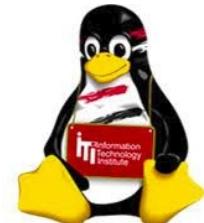




Prepared by:  
Noha Shehab

Updated and presented by:  
**Kareem Saeed**

Teaching Assistant  
Information Technology Institute (ITI)

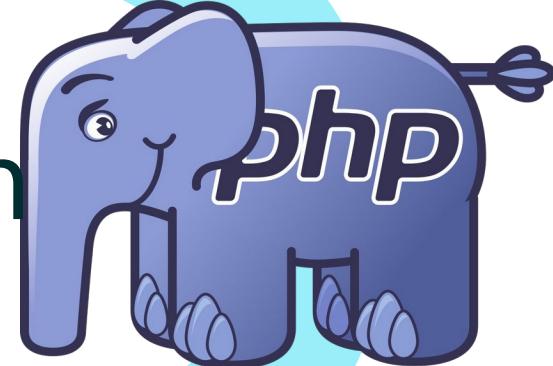


# Agenda

- String Manipulation
- Regular expressions
- Uploading files
- Sessions
- Cookies



# String Manipulation



# Popular functions

- Trim white spaces
  - **trim(), ltrim(), rtrim() == chop()**
- Applying some more sophisticated formatting
  - **printf(), sprintf() , nl2br()**
- Format the string into upper or lower cases
  - **Strtoupper(), strtolower(), ucfirst() and ucwords().**
- Formatting string to storage
  - **addslashes() and stripslashes()**



# Popular functions

- Join, Split strings
  - **Explode, implode, join**
- Comparing strings
  - **strcmp(), strcasecmp()**
- Search string:
  - **strstr (), strchr(), and striistr()**
- String position:
  - **strpos (), strrpos () and stripos**
- String replacement:
  - **str\_replace(), md5(), ord() str\_repeat()**



# Trim function examples

```
$text = "\t\tThese are a few words :) ... ";
var_dump($text);
$trimmed = trim($text);
var_dump($trimmed);
```

```
StringManipulation.php:14:string 'These are a few words :) ...' (Length=32)
```

```
StringManipulation.php:16:string 'These are a few words :) ...' (Length=28)
```

```
$text = "\t\tThese are a few words :) ... ";
var_dump($text);
$trimmed = trim($text, "\tThe");
var_dump($trimmed);
```

```
StringManipulation.php:14:string 'These are a few words :) ...' (Length=32)
```

```
StringManipulation.php:16:string 'se are a few words :) ...' (Length=27)
```



# Sophisticated formatting

- nl2br(): Returns string with <br /> or <br> inserted before all newlines (\r\n, \n\r, \n and \r)

```
$str="You came  
to me  
in that hour  
of need";  
echo($str."<br>");  
echo "<h2> After applying the function </h2>";  
echo(nl2br($str));
```

You came to me in that hour of need

## After applying the function

You came  
to me  
in that hour  
of need

need to  
word first in



# Sophisticated formatting

- `Sprintf()`

```
$num = 5;  
$location = 'tree';  
$format = 'There are %d monkeys in the %s';  
echo sprintf($format, $num, $location);
```

There are 5 monkeys in the tree

- `Printf("pattern",$string)`

```
printf($format, $num, $location);
```

There are 5 monkeys in the tree



# String into upper or lower cases

- ucfirst ():
  - Capitalizes first character of string if it is alphabetic.
- Ucwords:
  - Capitalizes first character of each word in the string that begins with an alphabetic character.

```
$string= "welcome to iti";
echo strtoupper($string). "<br>" ;
echo strtolower($string). "<br>" ;
echo ucfirst($string). "<br>" ;
echo ucwords($string). "<br>" ;
```

WELCOME TO ITI  
welcome to iti  
Welcome to iti  
Welcome To Iti



# Formatting string to storage

- Some strings can have quotation marks single or double, slashes and null char so you need to escape these chars, some dbms doesn't accept special chars, you can overcome this using **addslashes()**, **stripslashes()**.

```
$str = "What's your name?";
$newString=addslashes($str);
echo $newString . "<br>";
/// so it can be interpreted in the data
#stripslashes
echo stripslashes($newString) . "<br>";
```

What's your name?  
What\\'s your name?  
What's your name?



# Joining and splitting a string

- `Implode`: joins the array elements into one string, separate them with a delimiter if it is specified.

```
$InputArray = array('OS','Application','Track');
// Join without separator
print_r(implode($InputArray));
echo "<br>";
// Join with separator
print_r(implode("-",$InputArray));
echo "<br>";
print_r(join("_",$InputArray));
echo "<br>";
```

OSApplicationTrack

OS-Application-Track

OS\_Application\_Track



# Joining and splitting a string

- `explode()`: This function takes a string input and splits it into pieces on a specified separator string.
  - The pieces are returned in an array .
  - You can limit the number of pieces with the optional limit parameter

```
$str="I love coffee so much";
$arrstr=explode(" ",$str);

var_dump($arrstr);
$arrstr1=explode(" ",$str,2);
var_dump($arrstr1);  
array (size=5)
  0 => string 'I' (length=1)
  1 => string 'love' (length=4)
  2 => string 'coffee' (length=6)
  3 => string 'so' (length=2)
  4 => string 'much' (length=4)
  
array (size=2)
  0 => string 'I' (length=1)
  1 => string 'love coffee so much' (length=19)
```



# Joining and splitting a string

- Splitting string using substr(\$str, offset, limit):substr — Return part of a string.

```
$phptxt="PHP is simple";
echo substr($phptxt,1); // HP is simple
```

```
echo substr($phptxt,1,5); // "HP is"
```

- Can accept negative offsets

```
echo substr($phptxt,-2); // "le"
```



# Comparing strings

- Strcmp:

```
$var1 = "Hello";
$var2 = "hello";
if (strcmp($var1, $var2) !== 0) {
    echo '$var1 is not equal to $var2 in
a case sensitive string comparison';
}
```

- Strcasecmp

```
$var1 = "Hello";
$var2 = "hello";
if (strcasecmp($var1, $var2) !== 0) {
    echo '$var1 is equal to $var2 in
an incase sensitive string comparision';
}
```



# String length, Searching

- Strlen():

```
$str="Welcome to php";
var_dump(strlen($str)); //int 14
```

- **strstr() == strchr()** can be used to find a string or character match within a longer string.

```
$email = 'name@example.com';
$domain = strstr($email, '@');
echo $domain."<br>";
```

- If the string is found, the function returns with a string started from the pattern needed to be found to the end of the string

@example.com



# String hashing functions

- md5()

*Calculate the md5 hash of a string*

```
$string = 'Hello World!';  
$testt= md5($string); ed076287532e86365e841e92bfc50d8c  
echo($testt)."<br>";
```

- ord():Convert the first byte of a string to a value between 0 And 255 (ASCII code)

```
echo(ord("Noha"))."<br>"; // 78
```

- Str\_repeat,shuffle

```
echo str_repeat("iti ", 5)."<br>";  
$str = 'abcdef'; iti iti iti iti iti  
echo str_shuffle($str)."<br>"; befdac
```



# String replace functions,

- str\_replace(),

```
$vowels = array("a", "e", "i", "o", "u", "A", "E", "I", "O", "U");
$onlyconsonants = str_replace($vowels, "", "Hello World of PHP");
echo $onlyconsonants."<br>";
```

Hll Wrld f PHP

- substr\_replace (string)

```
$input = array('A: XXX', 'B: XXX', 'C: XXX');
$input=substr_replace($input, 'YYY', 3, 3);
var_dump($input);
echo implode(';', $input);
```

A: YYY; B: YYY; C: YYY



# Regular expressions

- PHP offers functions specific to two sets of regular expression functions, each corresponding to a certain type of regular expression. You can use any of them based on your comfort.
  - POSIX Regular Expressions
  - PERL Style Regular Expressions



# Regular expressions (POSIX)

- A regular expression is a way of describing a pattern in a piece of text. (POSIX Syntax)
- Wild characters:
  - . (**DOT**): matches single character.
    - Ex .at will match cat rat hat ... or #at
  - To match a character class, we use [az]
    - Ex [a z]at will match letters only as hat rat cat  
NOT #at or \$at



# Regular expressions

- Common patterns
  - Get all words that contain any vowel: [aeiou]
  - [azA Z]
  - [^az] Not contain
  - [az]\* can be repeated zero or more times.
  - [az]+ can be repeated one or more times.
  - ^[AZ] beginning with
  - [AZ]\$ Ending by



# PHP and regular expressions

- preg\_match (): function can be used to parse string using regular expression.
- To validate strings, like emails, passwords, user\_names and others.
- Returns with the string if it matches the pattern specified, else returns with false.

```
// example:  
$email='nshehab@iti.gov.eg';  
$pattern="/^([a-zA-Z0-9\+\_\-\ ]+)(\.[a-zA-Z0-9\+\_\-\ ]+)*@[a-zA-Z0-9\-\ ]+\.\+[a-zA-Z]{2,6}\$/ix";  
if(preg_match($pattern,$email)){  
    echo "<br>". 'well formed';  
}else{  
    echo "<br>". 'not well formed';  
}
```

well formed



# PHP and regular expressions

- preg\_match\_all("pattern",\$string,\$outputArray):
  - Searches \$string for all matches to the regular expression given in pattern and puts them in \$outputArray.

```
$str = "The rain in SPAIN falls mainly on the plains.";  
      #pattern inside // , i for ignoring case  
$pattern = "/ain/i";  
if(preg_match_all($pattern, $str, $matches)) {  
    print_r($matches);  
}  
Array ( [0] => Array ( [0] => ain [1] => AIN [2] => ain [3] => ain ) )
```



# PHP and regular expressions

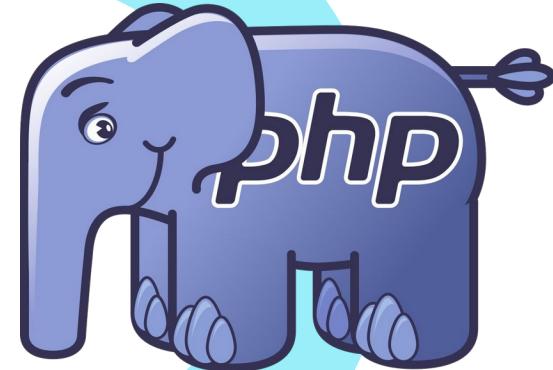
- Filtervar():

```
$email="nohashehab.iti@gmail.com";
if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
    $emailErr = "Invalid email format";
} else{
    echo "<br>".'Checked by php functions and well formed';
}
```

Checked by php functions and well formed



# File uploading



# File uploading to PHP server

- You can control uploading files to your website via setting some parameters in php.ini
  - You need to restrict/ allow uploading files like
    - `file_uploads = On`
    - `max_file_size` or what like it



# HTML Part

- **HTML Part:** Files are uploaded to the servers using POST method only.

```
<form action="uploadingfiles.php" method="POST"
      enctype="multipart/form-data">
    <h1>Please choose your file </h1>
    <label>File </label>
    <input type="file" name="file" />
    <input type="hidden" name="MAX_FILE_SIZE" value="1000000"/>
    <input type="text" name="opensource"/>
    <input type="submit"/>
</form>
```

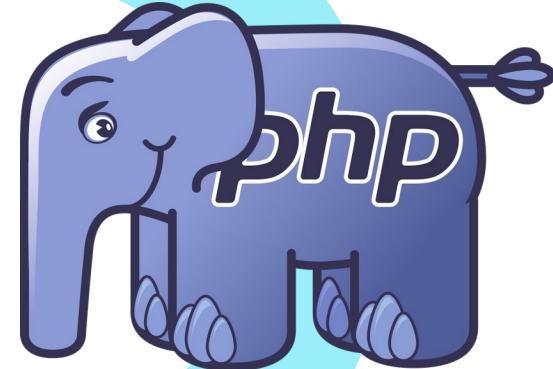


# PHP Part

```
if(isset($_FILES['file'])){
    $errors= array();
    // var_dump($_FILES);
    $file_name = $_FILES['file']['name'];
    $file_size =$_FILES['file']['size'];
    $file_tmp =$_FILES['file']['tmp_name'];
    $file_type=$_FILES['file']['type'];
    // get file extension
    $ext=explode('.',$file_name);
    $file_ext=strtolower(end($ext));
    // or
    $ext= pathinfo($file_name)["extension"];
    $extensions= array("jpeg","jpg","png","pdf","doc","txt","csv");
    if(in_array($file_ext,$extensions)== false){
        $errors[]="extension not allowed, please choose a JPEG or PNG
file.";
    }
    if($file_size > 2097152){
        $errors[]='File size must be excately 2 MB';
    }
    if(empty($errors)==true){
        move_uploaded_file($file_tmp,"files/".$file_name);
        echo "Success";
    }else{
        print_r($errors);
    }
}
```

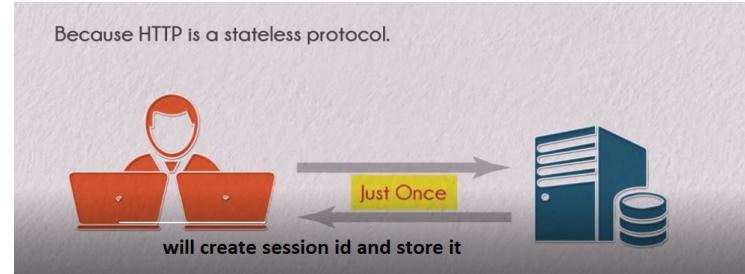


# Sessions and cookies



# Http Protocol

- HTTP
  - HTTP is a stateless protocol . This means that the protocol has no built in way of maintaining state between two transactions.
  - When a user requests one page, followed by another, HTTP does not provide a way for you to tell that both requests came from the same user.



# Sessions

- Sessions
  - The idea of session control is to be able to track a user during a single session on a website.
  - If you can do this, you can easily support logging in a user and showing content according to his authorization level or personal preferences.



# Session control in PHP

- Sessions in PHP are driven by a unique session ID , a cryptographically random number.
- This session ID is generated by PHP and stored on the server side for the lifetime of a session.
- It can be either stored on a user's computer in a cookie or passed along through URLs.



# Session control in PHP

- The basic steps of using sessions are
  - Starting a session
  - Registering session variables
  - Using session variables
  - De-registering variables and destroying the session



# Starting sessions in PHP

- `session_start()`
  - It's essential to call `session_start()` at the start of all your scripts that use sessions.
  - If this function is not called, anything stored in the session will not be available to this script.
- you can begin a session automatically by setting PHP to start one automatically by setting **`session.auto_start`** option in your `php.ini` file.



# Starting sessions in PHP

```
#sent file

# 1- start session
session_start();
#PHP Session ID for each user , cookie
echo "Welcome to the server";

# use $_SESSION to store variables
$_SESSION["username"] = "Kareem";
$_SESSION["course"] = "PHP";
$_SESSION["msg"] = "Good morning";
```

```
session_start();
var_dump($_SESSION);
// remove the session variable
unset($_SESSION["msg"]);
session_destroy();
```



# Starting sessions in PHP

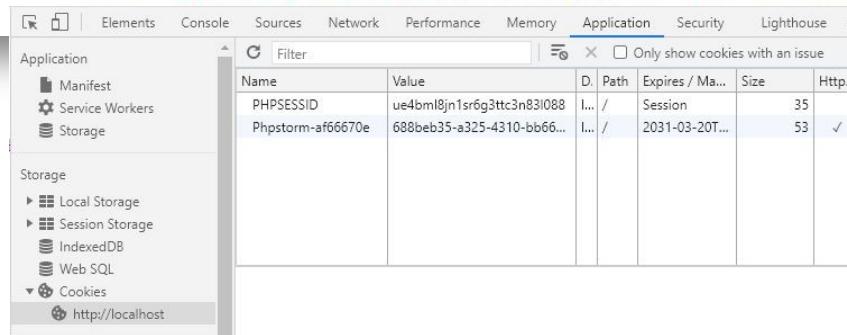
- When you are finished with a session, you should first unset all the variables
  - `$_SESSION = array();`
- Call `session_destroy()` to clean up the session ID.



# Session and cookies

- Session is a place on server to store info about user.
- And cookie is a place on the #user machine that can hold info to identify the user.  
`var_dump($_COOKIE);`
- Session\_Id is stored in the browser in a cookie...

```
array (size=2)
  'Phpstorm-af66670e' => string '688beb35-a325-4310-bb66-09d0ec706578' (Length=36)
  'PHPSESSID' => string 'ue4bml8jn1sr6g3ttc3n83l088' (Length=26)
```



The screenshot shows the Chrome DevTools Application tab with the Cookies section selected. It displays two cookies:

Name	Value	D.	Path	Expires / Ma...	Size	Http...
PHPSESSID	ue4bml8jn1sr6g3ttc3n83l088	I...	/	Session	35	
Phpstorm-af66670e	688beb35-a325-4310-bb66...	I...	/	2031-03-20T...	53	✓

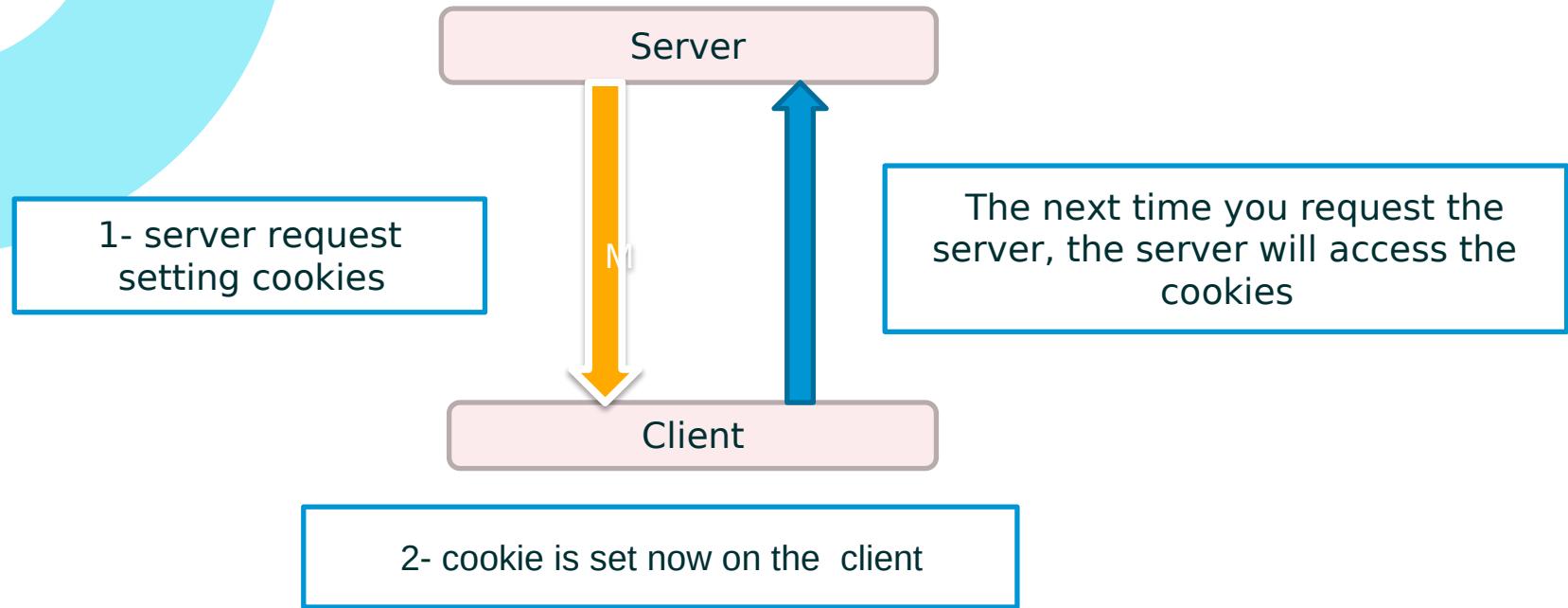


# Cookies

- A cookie is a small piece of information that scripts can store on a client side machine.
- Cookie life cycle
  - There are no cookies when the browser connects to server for first time
  - When the request is made to the PHP script, the script makes a call to the setcookie() function
  - This causes a Set-Cookie HTTP header to be sent in the response that contains the name and value of the cookie to be set.

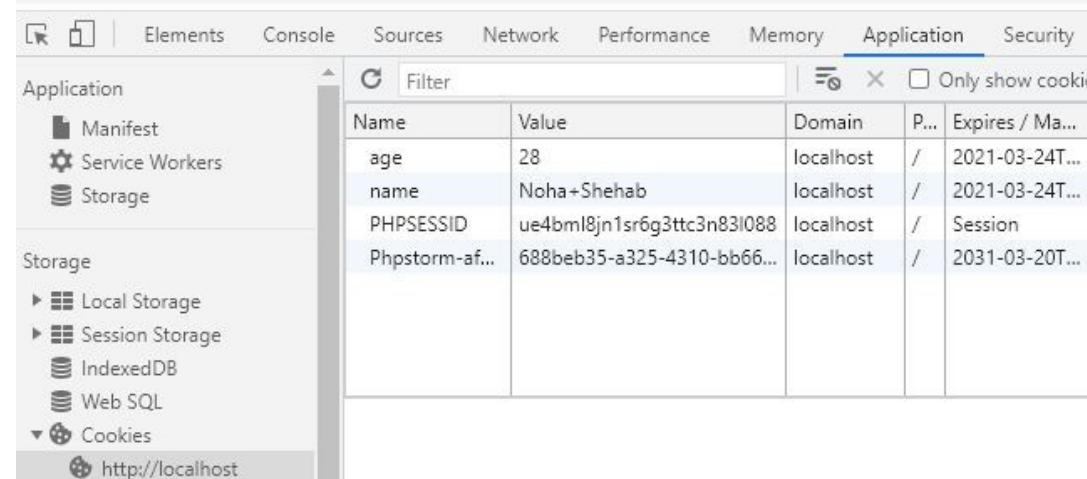


# Cookies



# Cookies

```
#      #one cookie has one value
#1- set cookies
setcookie("name", "Kareem", time()+3600, "/", "", 0);
setcookie("age", "28", time()+3600, "/", "", 0);
var_dump($_COOKIE);
```



The screenshot shows the Chrome DevTools interface with the Application tab selected. On the left, there's a sidebar with sections for Manifest, Service Workers, Storage, and Cookies. Under Cookies, it shows a list of stored cookies for the domain `http://localhost`. The table has columns for Name, Value, Domain, Path, and Expires / Max-Age.

Name	Value	Domain	P...	Expires / Ma...
age	28	localhost	/	2021-03-24T...
name	Noha+Shehab	localhost	/	2021-03-24T...
PHPSESSID	ue4bml8jn1sr6g3ttc3n83l088	localhost	/	Session
Phpstorm-af...	688beb35-a325-4310-bb66...	localhost	/	2031-03-20T...



# Cookies

```
#3- get cookies
if (isset($_COOKIE["name"])){
    echo "welcome ".$_COOKIE["name"]. "<br />";
    setcookie( "name", "", time()- 60, "/", "", 0);

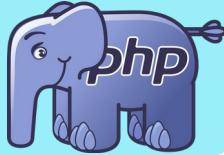
}else{
    echo 'no name cookie here'. "<br />";
}

if (isset($_COOKIE["age"])){
    echo "Your age is ".$_COOKIE["age"]. "<br />";
    setcookie( "age", "", time()- 60, "/", "", 0);

}else{
    echo 'no age cookie here'. "<br />";
}
```



# Lab 03



- 1-Implement the form in php
- 2-Apply the validation on the email field on the php side using 2 ways
- 3- Room number should be a drop down that contains 'Application1, Application2 and cloud)
- 4-Upload a profile picture and ensure that is a photo



A screenshot of a web browser window titled "A Web Page". The address bar shows "http://". The page header includes links for Home, Products, Users, Manual Order, Checks, and a user icon labeled "Admin". The main content area is titled "Add User". It contains the following fields:

Name	<input type="text"/>
Email	<input type="text"/>
Password	<input type="text"/>
Confirm Password	<input type="text"/>
Room No.	<input type="text"/>
Ext.	<input type="text"/>
Profile picture	<input type="text"/> <a href="#">Browse</a>

At the bottom are two buttons: "Save" and "Reset".





# Lab 03



- Once the user submit his info, you should store it in a file called users.
- Create a login page with username and password ((check valid login data from the file))
- Once the user logged in, start a session and display a welcome message to him.

**Bonus: add validation to the password field,**

- a.Only 8 chars
- b.Doesn't allow special chars -only underscore allowed
- c. Doesn't accept Capital characters

A Web Page

http://

Cafeteria

Email

Password

Login

[Forgot Password?](#)

A screenshot of a web browser window titled "A Web Page". The address bar shows "http://". The main content area has a title "Cafeteria". It contains two input fields: one for "Email" and one for "Password". Below the password field is a "Login" button. At the bottom of the form is a link "Forgot Password?".



# Thanks ^^

Kareem Saeed  
[kareemmorsy30@gmail.com](mailto:kareemmorsy30@gmail.com)