\_affected\_rows($dbc) == 1): $password\_change = true;
33. endif;
34. endif;
35. endif;
36. $body = " < div > < h3 class = 'pl-2' > Change Your Password < /h3> < p class = 'ml-2' > Use the form below to change your password < /p> < form action = '".str\_replace('
37. \_ ',' - ',basename($\_SERVER['
38. PHP\_SELF '], '.php '))."'
39. method = 'post'
40. class = 'ml-2' > < div class = 'form-inline form-group' > < label
41. for = 'pass' > Current Password & nbsp; < /label> < input type = 'password'
42. name = 'pass'
43. id = 'pass'
44. class = 'form-control' / > < span class = 'text-danger' > ".$change\_errors['pass']." < /span> &nbsp; < /div> < div class = 'form-inline form-group' > < label
45. for = 'pass1' > New Password & nbsp; < /label> < input type = 'password'
46. name = 'pass1'
47. id = 'pass1'
48. class = 'form-control' / > < span class = 'text-danger' > ".$change\_errors['pass1']." < /span> &nbsp; < /div> < div class = 'form-inline form-group' > < label
49. for = 'pass2' > Confirm New Password & nbsp; < /label> < input type = 'password'
50. name = 'pass2'
51. id = 'pass2'
52. class = 'form-control' / > < span class = 'text-danger' > ".$change\_errors['pass2']." < /span> &nbsp; < /div> < input type = 'submit'
53. class = 'btn btn-success'
54. name = 'changepass'
55. value = 'Confirm' / > < /form>";
56. if ($password\_change): $body. = "<span class = 'text-success'>Password Changed Successfully</span>";
57. endif;
58. $body. = "</div>";
59. else :echo "<h1 class = 'text-danger'> You are not allowed to have access to this page </h1>";
60. echo "<meta http-equiv='refresh' content='1;url=/E-Commerce'/>";
61. endif;
62. require\_once($\_SERVER['DOCUMENT\_ROOT'].
63. "/E-Commerce/includes/body.php");
64. require\_once($\_SERVER['DOCUMENT\_ROOT'].
65. "/E-Commerce/includes/footer.php");

### Configuration

1. <? php $live = false;
2. $contact\_email = "khan.photon@gmail.com"; // define('BASE\_URI', $\_SERVER['DOCUMENT\_ROOT'].'/E-Commerce/includes/pdfs/'); // define('BASE\_URL', $\_SERVER['DOCUMENT\_ROOT'].'/E-Commerce/');
3. define('MYSQL', $\_SERVER['DOCUMENT\_ROOT'].
4. '/E-Commerce/includes/mysql.inc.php');
5. session\_start();
7. function my\_error\_handler($e\_number, $e\_message, $e\_file, $e\_line, $e\_vars) {
8. global $live, $contact\_email;
9. $message = "An error occured in script {$e\_file} on line {$e\_line}: \n {$e\_message}\n";
10. $message. = "<pre>".print\_r(debug\_backtrace(), 1).
11. "</pre>\n";
12. $message. = "<pre>".print\_r($e\_vars, 1).
13. "</pre>\n";
14. if (!$live): echo "<div class = 'error'>".nl2br($message).
15. "</div>";
16. else :error\_log($message, 1, $contact\_emails, "From:khan.photon@gmail.com");
17. endif;
18. if ($e\_number != E\_NOTICE): echo "<div class = 'error'> A system error occurred. We apologize for the inconvenience. </div>";
19. endif;
20. return true;
21. }
22. set\_error\_handler("my\_error\_handler");

### Contact

1. <? php require\_once($\_SERVER['DOCUMENT\_ROOT'].
2. '/E-Commerce/includes/config.inc.php');
3. $page\_title = "Contact";
4. require\_once($\_SERVER['DOCUMENT\_ROOT'].
5. '/E-Commerce/includes/header.php');
6. require\_once(MYSQL);
7. require($\_SERVER['DOCUMENT\_ROOT'].
8. '/E-Commerce/includes/form\_functions.inc.php'); // $\_SESSION['user\_id'] = 1; // $\_SESSION['user\_type'] = 'admin';
9. $body = '<h3 class = "pl-2"> Contact </h3> < p class = "pl-2" > Welcome to Knowledge is Power, a site dedicated to keeping you up to date on the Web security and programming information you need know < /p>';
10. require\_once($\_SERVER['DOCUMENT\_ROOT'].
11. "/E-Commerce/includes/body.php");
12. require\_once($\_SERVER['DOCUMENT\_ROOT'].
13. "/E-Commerce/includes/footer.php"); ?>

### Footer

1. <? php
2. if (isset($\_SESSION['user\_id'])): ?> < div class = "row text-center" > < div class = "col" > < h6 > Manage Your Account < /h6> < ul class = "list-unstyled list-inline" > < li class = "list-inline-item" > < a href = "/E-Commerce/renew-account"
3. class = "nounderline"
4. title = "Renew Your Account" > < code > Renew Your Account < /code> < /a> < /li> < li class = "list-inline-item" > < a href = "/E-Commerce/change-password"
5. class = "nounderline"
6. title = "Change Password" > < code > Change Password < /code> < /a> < /li> <!--   <li class = "list-inline-item">
7. <a href = "/E-Commerce/favorites" class = "nounderline" title = "View Your Favorites">
8. <code> Favorites </code>
9. </a>
10. </li> -->
11. <!-- <li class = "list-inline-item">
12. <a href = "/E-Commerce/recommentdations" class = "nounderline" title = "View Your Recommendations">
13. <code> Recommendations </code>
14. </a>
15. </li> -->
16. <li class = "list-inline-item">
17. <a href = "/E-Commerce/pdf" class = "nounderline" title = "View Our Popular PDFs">
18. <?php
19. $query = "SELECT tmp\_name, title, description, size, file\_name FROM pdfs ORDER BY date\_created DESC";
20. $result = mysqli\_query($dbc, $query);
21. $number = mysqli\_num\_rows($result);
22. ?>
23. <code>
24. Popular
25. <span class = "badge badge-dark"><?php echo $number; ?></span>
26. </code>
27. </a>
28. </li>
29. <li class = "list-inline-item">
30. <a href = "/E-Commerce/logout" class = "nounderline" title = "Log Out">
31. <code> Logout </code>
32. </a>
33. </li>
34. </ul>
35. </div>
36. </div>
37. <?php endif; ?>
38. <div class = "row footer navbar navbar-fixed-bottom text-center">
39. <div class = "col">
40. <!-- <a href = "site\_map.php" title = "Site Map"> Site Map |</a> -->
41. <!-- <a href = "policies.php" title = "Policies"> Policies |</a> -->
42. <code>&copy; <?php echo date("Y");?> All Rights Reserved | Designed By Photon &trade;</code>
43. </div>
44. </div>
45. </body>
46. </html>

### Forgot Password

<?php

1. require($\_SERVER['DOCUMENT\_ROOT'].
2. '/E-Commerce/includes/config.inc.php');
3. $page\_title = "Forgot Your Password?";
4. include($\_SERVER['DOCUMENT\_ROOT'].
5. '/E-Commerce/includes/header.php');
6. require(MYSQL);
7. require($\_SERVER['DOCUMENT\_ROOT'].
8. '/E-Commerce/includes/form\_functions.inc.php');
9. $pass\_errors['email'] = '';
10. $recovery\_message = false;
11. if ($\_SERVER['REQUEST\_METHOD'] == "POST" && isset($\_POST['recovery'])): if (filter\_var($\_POST['email'], FILTER\_VALIDATE\_EMAIL)): $e = mysqli\_real\_escape\_string($dbc, $\_POST['email']);
12. $query = "SELECT id FROM users WHERE email = '$e'";
13. $result = mysqli\_query($dbc, $query);
14. if (mysqli\_num\_rows($result) == 1): $row = mysqli\_fetch\_array($result, MYSQLI\_NUM);
15. $user\_id = $row[0];
16. $e\_recovery = 0;
17. foreach($pass\_errors as $pass\_error): if ($pass\_error != ''): $e\_recover += 1;
18. endif;
19. endforeach;
20. if ($e\_recovery == 0): $letters = 5;
21. $p = '';
22. for ($i = 0; $i < $letters; $i++): $p. = chr(rand(97, 122));
23. endfor;
24. $p. = "1!";
25. $pass\_hash = password\_hash($p, PASSWORD\_DEFAULT);
26. $query = "UPDATE users SET pass = '$pass\_hash' WHERE id = '$user\_id'";
27. $result = mysqli\_query($dbc, $query);
28. if (mysqli\_affected\_rows($dbc) == 1): $from = "khan.photon@gmail.com";
29. $to = $e;
30. $subject = "Password Recovery: Temporary Password";
31. $message = "Your password to log into CODE BREAK has been temporarily changed to $p Please log in using that password and this email address.";
32. $headers = "From:".$from;
33. if (mail($to, $subject, $message, $headers)): echo "Email sent.";
34. else :echo "Failed to send the email.";
35. endif;
36. $recovery\_message = true;
37. endif;
38. endif;
39. else :$pass\_errors['email'] = "Your email is not registered!";
40. endif;
41. else :$pass\_errors['email'] = "Invalid Email Address!";
42. endif;
43. endif;
44. if ($recovery\_message): $r\_message = "<h3 class = 'ml-2 mb-0'> Your password has been changed. </h3> < p class = 'ml-2 mb-0' > You will receive the new, temporary password via email. < /p> < p class = 'ml-2 mb-0' > Once you have logged in with this new password < /p> < p class = 'ml-2 mb-0 text-danger' > You may change it by clicking on the 'Change Password'
45. ";
46. endif;
47. $body = " < div > < h3 class = 'pl-2' > Reset Your Password < /h3> < p class = 'ml-2' > Enter your email address below to reset your password. < /p> < form action = '".str\_replace('
48. \_ ',' - ',basename($\_SERVER['
49. PHP\_SELF '],'.php '))."'
50. method = 'post'
51. class = 'ml-2' > < div class = 'form-inline form-group' > < label
52. for = 'email' > Email & nbsp; < /label> < input type = 'text'
53. name = 'email'
54. id = 'email'
55. class = 'form-control' / > < input type = 'submit'
56. class = 'btn btn-success'
57. name = 'recovery'
58. value = 'Confirm' / > < /div> < span class = 'text-danger' > ".$pass\_errors['email']." < /span> &nbsp; < /form>";
59. $body. = isset($r\_message) ? $r\_message : '';
60. $body. = "</div>";
61. require\_once($\_SERVER['DOCUMENT\_ROOT'].
62. "/E-Commerce/includes/body.php");
63. require\_once($\_SERVER['DOCUMENT\_ROOT'].
64. "/E-Commerce/includes/footer.php"); ?>

### Form Function

1. <? php
2. function create\_form\_input($name, $type) {
3. $value = false;
4. if (isset($\_POST[$name])): $value = $\_POST[$name];
5. endif;
6. if ($value && get\_magic\_quotes\_gpc()): $value = stripslashes($value);
7. endif;
8. if ($type == 'text' || $type == 'password'): echo "<input type = '".$type.
9. "' name = '".$name.
10. "' id = '".$name.
11. "' ";
12. if ($value): echo "value = '".htmlspecialchars($value).
13. "' ";
14. echo "class = 'form-control'/>";
15. else :echo "class = 'form-control'/>";
16. endif;
17. endif;
18. }

### Header

1. < !DOCTYPE html > < html lang = "en" > < head > < meta charset = "UTF-8" / > < meta name = "viewport"
2. content = "width=device-width,  initial-scale=1" / > < title > <? php $title = isset($page\_title) ? $page\_title : "Knowledge is Power: And It Pays to Know"; ?> <? php echo $title; ?> < /title> < link rel = "stylesheet"
3. href = "/E-Commerce/vendor/bootstrap.min.css"
4. media = "screen" / > < link rel = "stylesheet"
5. href = "/E-Commerce/CSS/style.css"
6. media = "screen" / > < script src = "/E-Commerce/vendor/jquery.min.js" > < /script> < script src = "/E-Commerce/vendor/popper.min.js" > < /script> < script src = "/E-Commerce/vendor/bootstrap.min.js" > < /script> < /head> < body > < div class = "row"
7. id = "wrap" > < div class = "col header text-center bg-light" > < h1 > < a href = "/E-Commerce/index"
8. class = "nounderline" > < code class = "text-info" > Knowledge is Power < /code> < /a> < /h1> < h2 > < code class = "text-warning" > and it pays to know < /code>  < /h2> < /div>  < /div> < div class = "row" > < nav id = "nav"
9. class = "col navbar navbar-expand-sm navbar-dark bg-dark sticky-top" > < a class = "navbar-brand"
10. href = "#" > & nbsp; < code class = "text-warning" > Code Break! < /code> < /a> < button class = "navbar-toggler"
11. type = "button"
12. data - toggle = "collapse"
13. data - target = "#collapsibleNavbar" > < span class = "navbar-toggler-icon" > < /span> < /button> < div class = "collapse navbar-collapse"
14. id = "collapsibleNavbar" > < ul class = "navbar-nav" > <? php $pages = ["Home" => "index", "About" => "about", "Contact" => "contact", "Register" => "register"];
15. $this\_page = basename($\_SERVER['PHP\_SELF'], ".php");
16. foreach($pages as $key => $value): if ($value == $this\_page): echo '<li class = "nav-item active">';
17. else :echo '<li class = "nav-item">';
18. endif;
19. echo '  <a href = "/E-Commerce/'.$value.
20. '" class = "nav-link">';
21. echo $key;
22. echo '  </a>';
23. echo '</li>';
24. endforeach; ?> < /ul> < /div> < /nav> < /div>

### IPN

1. <? php require\_once($\_SERVER['DOCUMENT\_ROOT'].
2. '/E-Commerce/includes/config.inc.php');
3. $request = 'cmd=\_notify-validate';
4. foreach($\_POST as $key => $value): $value = urlencode(stripslashes($value));
5. $request. = "&$key=$value";
6. endforeach;
7. $fp = fsockopen('ssl://www.sandbox.paypal.com', 443, $errno, $errtsr, 30);
8. if ($fp): $header = "POST/cgi-bin/webscrHTTP/1.1\r\n";
9. $header. = "Content-Type:application/x-www-form-urlencoded\r\n";
10. $header. = "Content-Length:".strlen($request).
11. "\r\n";
12. $header. = "Connection: Close\r\n";
13. $header. = "Host: www.sandbox.paypal.com\r\n\r\n";
14. fputs($fp, $header.$request);
15. $from = "admin@sphotonkhan.com";
16. $to = "khan.photon@gmail.com";
17. $subject = "Response";
18. $message = $request;
19. $headers = "From:".$from;
20. if (mail($to, $subject, $message, $headers)): echo "Email sent.";
21. else :echo "Failed to send the email.";
22. endif;
23. while (!feof($fp)): $response = fgets($fp, 1024);
24. if (strcmp($response, "VERIFIED") == 0): if (isset($\_POST['payment\_status']) && $\_POST['payment\_status'] == 'Completed'): if ($\_POST['receiver\_email'] == 'photonkhan@gmail.com'): if (($\_POST['mc\_gross'] == 10.00) && $\_POST['mc\_currency'] == 'USD'): if (!empty($\_POST['txn\_id'])): require(MYSQL);
25. $txn\_id = mysqli\_real\_escape\_string($dbc, $\_POST['txn\_id']);
26. $query = "SELECT id FROM orders WHERE transaction\_id = '$txn\_id'";
27. $result = mysqli\_query($dbc, $query);
28. if (mysqli\_num\_rows($result) == 0): $e = isset($\_POST['email']) ? $\_POST['email'] : '';
29. $query = "SELECT id FROM users where email = '$e'";
30. $result = mysqli\_query($dbc, $query);
31. while ($row = mysqli\_fetch\_array($result)): $uid = $row['id'];
32. endwhile;
33. $status = mysqli\_real\_escape\_string($dbc, $\_POST['payment\_status']);
34. $amount = (float) $\_POST['mc\_gross'];
35. $query = "INSERT INTO orders (user\_id, transaction\_id, payment\_status, payment\_amount)
36. VALUES('$uid', '$txn\_id', '$status', '$amount')
37. ";
38. $result = mysqli\_query($dbc, $query);
39. if (mysqli\_affected\_rows($dbc) == 1): if ($uid > 0): $query = "UPDATE users SET date\_expires = DATE\_ADD(NOW(), INTERVAL 1 YEAR), date\_modified = NOW()
40. WHERE id = '$uid'
41. ";
42. $result = mysqli\_query($dbc, $query);
43. if (mysqli\_affected\_rows($dbc) != 1): trigger\_error("The user's expiration date could not be updated");
44. endif;
45. endif;
46. endif;
47. endif;
48. endif;
49. endif;
50. endif;
51. endif;
52. endif;
53. endwhile;
54. else :trigger\_error("Could not connect to the IPN");
55. endif;

### Login

1. <? php $login\_errors['user\_pass'] = '';
2. $login\_errors['user\_email'] = '';
3. if ($\_SERVER['REQUEST\_METHOD'] == "POST" && isset($\_POST['login'])): if (filter\_var($\_POST['user\_email'], FILTER\_VALIDATE\_EMAIL)): $e = mysqli\_real\_escape\_string($dbc, $\_POST['user\_email']);
4. else :$login\_errors['user\_email'] = "Please enter a valid email address!";
5. endif;
6. if (!empty($\_POST['user\_pass'])): $pattern = "/^(?=.\*\d)(?=.\*[a-zA-Z])(?=.\*[\\_!-\.@#%]).{6,16}$/";
7. if (preg\_match($pattern, $\_POST['user\_pass'])): $p = mysqli\_real\_escape\_string($dbc, $\_POST['user\_pass']);
8. else :$login\_errors['user\_pass'] = "Invalid Password!";
9. endif;
10. else :$login\_errors['user\_pass'] = "Password field cannot be empty!";
11. endif;
12. $err\_login = 0;
13. foreach($login\_errors as $login\_error): if ($login\_error != ''): $err\_login += 1;
14. endif;
15. endforeach;
16. if ($err\_login == 0): $query = "SELECT id, username, type, IF(date\_expires > Now(), true, false), pass
17. FROM users WHERE email = '$e'
18. ";
19. $result = mysqli\_query($dbc, $query);
20. if (mysqli\_num\_rows($result) == 1): $row = mysqli\_fetch\_array($result, MYSQLI\_NUM);
21. if (password\_verify($p, $row[4])): echo "<meta http-equiv='refresh' content='1'>";
22. if ($row[2] == 'admin'): // session\_regenerate\_id(true);
23. $\_SESSION['user\_admin'] = true;
24. endif;
25. $\_SESSION['user\_id'] = $row[0];
26. $\_SESSION['username'] = $row[1]; // setcookie('user\_id', $row[0], time() + (86400 \* 30), "/"); // setcookie('username', $row[1], time() + (86400 \* 30), "/");
27. if ($row[3] == 1): $\_SESSION['user\_not\_expired'] = true;
28. endif;
29. else :$login\_errors['user\_pass'] = "Password do not match";
30. endif;
31. endif;
32. if (mysqli\_num\_rows($result) == 0): $login\_errors['user\_pass'] = "Email Address do not exist!";
33. endif;
34. endif;
35. endif; ?> < div class = "text-success border border-warning p-2" > < h4 > Login < /h4> < form method = "post"
36. action = "<?php echo str\_replace('\_','-',basename($\_SERVER['PHP\_SELF'], '.php')); ?>" > < div class = "form-group" > < label
37. for = "user\_email" > Email < /label> <? php create\_form\_input('user\_email', 'text'); ?> < span class = "text-danger" > <? php echo $login\_errors['user\_email']; ?> < /span> < /div> < div class = "form-group" > < label
38. for = "user\_pass" > Password < /label> <? php create\_form\_input('user\_pass', 'password'); ?> < span class = "text-danger" > <? php echo $login\_errors['user\_pass']; ?> < /span> < /div> < input type = "submit"
39. class = "btn btn-success"
40. name = "login"
41. value = "Confirm" / > < /form> < a href = "forgot-password"
42. class = "nounderline text-info" > < code > Forgot your password ? < /code> < /a> < /div>

### Log Out

1. <? php require($\_SERVER['DOCUMENT\_ROOT'].
2. '/E-Commerce/includes/config.inc.php');
3. if (isset($\_SESSION['username']) && isset($\_SESSION['user\_id'])): if (!empty($\_SESSION['username']) && !empty($\_SESSION['user\_id'])): $\_SESSION = array();
4. if (isset($\_COOKIE[session\_name()])): setcookie(session\_name(), '', time() - 3600, '/');
5. endif;
6. session\_destroy();
7. endif;
8. endif;
9. if (isset($\_COOKIE['username']) && isset($\_COOKIE['user\_id'])): if (!empty($\_COOKIE['username']) && !empty($\_COOKIE['user\_id'])): setcookie('username', ' ', time() - 3600, "/");
10. setcookie('user\_id', ' ', time() - 3600, "/");
11. $home\_url = $\_SERVER['HTTP\_REFERER'];
12. header("Location: ".$home\_url);
13. endif;
14. else :$home\_url = $\_SERVER['HTTP\_REFERER'];
15. header("Location: ".$home\_url);
16. endif;

### MySQL

1. <? php define('DB\_HOST', 'localhost');
2. define('DB\_USER', 'root');
3. define('DB\_PASS', '');
4. define('DB\_NAME', 'ecommerce1');
5. $dbc = mysqli\_connect(DB\_HOST, DB\_USER, DB\_PASS, DB\_NAME) or die("Server Denied");
6. mysqli\_set\_charset($dbc, 'utf8');
8. function escape\_data($data) {
9. global $dbc;
10. return mysqli\_real\_escape\_string(trim($data), $dbc);
11. }

### Page

1. <? php require($\_SERVER['DOCUMENT\_ROOT'].
2. "/E-Commerce/includes/config.inc.php");
3. require(MYSQL);
4. if (isset($\_GET['id']) && filter\_var($\_GET['id'], FILTER\_VALIDATE\_INT, ['min\_range' => 1])): $query = "SELECT title, description, content FROM pages WHERE id = '".$\_GET['id'].
5. "'";
6. $result = mysqli\_query($dbc, $query);
7. if (mysqli\_num\_rows($result) != 1): $page\_title = "Error!";
8. include($\_SERVER['DOCUMENT\_ROOT'].
9. "/E-Commerce/includes/header.php");
10. $body = "<h1 class = 'ml-2 mt-2 text-danger'> This page has been accessed in error </h1>";
11. include($\_SERVER['DOCUMENT\_ROOT'].
12. "/E-Commerce/includes/body.php");
13. include($\_SERVER['DOCUMENT\_ROOT'].
14. "/E-Commerce/includes/footer.php");
15. exit();
16. endif;
17. if (isset($\_SESSION['user\_not\_expired']) && isset($\_SESSION['user\_id'])): while ($row = mysqli\_fetch\_array($result)): $page\_title = $row['title'];
18. include($\_SERVER['DOCUMENT\_ROOT'].
19. "/E-Commerce/includes/header.php");
20. $body = "<h1 class = 'text-primary'> ".$row['title'].
21. " </h1>";
22. $body. = "<h2>".$row['description'].
23. "</h2>";
24. $body. = "<h2>".$row['content'].
25. "</h2>";
26. endwhile;
27. include($\_SERVER['DOCUMENT\_ROOT'].
28. "/E-Commerce/includes/body.php");
29. include($\_SERVER['DOCUMENT\_ROOT'].
30. "/E-Commerce/includes/footer.php");
31. endif;
32. if (!isset($\_SESSION['user\_not\_expired'])): $page\_title = "User Expired";
33. include($\_SERVER['DOCUMENT\_ROOT'].
34. "/E-Commerce/includes/header.php");
35. $body = "<h1 class = 'ml-2 mt-2 text-success'> Thank you for you interest in this content </h1>";
36. $body. = "<p class = 'ml-2 mt-2 text-danger'> Unfortunately you account has been expired. </p>";
37. $body. = "<p class = 'ml-2 mt-2 text-danger'> <a href = 'renew' class = 'nounderline'>Renew Your Account</a></p>";
38. $body. = "<p class = 'ml-2 mt-2 text-danger'>In order to access site content.</p>";
39. include($\_SERVER['DOCUMENT\_ROOT'].
40. "/E-Commerce/includes/body.php");
41. include($\_SERVER['DOCUMENT\_ROOT'].
42. "/E-Commerce/includes/footer.php");
43. exit();
44. endif;
45. if (!isset($\_SESSION['user\_id'])): $page\_title = "User not Logged in";
46. include($\_SERVER['DOCUMENT\_ROOT'].
47. "/E-Commerce/includes/header.php");
48. $body = "<h1 class = 'ml-2 mt-2 text-success'> Thank you for you interest in this content </h1>";
49. $body. = "<p class = 'ml-2 mt-2 text-danger'>You must be logged in as a registered user to view site content. </p>";
50. include($\_SERVER['DOCUMENT\_ROOT'].
51. "/E-Commerce/includes/body.php");
52. include($\_SERVER['DOCUMENT\_ROOT'].
53. "/E-Commerce/includes/footer.php");
54. exit();
55. endif;
56. endif; ?>

### PDF

1. <? php require\_once($\_SERVER['DOCUMENT\_ROOT'].
2. "/E-Commerce/includes/config.inc.php");
3. require(MYSQL);
4. if (isset($\_SESSION['user\_not\_expired']) && isset($\_SESSION['user\_id'])): $page\_title = "PDFs";
5. require\_once($\_SERVER['DOCUMENT\_ROOT'].
6. "/E-Commerce/includes/header.php");
7. $body = "<h3 class = 'pl-2'>PDF Guides</h3>";
8. $query = "SELECT id, tmp\_name, title, description, size, file\_name FROM pdfs ORDER BY date\_created DESC";
9. $result = mysqli\_query($dbc, $query);
10. if (mysqli\_num\_rows($result) > 0): while ($row = mysqli\_fetch\_array($result)): $body. = "<p class = 'text-primary ml-2'><a href = '/E-Commerce/pdf/view-pdf/".$row['id'].
11. "' class = 'nounderline'>".$row['title'].
12. "</a></p>";
13. endwhile;
14. require\_once($\_SERVER['DOCUMENT\_ROOT'].
15. "/E-Commerce/includes/body.php");
16. require\_once($\_SERVER['DOCUMENT\_ROOT'].
17. "/E-Commerce/includes/footer.php");
18. else :$body. = "<h2 class = 'ml-2'>Currently, there are no PDFS to view!</h2>";
19. endif;
20. endif;
21. if (!isset($\_SESSION['user\_not\_expired'])): $page\_title = "User Expired";
22. include($\_SERVER['DOCUMENT\_ROOT'].
23. "/E-Commerce/includes/header.php");
24. $body = "<h1 class = 'ml-2 mt-2 text-success'> Thank you for you interest in this content </h1>";
25. $body. = "<p class = 'ml-2 mt-2 text-danger'> Unfortunately you account has been expired. </p>";
26. $body. = "<p class = 'ml-2 mt-2 text-danger'> <a href = 'renew' class = 'nounderline'>Renew Your Account</a></p>";
27. $body. = "<p class = 'ml-2 mt-2 text-danger'>In order to access site content.</p>";
28. include($\_SERVER['DOCUMENT\_ROOT'].
29. "/E-Commerce/includes/body.php");
30. include($\_SERVER['DOCUMENT\_ROOT'].
31. "/E-Commerce/includes/footer.php");
32. exit();
33. endif;
34. if (!isset($\_SESSION['user\_id'])): $page\_title = "User not Logged in";
35. include($\_SERVER['DOCUMENT\_ROOT'].
36. "/E-Commerce/includes/header.php");
37. $body = "<h1 class = 'ml-2 mt-2 text-success'> Thank you for you interest in this content </h1>";
38. $body. = "<p class = 'ml-2 mt-2 text-danger'>You must be logged in as a registered user to view site content. </p>";
39. include($\_SERVER['DOCUMENT\_ROOT'].
40. "/E-Commerce/includes/body.php");
41. include($\_SERVER['DOCUMENT\_ROOT'].
42. "/E-Commerce/includes/footer.php");
43. exit();
44. endif; ?>

### Register

1. <? php require\_once($\_SERVER['DOCUMENT\_ROOT'].
2. '/E-Commerce/includes/config.inc.php');
3. $page\_title = "Register";
4. require\_once($\_SERVER['DOCUMENT\_ROOT'].
5. '/E-Commerce/includes/header.php');
6. require(MYSQL);
7. require($\_SERVER['DOCUMENT\_ROOT'].
8. '/E-Commerce/includes/form\_functions.inc.php');
9. $reg\_errors['first\_name'] = '';
10. $reg\_errors['last\_name'] = '';
11. $reg\_errors['username'] = '';
12. $reg\_errors['email'] = '';
13. $reg\_errors['pass1'] = '';
14. $reg\_errors['pass2'] = '';
15. if ($\_SERVER['REQUEST\_METHOD'] == 'POST' && isset($\_POST['registration'])): if (empty($\_POST['first\_name'])): $reg\_errors['first\_name'] = "Field cannot be empty";
16. else :$pattern = "/^[a-zA-Z\'.-]{2,20}$/";
17. if (!preg\_match($pattern, $\_POST['first\_name'])): $reg\_errors['first\_name'] = "Invalid First Name";
18. else :$fn = mysqli\_real\_escape\_string($dbc, $\_POST['first\_name']);
19. endif;
20. endif;
21. if (empty($\_POST['last\_name'])): $reg\_errors['last\_name'] = "Field cannot be empty";
22. else :$pattern = "/^[a-zA-Z\'.-]{2,40}$/";
23. if (!preg\_match($pattern, $\_POST['last\_name'])): $reg\_errors['last\_name'] = "Invalid Last Name";
24. else :$ln = mysqli\_real\_escape\_string($dbc, $\_POST['last\_name']);
25. endif;
26. endif;
27. if (empty($\_POST['username'])): $reg\_errors['username'] = "Field cannot be empty";
28. else :$pattern = "/^[a-zA-Z\'.-\\_0-9]{8,30}$/";
29. if (!preg\_match($pattern, $\_POST['username'])): $reg\_errors['username'] = "Minimum 8 characters, avoid @,!,#,+,?";
30. else :$u = mysqli\_real\_escape\_string($dbc, $\_POST['username']);
31. endif;
32. endif;
33. if (empty($\_POST['email'])): $reg\_errors['email'] = "Field cannot be empty";
34. else :if (!filter\_var($\_POST['email'], FILTER\_VALIDATE\_EMAIL)): $reg\_errors['email'] = "Invalid Email Address";
35. else :$e = mysqli\_real\_escape\_string($dbc, $\_POST['email']);
36. endif;
37. endif;
38. if (empty($\_POST['pass1']) || empty($\_POST['pass2'])): if (empty($\_POST['pass1'])): $reg\_errors['pass1'] = "Field cannot be empty";
39. endif;
40. if (empty($\_POST['pass2'])): $reg\_errors['pass2'] = "Field cannot be empty";
41. endif;
42. else :if ($\_POST['pass1'] != $\_POST['pass2']): $reg\_errors['pass2'] = "Passwords do not match";
43. else :$pattern = "/^(?=.\*\d)(?=.\*[a-zA-Z])(?=.\*[\\_!-\.@#%]).{6,16}$/"; if (!preg\_match($pattern, $\_POST['pass1'])): $reg\_errors['pass2'] = "Minimum 6 characters, Needs to have least 1 number and 1 special character";
44. else :$p = password\_hash(mysqli\_real\_escape\_string($dbc, $\_POST['pass1']), PASSWORD\_DEFAULT);
45. endif;
46. endif;
47. endif;
48. $err = 0;
49. foreach($reg\_errors as $error): if ($error != ''): $err += 1;
50. endif;
51. endforeach;
52. $message = false;
53. $email\_taken = false;
54. $user\_taken = false;
55. if ($err == 0): $query = "SELECT username, email FROM users WHERE username = '$u' OR email = '$e'";
56. $result = mysqli\_query($dbc, $query) or die("Select Query Denied");
57. if (mysqli\_num\_rows($result) == 0): $query = "INSERT INTO users(type, username, email, pass, first\_name,
58. last\_name, date\_expires) VALUES('member', '$u', '$e', '$p', '$fn', '$ln', DATE\_ADD(NOW(), INTERVAL 1 MONTH))
59. ";
60. $result = mysqli\_query($dbc, $query) or die("Insert Query Denied");
61. if (mysqli\_affected\_rows($dbc) == 1): $message = true;
62. $from = "khan.photon@gmail.com";
63. $to = $e;
64. $subject = "Registration Confirmation";
65. $message = "Thank you for registering with code break";
66. $headers = "From:".$from;
67. if (mail($to, $subject, $message, $headers)): echo "Email sent.";
68. else :echo "Failed to send the email.";
69. endif;
70. else :trigger\_error("You could not be registered due to a system error. We apologize for the inconvenience.");
71. endif;
72. elseif(mysqli\_num\_rows($result) == 1): while ($row = mysqli\_fetch\_array($result)): if ($row['username'] == $u): $user\_taken = true;
73. endif;
74. if ($row['email'] == $e): $email\_taken = true;
75. endif;
76. endwhile;
77. endif;
78. endif;
79. endif;
80. $body = "<h3 class = 'pl-2'> Register </h3> < p class = 'pl-2 mb-0' > Access to the site 's content is available to registered users at a cost of <strong>1000 TK</strong> per month.</p> < p class = 'pl-2 mb-0 text-primary' > Use the form below to beigin registration process < /p> < p class = 'pl-2 mb-0 text-danger' > < strong > Note: All fields are required < /strong></p > < p class = 'text-success pl-2 mb-0' > After completing this form, you 'll be presented with the opportunity</p> < p class = 'text-success pl-2 mb-0' > to securely pay
81. for your yearly subscription via < kbd > < a href = 'http://www.paypal.com'
82. class = 'nounderline text-white' > PayPal < /a><kbd></p > < button type = 'button'
83. class = 'btn btn-primary ml-3 mt-3'
84. data - toggle = 'modal'
85. data - target = '#registrationModal' > Register < /button> ";
86. if (isset($email\_taken) && isset($user\_taken)): if ($user\_taken == true && $email\_taken == false): $body. = "<div class = 'card ml-3 mt-3' style = 'width:250px'> < div class = 'card-header' > < img src = '/E-Commerce/images/user.png'
87. class = 'card-img-top' / > < h4 class = 'text-warning' > Username Already Taken < /h4> < /div> < div class = 'card-body' > < p class = 'text-secondary' > Please choose other username! < /p> < /div>  < /div>";
88. endif;
89. if ($email\_taken == true && $user\_taken == false): $body. = "<div class = 'card ml-3 mt-3' style = 'width:250px'> < div class = 'card-header' > < img src = '/E-Commerce/images/hdd.png'
90. class = 'card-img-top' / > < h4 class = 'text-warning' > This email address has already been registered < /h4> < /div> < div class = 'card-body' > < p class = 'text-secondary' > Forgot Password ? Click link to the right < /p> < /div>  < /div>";
91. endif;
92. if ($email\_taken == true && $user\_taken == true): $body. = "<div class = 'card ml-3 mt-3' style = 'width:250px'> < div class = 'card-header' > < img src = '/E-Commerce/images/hdd.png'
93. class = 'card-img-top' / > < h4 class = 'text-warning' > This email address has already been registered < /h4> < /div> < div class = 'card-body' > < p class = 'text-secondary' > Forgot Password ? Click link to the right < /p> < /div>  < /div>";
94. endif;
95. endif;
96. if (isset($message)): if ($message == true): $body. = "<div class = 'card ml-3 mt-3' style = 'width:250px'> < div class = 'card-header' > < img src = '/E-Commerce/images/geek.png'
97. class = 'card-img-top' / > < h4 class = 'text-success' > Thanks! < /h4> < /div> < div class = 'card-body' > < p > Thank you
98. for registering at < code > Code Break < /code></p > < form action = 'https://www.sandbox.paypal.com/cgi-bin/webscr'
99. method = 'post'
100. target = '\_top'
101. class = 'mx-auto' > < input type = 'hidden'
102. name = 'cmd'
103. value = '\_s-xclick' > < input type = 'hidden'
104. name = 'hosted\_button\_id'
105. value = 'HXR2K5JUG864L' > < input type = 'image'
106. src = 'https://www.sandbox.paypal.com/en\_US/i/btn/btn\_subscribeCC\_LG.gif'
107. border = '0'
108. name = 'submit'
109. alt = 'PayPal - The safer, easier way to pay online!' > < img alt = ''
110. border = '0'
111. src = 'https://www.sandbox.paypal.com/en\_US/i/scr/pixel.gif'
112. width = '1'
113. height = '1' > < /form> < /div>  < /div>";
114. elseif($email\_taken == true || $user\_taken == true):
115. else :$body. = "<div class = 'card ml-3 mt-3' style = 'width:250px'> < div class = 'card-header' > < img src = '/E-Commerce/images/denied.png'
116. class = 'card-img-top' / > < h4 class = 'text-danger' > Oops!Something went wrong with registeration! < /h4> < /div> < div class = 'card-body' > < p class = 'text-danger' > We are sorry
117. for the inconveniece! < /p> < p class = 'text-info' > Please click the register button again! < /p> < /div>  < /div>";
118. endif;
119. endif;
120. echo "<div class = 'modal fade' id = 'registrationModal'>";
121. echo "  <div class = 'modal-dialog'>";
122. echo "      <div class = 'modal-content'>";
123. echo "      <div class = 'modal-header'>";
124. echo "          <h4 class = 'modal-title'> Registration </h4>";
125. echo "          <button type = 'button' class = 'close' data-dismiss = 'modal'>&times;</button>";
126. echo "      </div>";
127. echo "      <div class = 'modal-body'>";
128. echo '  <form method = "post" action = "'.str\_replace('\_', '-', basename($\_SERVER['PHP\_SELF'], '.php')).
129. '" class = "form-group">';
130. echo '      <div class = "form-group">';
131. echo '          <label for = "first\_name">First Name</label>';
132. create\_form\_input("first\_name", "text");
133. echo '      <span class = "text-danger">'.$reg\_errors['first\_name'].
134. '</span>';
135. echo '</div>';
136. echo '      <div class = "form-group">';
137. echo '          <label for = "last\_name">Last Name</label>';
138. create\_form\_input("last\_name", "text");
139. echo '      <span class = "text-danger">'.$reg\_errors['last\_name'].
140. '</span>';
141. echo '</div>';
142. echo '      <div class = "form-group">';
143. echo '          <label for = "username">Desired Username</label>';
144. create\_form\_input("username", "text");
145. echo '      <span class = "text-danger">'.$reg\_errors['username'].
146. '</span>';
147. echo '</div>';
148. echo '      <div class = "form-group">';
149. echo '          <label for = "email">Email</label>';
150. create\_form\_input("email", "text");
151. echo '      <span class = "text-danger">'.$reg\_errors['email'].
152. '</span>';
153. echo '</div>';
154. echo '      <div class = "form-group">';
155. echo '          <label for = "pass1">Password</label>';
156. create\_form\_input("pass1", "password");
157. echo '      <span class = "text-danger">'.$reg\_errors['pass1'].
158. '</span>';
159. echo '</div>';
160. echo '      <div class = "form-group">';
161. echo '          <label for = "pass2">Confirm Password</label>';
162. create\_form\_input("pass2", "password");
163. echo '      <span class = "text-danger">'.$reg\_errors['pass2'].
164. '</span>';
165. echo '</div>';
166. echo '      <input type = "submit" class = "btn btn-success" id = "submit\_button" name = "registration"';
167. echo '             value = "Next &rarr;"/>';
168. echo '  </form>   ';
169. echo "      </div>";
170. echo "      <div class = 'modal-footer'>";
171. echo "          <button type = 'button' class = 'btn btn-danger' data-dismiss = 'modal'> Close </button>";
172. echo "      </div>";
173. echo "      </div>";
174. echo "  </div>";
175. echo "</div>";
176. require\_once($\_SERVER['DOCUMENT\_ROOT'].
177. "/E-Commerce/includes/body.php");
178. require\_once($\_SERVER['DOCUMENT\_ROOT'].
179. "/E-Commerce/includes/footer.php");

### View PDF

1. <? php require\_once($\_SERVER['DOCUMENT\_ROOT'].
2. '/E-Commerce/includes/config.inc.php');
3. require\_once(MYSQL);
4. if (isset($\_GET['id']) && isset($\_SESSION['user\_not\_expired'])): $i = (int) $\_GET['id'];
5. $query = "SELECT \* FROM pdfs WHERE id = '".$\_GET['id'].
6. "'";
7. $result = mysqli\_query($dbc, $query);
8. if (mysqli\_num\_rows($result) == 1): while ($row = mysqli\_fetch\_array($result)): $page\_title = $row['title'];
9. require\_once($\_SERVER['DOCUMENT\_ROOT'].
10. '/E-Commerce/includes/header.php');
11. $file = $\_SERVER['DOCUMENT\_ROOT'].
12. '/E-Commerce/includes/pdfs/'.$row['file\_name'];
13. if (file\_exists($file) && is\_file($file)): //    header('Content-Type:application/pdf'); //    header('Content-Disposition:inline; filename="'.$row['file\_name'].'"'); //    header('Content-Length: '.filesize($file)); // @readfile($file);
14. $body = "<iframe src= '/E-Commerce/includes/pdfs/".$row['file\_name'].
15. "'
16. width = '100%'
17. style = 'height:100%'
18. '> < /iframe>";
19. require\_once($\_SERVER['DOCUMENT\_ROOT'].
20. "/E-Commerce/includes/body.php");
21. require\_once($\_SERVER['DOCUMENT\_ROOT'].
22. "/E-Commerce/includes/footer.php");
23. endif;
24. endwhile;
25. endif;
26. else :$page\_title = "Error!";
27. require\_once($\_SERVER['DOCUMENT\_ROOT'].
28. '/E-Commerce/includes/header.php');
29. $body = '<h3 class = "pl-2 text-danger"> You do not have the access to this page!</h3> < p class = "pl-2" > Please login as a registered user < /p>';
30. $body. = "<meta http-equiv='refresh' content='1'>";
31. require\_once($\_SERVER['DOCUMENT\_ROOT'].
32. "/E-Commerce/includes/body.php");
33. require\_once($\_SERVER['DOCUMENT\_ROOT'].
34. "/E-Commerce/includes/footer.php");
35. endif;

# Stock Keeping Unit Based Business

Stock Keeping Unit (SKUs), it usually has unique identifier of each item which helps to improve the inventory management. Model View Controller (MVC) pattern is usually used and controller usually handles the behavior and acts accordingly.

**Cart**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **id** | **quantity** | **user\_session\_id** | **product\_type** | **product\_id** | **date\_created** | **date\_modified** |

**Customers**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **id** | **email** | **first\_name** | **last\_name** | **address1** | **adddress2** | **city** | **state** | **zip** | **phone** | **date\_created** |

**General Coffees**

|  |  |  |  |
| --- | --- | --- | --- |
| **id** | **category** | **description** | **image** |

**Non-Coffee Categories**

|  |  |  |  |
| --- | --- | --- | --- |
| **id** | **category** | **description** | **image** |

**Orders**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **id** | **customer\_id** | **total** | **shipping** | **credit\_card\_number** |

**Order Contents**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **id** | **order\_id** | **product\_type** | **product\_id** | **quantity** | **price\_per** |

**Sales**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **id** | **product\_type** | **product\_id** | **price** | **start\_date** | **end\_date** |

**Size**

|  |  |
| --- | --- |
| **id** | **size** |

**Transactions**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **id** | **order\_id** | **type** | **amount** | **response\_code** | **response\_reason** | **transaction\_id** | **response** | **date\_created** |

**Wish Lists**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **id** | **quantity** | **user\_session\_id** | **product\_type** | **product\_id** | **date\_created** | **date\_modified** |

## Hypertext Access

**RewriteEngine On**

# Turn on the rewriting engine

**RewriteRule ^pet-care/?$** pet\_care\_info\_01\_02\_2008.php **[NC,L]**

RewriteRule 🡪 Tells Apache that this like refers to a single RewriteRule.

^/pet-care/?$ 🡪 The "pattern"

pet\_care\_info\_01\_02\_2003.php 🡪 The "substitution".

[NC,L] - "Flags" 🡪 tells Apache how to apply the rule

"NC" 🡪 tells Apache that this rule should be case-insensitive

"L" 🡪 tells Apache not to process any more rules if this one is used.

RewriteRule **^products/([0-9]+)/?$**

show\_product.php?product\_id=**$1 [NC,L]**

show\_product.php?product\_id=2

products/2

This allows us to "back-reference" (refer back to) that section of the URL in the following "substitution" section. The "$1" in the substitution tells Apache to put whatever matched the earlier bracketed pattern into the URL at this point. We can have lots of backreferences, and they are numbered in the order they appear.

RewriteRule **^browse/([A-Za-z\+]+)/([A-Za-z\+]+)/([0-9]+)?$**

browse.php?type=$1&category=$2&id=$3

browse.php?type=coffee&category=Kona&id=3

browse/coffee/Kona/3

## Enforcing SSL

To do this, we must have SSL enabled.

RewriteCond **%{HTTPS} off**

RewriteRule **^(checkout\.php | billing\.php | final\.php | admin/(.\*))$**

https://%{HTTP\_HOST}/$1 [R=301,L]

The first line checks for the condition where HTTPS is off: %{something}. It refers to a server environmental variable. Then the rule attempts to match checkout.php, billing.php, final.php, or admin/anything. If the match is made the URL is written to <https://hostname/$1>

R=301,L 🡪 This should be a permanent redirection type (server code 301)

## Prepared Statements

In database-based applications, many times the same query will be executed repeatedly using just slightly different parameters. A prepared statement separates the static content from the dynamic values, using placeholders for latter. The variables bound to the placeholders must have their appropriate values when the mysqli\_stmt\_execute() function is called. We do not need to take any extra stesp to prevent SQL Injection attacks because the prepared statements already prevent those simple by separating the values from the rest of the query.

Using Login as an example:

**//Query**

$q =SELECT id, username, type, IF(date\_expires >= NOW(), true, false)

FROM users WHERE (email=? and password=?);

**//Prepare Statement**

$stmt = mysqli\_prepare($dbc, $q);

**//Check Errors**

if(!$stmt) echo mysqli\_stmt\_error($stmt)

**//Bind Parameters**

mysqli\_stmt\_bind\_param($stmt, ‘ss’, $email, $pass);

The first argument is the statement representative variable. The next is an indicator of the formats of the various placeholders, using one symbol for each placeholder.

Bound Value Types

|  |  |
| --- | --- |
| **Letter** | **Represents** |
| d | Decimal |
| i | Integer |
| b | Blob (binary data) |
| s | All other types |

**//Execute the statement**

mysqli\_stmt\_execute($stmt)

**//UPDATE, INSERT**

if(mysqli\_stmt\_affected\_rows($stmt) == 1):

endif;

**//SELECT**

mysqli\_stmt\_store\_result($stmt);

if(mysqli\_stmt\_num\_rows($stmt) >= 1):

endif;

**//Close**

mysqli\_stmt\_close($stmt);

All the previous code demonstrates inbound prepared statements, in which the value used in a query come from variables. We can also use outbound prepared statements – with or without inbound parameters, in which case the query’s results are assigned to variables.

**Summary**

* Connect

mysqli\_connect($host, $user, $pass, $db);

$query = “SELECT id, username, IF(date\_expire > = NOW(), true, false), email FROM users where (email = ? and password = ?)

* PREPARE

$stmt = mysqli\_prepare($dbc, $query);

if(!$stmt) echo mysqli\_stmt\_error($stmt)

$email = $\_POST[‘email’];

$pass = password\_hash($\_POST[‘pass’]);

* BIND

mysqli\_stmt\_bind\_param($stmt, ‘ss’, $email, $pass);

* EXECUTE

mysqli\_stmt\_execute($stmt);

* INSERT, UPDATE

if(mysqli\_stmt\_affected\_rows($stmt) == 1):

endif;

* SELECT

mysqli\_stmt\_store\_result($stmt)

* MULTIPLE DATA

if(mysqli\_stmt\_num\_rows($stmt) > = 1):

endif;

* SINGLE DATA

if(mysqli\_stmt\_num\_rows($stmt) == 1):

mysqli\_stmt\_bind\_result($stmt, $id, $username, $type, $expired);

mysqli\_stmt\_fetch($stmt);

endif;

* CLOSE

mysqli\_stmt\_close($stmt);

## Stored Procedure

* Improved Security
* Better Performance
* Cleaner model-controller separation
* Increased application portability

Stored procedures are created using SQL command

**DELIMITER $$**

**CREATE PROCEDURE name(arguments)**

**BEGIN**

**CODE**

**END$$**

Ex:CREATE PROCEDURE do\_this(age INT, name VARCHAR(20)….)

$result = mysqli\_query(“CALL do\_this”);

mysqli\_fetch\_array($result);

**DELIMITER $$**

**CREATE PROCEDURE get\_description(unique\_id INT)**

**BEGIN**

**SELECT first\_name, last\_name, gender**

**FROM form WHERE id = unique\_id;**

**END$$**

This are all stored in the Routine part of the MYSQL. For this we need the CREATE ROUTINE, ALTER ROUTINE and EXECUTE permissions.

**Selecting every coffee product**

SELECT gc.description, gc.image, CONCAT("C", sc.id) AS Sku,

CONCAT\_WS("-", s.size, sc.caf\_decaf, sc.ground\_whole,

sc.price) AS name, sc.stock

FROM specific\_coffees AS sc

INNER JOIN sizes AS s

ON s.id = sc.size\_id

INNER JOIN general\_coffees AS gc

ON gc.id = sc.general\_coffee\_id

WHERE general\_coffee\_id = <some\_category\_id> AND stock>0

ORDER by name ASC;

**Selecting every non-coffee product**

SELECT ncc.description AS g\_description,

ncc.image AS g\_image, CONCAT("O", ncp.id) AS sku,

ncp.name, ncp.description, ncp.image, ncp.price, ncp.stock

FROM non\_coffee\_products AS ncp

INNER JOIN non\_coffee\_categories AS ncc

ON ncc.id = ncp.non\_coffee\_category\_id

WHERE non\_coffee\_category\_id = 3

ORDER by date\_created DESC

# Google Chart API

**Sample Data**

$chartdata = [

"cols" => [

["id" => "", "label" => "Reason", "pattern" => "", "type" => "string"],

["id" => "", "label" => "Percentage", "pattern" => "", "type" => "number"]

],

"rows" => [

["c" => [["v" => 45, "f" => "Social Meet up"], ["v" => 45,

"f" => " Social Meet up "]]],

["c" => [["v" => 30, "f" => "Coffee"], ["v" => 30, "f" => "Coffee"]]],

["c" => [["v" => 8, "f" => "Yoga"], ["v" => 8, "f" => "Yoga"]]],

["c" => [["v" => 8, "f" => "Read Books"], ["v" => 8,

"f" => "Read Books "]]],

["c" => [["v" => 9, "f" => "Study"], ["v" => 9, "f" => "Study"]]],

]

];

//For Charting Statistics

google.charts.load('current', {packages:['corechart']});

google.charts.setOnLoadCallback(drawChart);

function drawChart(){

$.getJSON('/cafe/includes/settings.php?chartdata=true', function(data){

var chartData = new google.visualization.DataTable(data);

$.getJSON('/cafe/includes/settings.php?chartoptions=true', function(data){

var chart = new google.visualization.PieChart(document.getElementById('piechart\_3d'));

chart.draw(chartData, data);

});

});

}

**PHP 🡪** $query = SELECT id, username, type,

IF(data\_expires >= NOW(), true, false)

FROM users WHERE (email = ? and password = ?)

**PHP 🡪** $stmt = mysqli\_prepare($dbc, $query);

**PHP 🡪** if(!$stmt): echo mysqli\_stmt\_error($stmt);

**PHP 🡪** mysqli\_stmt\_bind\_param($stmt, ‘ss’, $email, $pass)

s🡪 all other types

b🡪 blob

i🡪 integer

d🡪 decimal

**PHP 🡪** if(mysqli\_stmt\_affected\_rows($stmt) == 1)

**PHP 🡪** mysqli\_stmt\_store\_result($stmt)

**PHP 🡪** if(mysqli\_stmt\_num\_rows($stmt) >= 1)

**PHP 🡪** mysqli\_stmt\_close($stmt)

**PHP 🡪** mysqli\_stmt\_bind\_result($stmt, $id, $username, $type, $expired)

**PHP 🡪** mysqli\_stmt\_fetch($stmt)

**PHP 🡪** mysqli\_query(“CALL get\_non\_coffee\_products($id)”);

**DELIMITER $$**

**CREATE PROCEDURE name (arguments)**

**BEGIN**

**END$$**

**PHP 🡪** mysqli\_query(“CALL get\_non\_coffee\_products($id)”);

# Prepare Statement

DELIMITER$$

CREATE PROCEDURE select\_person\_info(IN type VARCHAR(10), IN select\_id INT)

BEGIN

IF(type = ‘all’) THEN

PREPARE statement FROM “

SELECT \* FROM person\_info

“;

EXECUTE STATEMENT;

ELSEIF(type = ‘specific’) THEN

PREPARE statement FROM “

SELECT \* FROM person\_info WHERE id = @p1;

“;

SET @p1 = select\_id;

EXECUTE statement USING @p1

END IF;

END$$

**PHP Prepare Statement**

try{

$handler = new PDO(“mysql:host=’host\_name’;dbname=’database’”, $user, $pass);

$query = “CALL select\_person\_info(:type,:select\_id)”;

$stmt = $handler 🡪 prepare($query);

$handler🡪beginTransaction();

$stmt->setFetchMode(PDO∷FETCH\_ASSOC);

$result = $stmt🡪execute([“:type”=>$type, “:select\_id”=>$select\_id]);

$handler🡪commit();

}catch(PDOException $e){

$handler🡪rollback();

echo $e🡪getMessage();

}

# Full Text Index

CREATE FULLTEXT\_INDEX idx\_person\_info ON person\_info(name, email);

SELECT (name, email) FROM person\_info

WHERE MATCH(name, email) AGAINST (+wham +bam IN BOOLEAN MODE)

ORDER BY MATCH(name, email) AGAINST(+what +bam IN BOOLEAN MODE) DESC;

**SHOW VARIABLES LIKE ‘ft\_min\_word\_len’;**