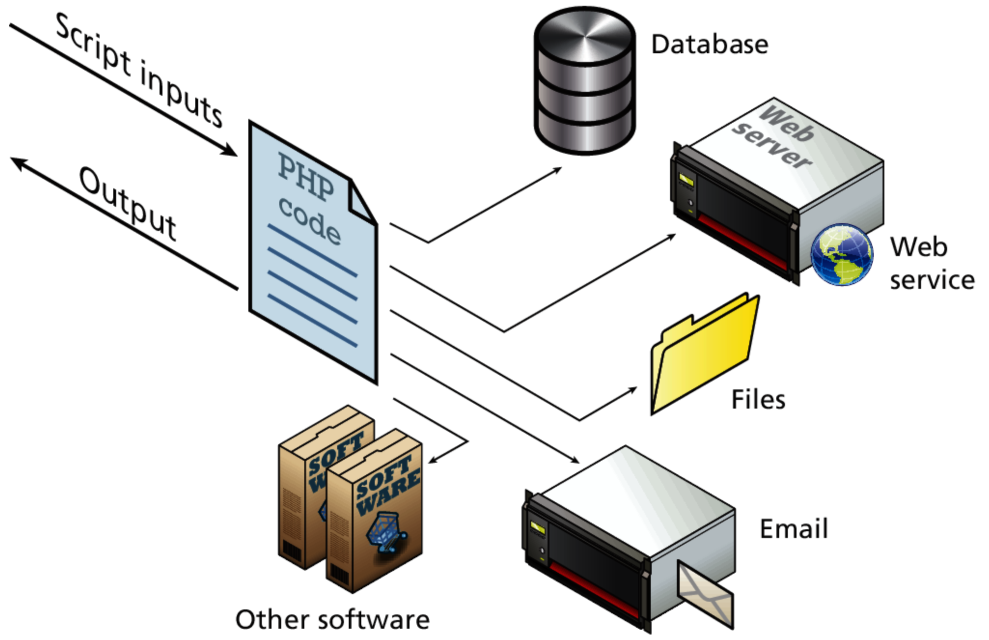
Intro to PHP & MySQL

2020/21 COMP3322 Modern Technologies on WWW

Contents

Server-side Scripting



- “Server-side scripting is a technique used in web development which involves **employing scripts** on a **web server** which produce a **response customized** for each client's request to the website.” – from Wikipedia.
- Customized means **dynamically generating content**.

Common Server-side Technologies

- PHP
- Python
  - Django
- Ruby
  - Ruby on Rails
- ASP.NET
- Node.js
- Perl

https://www.similartech.com/categories

History of PHP development

- PHP is an **open source technology** and runs on most operating systems and with most Web servers.
- It takes most of its syntax from C, Java, and Perl.
- PHP was written in the C programming language by Rasmus Lerdorf in 1994.
  - For managing his person information. For this reason, PHP originally stood for "**Personal Home Page**".
- Rasmus released PHP 1.0 in 1995; he extended it to work with web forms and databases.
- A development team began to form and PHP 2 was released in late 1997.
- The acronym was formally changed to **PHP: HyperText Preprocessor** since then.
- PHP 3 was released in 1998 and PHP 4 was released in 2000.
- PHP 5 was released in 2004 and the latest PHP version is 7, which was released in 2015.

A Quick Tour of PHP

PHP: Hypertext Preprocessor

- PHP, like JavaScript, is a **dynamically typed** language.
- It uses classes and functions in a way consistent with other object-oriented languages such as C++, C#, and Java.
- The syntax for loops, conditionals, and assignment is identical to JavaScript.
- Differs when you get to functions, classes, and in how you define variables.

PHP Tags

- The most important fact about PHP is that the programming code can be **embedded directly within** an HTML file.
- A PHP file will usually have the extension **.php**
- Programming code must be contained within
  - an opening **<?php** tag and
  - a matching closing **?>** tag
- Any code outside the tags is **echoed directly** out to the client
- On servers with shorthand support, a PHP script can start with **<?** and end with **?>**

PHP Tags

```
<?php
$user = "Tony";
?>
<!DOCTYPE>
<html>
<head>
<title>Example 1</title>
</head>
<body>
  <h1>Welcome <?php echo $user; ?></h1>
  <p>
    Current server time is <?php
      echo "<b>";
      echo date("H:i:s");
      echo "</b>";
    ?>
  </p>
</body>
</html>
```

```
<!DOCTYPE>
<html>
<head>
<title>Example 1</title>
</head>
<body>
  <h1>Welcome Tony</h1>
  <p>
    Current server time is <b>09:38:54</b>
  </p>
</body>
</html>
```

PHP Comments

```php
<?php

  # single-line comment

  /*
    This is a multiline comment.
    They are a good way to document functions
    or complicated blocks of code
  */

  $artist = readDatabase(); // end-of-line comment

?>
```

Variables

- Variables in PHP are **loosely typed** in that a variable can be assigned different data types over time.
  - **Similar to JavaScript**
- To declare a variable you must **preface the variable** name with the dollar (**$**) symbol.
  - $count = 42;
- A variable name must start with a letter or the underscore character.
- A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and _)
- Variable names are **case-sensitive**.

Variable Scope

- PHP has three different variable scopes:
  - Local scope
  - Global scope
  - Static scope
- **Local Scope**
  - A variable declared **in a function** can be referenced solely in that function.

```php
<?php
   $x = 4;

   function assignx () {
      $x = 10;
      print $x;  //$x is 10
   }

   assignx();
   print $x;  //$x is 4
?>
```

Variable Scope

- Global Scope
  - A variable defined in the main script (**outside a function**) has a GLOBAL SCOPE and can **only be accessed** outside a function.
  - PHP does allow variables with global scope to be accessed within a function using the **global** keyword

```php
<?php
  $a = 1;
  $b = 2;

  function Sum()
  {
    global $a, $b;
    $b = $a + $b;  //$b = 3
  }

  Sum();
  echo $b;  //3
?>
```

Variable Scope

```
<?php
  function myTest() {
    static $x = 0;
    echo $x;
    $x++;
  }

  myTest();  //0
  myTest();  //1
  myTest();  //2
?>
```

- Static Scope
  - When a function is completed, all of its variables are **deleted**.
  - A static variable exists **only in** a local function scope, but it **does not lose** its value when program execution leaves this scope.
  - A static variable is **initialized only in first call** of the function.

Writing Output

- To output something that will be seen by the browser, you can use the echo() or print() function.
  - **echo("hello");** or **echo "hello";** OR
  - **print("hello");** or **print "hello";**
- Output variables
  - **echo $name;**
- Another alternative is using the printf() function.
  - Like the C programming language; also have the variations like sprintf() and fprintf().
  - **printf("<h1> %s </h1>\n", $title);** https://www.w3schools.com/php/func_string_printf.asp

Writing Output

```
$course = array(
  "code" => "COMP3322",
  "title" => "Modern Tech on WWW",
  "sem" => 2,
  "class" => "B",
  "teacher" => array("last" => "Tam",
     "first" => "Anthony"));
```

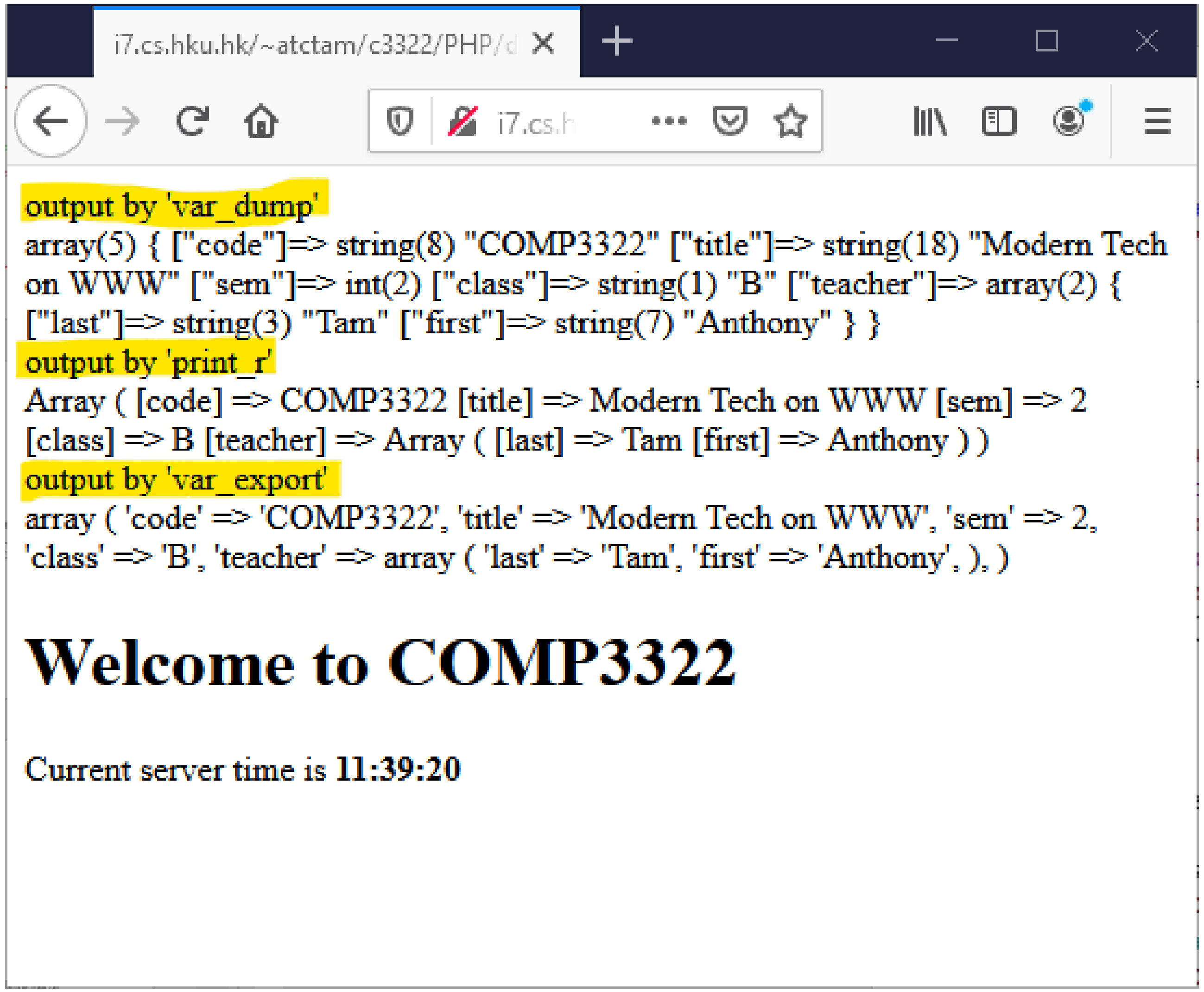
- **Debugging**
  - **var_dump()**, **var_export()**, and **print_r()** are functions that you can use to check values.
  - var_dump() shows **the values and their types** of a variable. Arrays and objects are explored recursively with values indented to show structure.
  - print_r() only shows **the value** in a human-readable format.
  - var_export() like the above two, but it returns the information in a **parsable** string representation.

```
var_dump($course);
array(5) {
  ["code"]=>
  string(8) "COMP3322"
  ["title"]=>
  string(18) "Modern Tech on WWW"
  ["sem"]=>
  int(1)
  ["class"]=>
  string(1) "A"
  ["teacher"]=>
  array(2) {
    ["last"]=>
    string(3) "Tam"
    ["first"]=>
    string(7) "Anthony"
  }
}
```

```
print_r($course);
Array
(
    [code] => COMP3322
    [title] => Modern Tech on WWW
    [sem] => 1
    [class] => A
    [teacher] => Array
        (
            [last] => Tam
            [first] => Anthony
        )

)
```

```
<?php
  $course = array(
  "code" => "COMP3322",
  "title" => "Modern Tech on WWW",
  "sem" => 2,
  "class" => "B",
  "teacher" => array("last" => "Tam",
     "first" => "Anthony"));
  echo "output by 'var_dump'<br>";
  var_dump($course);
  echo "<br>output by 'print_r'<br>";
  print_r($course);
  echo "<br>output by 'var_export'<br>";
  var_export($course);
 ?>
<!DOCTYPE>
<html>
<body>
  <h1>Welcome to <?php echo $course['code']; ?></h1>
  <p>
    Current server time is <?php
      echo "<b>";
      echo date("H:i:s");
      echo "</b>";
    ?>
  </p>
</body>
</html>
```

https://i.cs.hku.hk/~atctam/c3322/PHP/debug.php



```
output by 'var_dump'
array(5) { ["code"]=> string(8) "COMP3322" ["title"]=> string(18) "Modern Tech on WWW" ["sem"]=> int(2) ["class"]=> string(1) "B" ["teacher"]=> array(2) { ["last"]=> string(3) "Tam" ["first"]=> string(7) "Anthony" } }
output by 'print_r'
Array ( [code] => COMP3322 [title] => Modern Tech on WWW [sem] => 2 [class] => B [teacher] => Array ( [last] => Tam [first] => Anthony ) )
output by 'var_export'
array ( 'code' => 'COMP3322', 'title' => 'Modern Tech on WWW', 'sem' => 2, 'class' => 'B', 'teacher' => array ( 'last' => 'Tam', 'first' => 'Anthony', ), )
```

Welcome to COMP3322

Current server time is **11:39:20**

Data Types

Data Type	Description
boolean	A logical true or false value
integer	Whole numbers Max. size is platform-dependent, but at least 32-bit.
float	Decimal numbers Again platform-dependent; usually, in 64 bit IEEE format.
string	A sequence of characters (8 bits) enclosed in single or double quotes.
Array	An array in PHP is actually an ordered map. It supports numeric array, associative array, and multi-dimensional array.
Object	Instances of programmer-defined classes.
Null	NULL is the only possible value of type null.

Case Sensitivity

- Case sensitive
  - variables
  - constants
  - array keys
  - class properties
- Case insensitive
  - functions
  - class contructors/methods
  - keywords and constructs (e.g., if, else, echo, etc.)

Constants

```
define("DB_HOST", "localhost");
define("DB_NAME", "StudentDB");
define("USERNAME", "c3322");
define("PASSWORD", "ew#@rtycd");

$db = mysqli_connect(DB_HOST, USERNAME, PASSWORD, DB_NAME);
```

- Define the constant via the **define**() function
- Once a constant is defined, it can be referenced **without** using the $ symbol.

String

- A string can be any text inside quotes. You can use **single** or **double** quotes.
- String Concatenation
  - Strings can easily be appended together using the concatenate operator, which is the period (.) symbol.
    - Alert! JavaScript uses the plus (+) symbol.
  - Example:
    - $username = "World";
    - echo "Hello ". $username;
    - Will Output “Hello World”

String

- Difference between single quote and double quote strings.
- **Single quotes** are used to denote a “literal string”.
  - The system does **not** attempt to **parse** special characters or variables within the single quote string.
- You can add special characters (e.g., \n, \t) and variables in **double quote** string. The system understands.
- Example:

```
$username = "World";
echo "Hello  $username";
Will Output “Hello World”
```

```
$username = "World";
echo 'Hello  $username';
Will Output “Hello $username”
```

Arrays

- Defining an array
  - `$days = array();`
  - This declares an empty array.
- You can initialize it with a comma-delimited list of values using either of two following syntaxes:
  - `$days = array("Mon","Tue","Wed","Thu","Fri");`
  - `$days = ["Mon","Tue","Wed","Thu","Fri"];`
- You can also declare each subsequent element in the array individually:
  - `$days = array();`
  - `$days[0] = "Mon";`
  - `$days[1] = "Tue";`
  - `$days[] = "Wed";`

Arrays

- In most programming languages array keys are limited to integers, start at 0, and go up by 1.
- In PHP, array keys must be **either integers or strings** and **need not** be sequential.
- If you don't explicitly define the keys, they are 0,1,…
- For numeric indexes, you can skip some indexes.
  - `$menu[0] = "appetizer";`
  - `$menu[2] = "soup";`
  - `$menu[4] = "main course";`
  - `$menu[8] = "dessert";`
  - `print_r($menu);`

```
//Array([0] => appetizer [2] => soup [4] => main course [8] => dessert)
```

Arrays

- Associative Arrays
  - `$record = array("name" => "Tony Stark", "number" => "3015123456", "age" => 20, "email" => "tonystark@hku.hk");`
  - `$record = ["name" => "Tony Stark", "number" => "3015123456", "age" => 20, "email" => "tonystark@hku.hk"];`
- To loop through and print all the values of an associative array, we could use a foreach loop

```
foreach ($record as $x => $x_value) {
  echo "Key=" . $x . ", Value=" . $x_value;
}
```

https://www.w3schools.com/php/php_superglobals.asp

Superglobal Variables

- **Superglobal**
  - Several predefined variables in PHP can always be accessible, regardless of scope.
- Commonly used superglobal variables are:
  - $_GET
    - An associative array containing name/value pairs sent from the client with the get method
  - $_POST
    - An associative array containing name/value pairs sent from the client with the post method
  - $_COOKIE
    - An associative array containing cookie variables and values
  - $_SESSION
    - An associative array containing session variables and values
  - $_REQUEST
    - An associative array contains the contents of $_GET, $_POST and $_COOKIE
  - $_SERVER
    - An associative array contains information about request headers, paths, and script locations

Include Files

- PHP provides four different statements for including files, as shown below.
  - **include** "somefile.php";
  - **include_once** "somefile.php";
  - **require** "somefile.php";
  - **require_once** "somefile.php";
- The include and require statements are identical, except upon failure
  - With include, a warning is displayed and then execution continues. With require, an error is displayed and execution stops.

The Scope of Include Files

- Include files are the equivalent of **copying and pasting**.
- Variables defined within an include file will have the scope of the line on which the include occurs.
- Any variables available at that line in the calling file will be available within the called file.
- If the include occurs inside a function, then all of the code contained in the called file will behave as though it had been defined inside that function.

User Defined Functions in PHP

- A user-defined function declaration starts with the word function.

```
function functionName($arg1,$arg2,.....,$argX) {
    code to be executed;
}
```

- A function name must start with a letter or an underscore. Function names are NOT case-sensitive.
- Functions need not be defined before they are referenced.
- All functions in PHP have the global scope
  - They can be called outside a function even if they were defined inside another function.

User-Defined Functions

- Function parameters
  - These parameters work like variables inside your function; in principle, they are of dynamic type.
  - Since PHP 7, it is possible to declare types for the function parameters.
    - http://php.net/manual/en/functions.arguments.php#functions.arguments.type-declaration
- PHP supports passing arguments by value (the default), passing by reference, and default argument values.
  - Pass by reference: function myFunction(&$arg) { . . . . }

Using JSON in PHP

- PHP has some **built-in** functions to handle JSON.
- Objects and arrays in PHP can be converted into JSON string by using:

```
$json_str = json_encode($php_obj);

$json_str = json_encode($php_arr);
```

- Converting a JSON string into a PHP object or array by using:

```
$anArray = json_decode($json_str, true);

$anObject = json_decode($json_str);
```

When TRUE, returned objects will be converted into associative arrays.

- In the event of a failure to decode, json_last_error() can be used to determine the exact nature of the error.

decode()

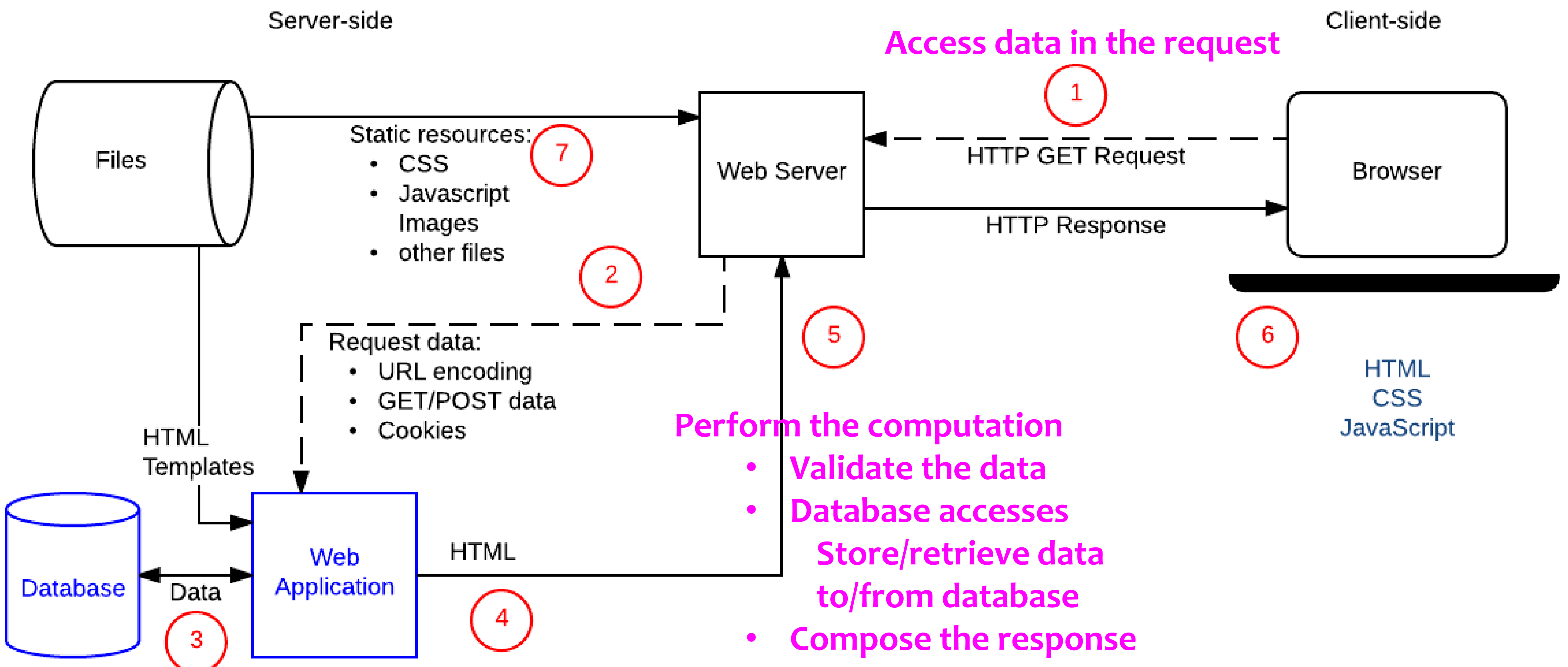
```php
<?php

$json_str = '{"library":{"DVD":[{"id":"1","title":"Breakfast at
Tiffany\'s","format":"Movie","genre":"Classic"},
{"id":"2","title":"Contact","format":"Movie","genre":"Science
fiction"}]}}';

$anArray = json_decode($json_str, true);

if (json_last_error() == JSON_ERROR_NONE) {
  echo $anArray["library"]["DVD"][0]["title"]; //-> Breakfast at
Tiffany's
}

$anObject = json_decode($json_str);

if (json_last_error() == JSON_ERROR_NONE) {
  echo $anObject->library->DVD[1]->title;      //-> Contact
}


?>
```

encode()

```php
<?php

$json_str =
'{"library":{"DVD":[{"id":"1","title":"Breakfast at
Tiffany\'s","format":"Movie","genre":"Classic"},
{"id":"2","title":"Contact","format":"Movie","genre":
"Science fiction"}]}}';

$anObject = json_decode($json_str);

$anObject->library->DVD[1]->title = "Avengers";
$anObject->library->DVD[1]->genre = "Action";

$new_str = json_encode($anObject);

echo $new_str;        //-> {"library":{"DVD":[{"id":"1","title":"Breakfast at
                      Tiffany's","format":"Movie","genre":"Classic"},{"id":"2","title":"Av
?>                    engers","format":"Movie","genre":"Action"}]}}
```

Common Server-side Scripting Scenarios

Common Actions



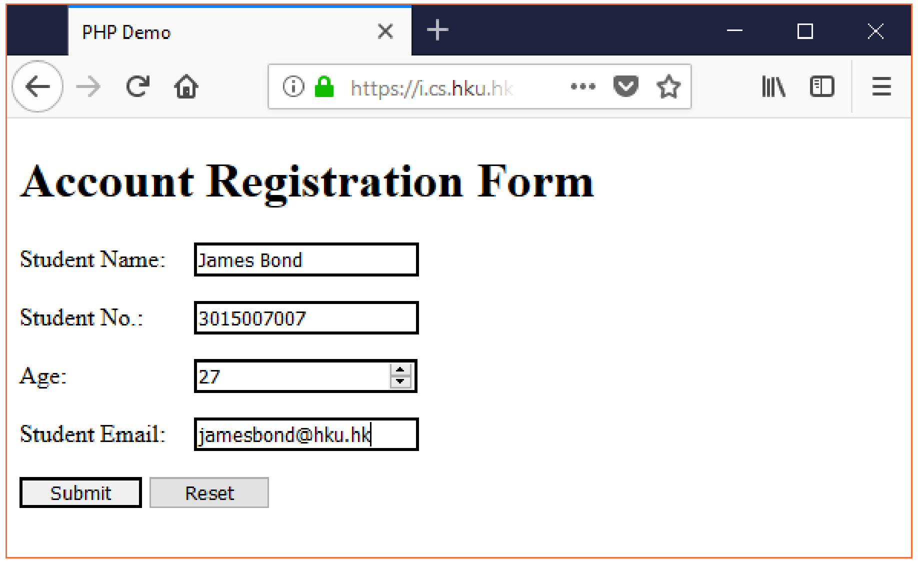
Access data in the request

Perform the computation
- Validate the data
- Database accesses
  Store/retrieve data to/from database
- Compose the response

Keep state information

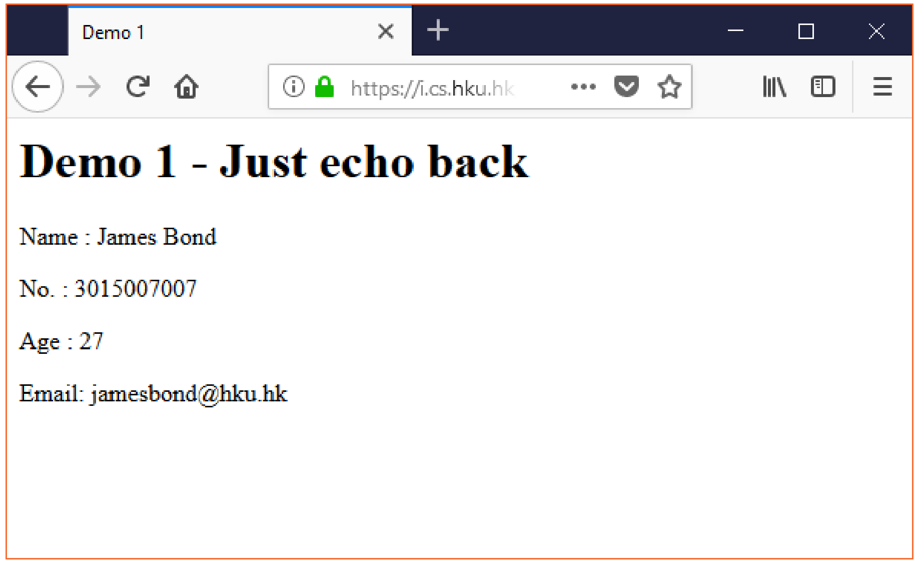
Demo 1 – Just Echo Back

```
<body>
 <h1>Account Registration Form</h1>
 <form id="RegForm" action="view1.php" method="get">
  <p>
   <label for="name">Student Name:</label>
   <input type="text" id="name" name="name" maxlength="50" required>
  </p>
    :
    :

<body>
 <h1>Account Registration Form</h1>
 <form id="RegForm" action="view2.php" method="post">
  <p>
   <label for="name">Student Name:</label>
   <input type="text" id="name" name="name" maxlength="50" required>
  </p>
    :
    :
```



Account Registration Form

Student Name:	James Bond
Student No.:	3015007007
Age:	27
Student Email:	jamesbond@hku.hk



Demo 1 - Just echo back

Name : James Bond

No. : 3015007007

Age : 27

Email: jamesbond@hku.hk

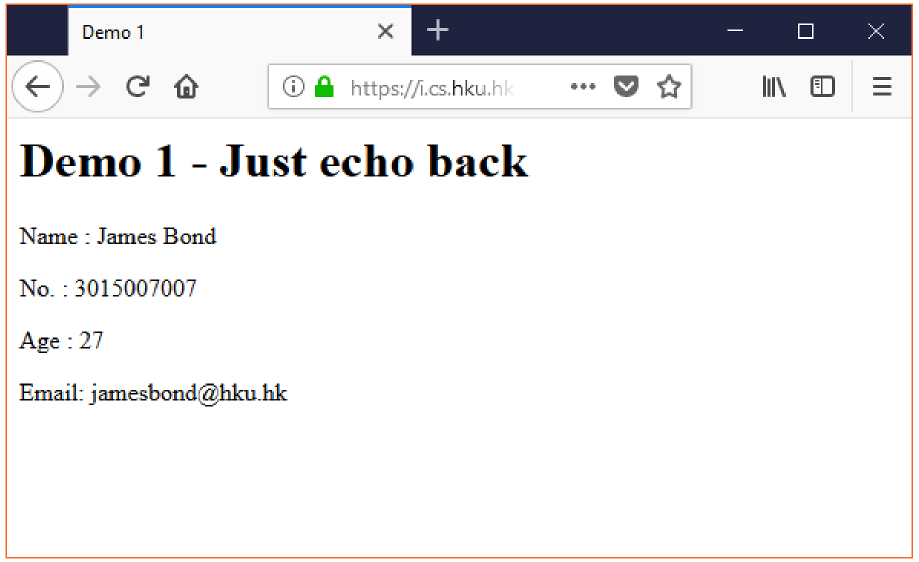
Demo 1 – Just Echo Back

view1.php http://i7.cs.hku.hk/~atctam/c3322/PHP/form-php1.html

```
<?php
   echo "<p>Name : ".$_GET['name']."</p>";
   echo "<p>No.  : ".$_GET['number']."</p>";
   echo "<p>Age  : ".$_GET['age']."</p>";
   echo "<p>Email: ".$_GET['email']."</p>";
 ?>
```

view2.php http://i7.cs.hku.hk/~atctam/c3322/PHP/form-php2.html

```
<?php
   echo "<p>Name : ".$_POST['name']."</p>";
   echo "<p>No.  : ".$_POST['number']."</p>";
   echo "<p>Age  : ".$_POST['age']."</p>";
   echo "<p>Email: ".$_POST['email']."</p>";
 ?>
```



Demo 1

https://i.cs.hku.hk

Demo 1 - Just echo back

Name : James Bond

No. : 3015007007

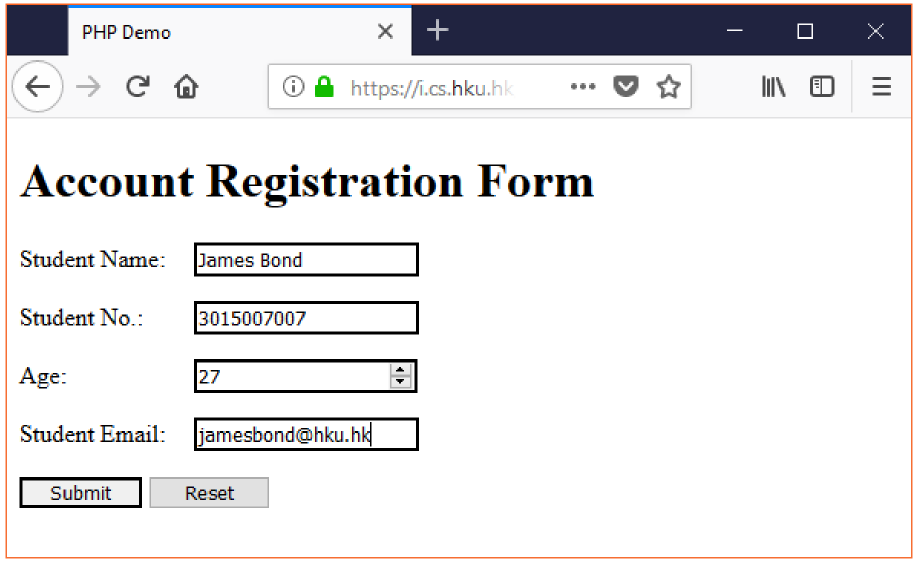
Age : 27

Email: jamesbond@hku.hk

```
<body>
  <h1>Demo 1 - Just echo back</h1>

  <p>Name : James Bond</p><p>No.  :
3015007007</p><p>Age  : 27</p><p>Email:
jamesbond@hku.hk</p>
</body>
```

Demo 2 – List All Records

```
<body>
 <h1>Account Registration Form</h1>
 <form id="RegForm" action="view3.php" method="post">
  <p>
   <label for="name">Student Name:</label>
   <input type="text" id="name" name="name" maxlength="50" required>
  </p>
    :
    :
```



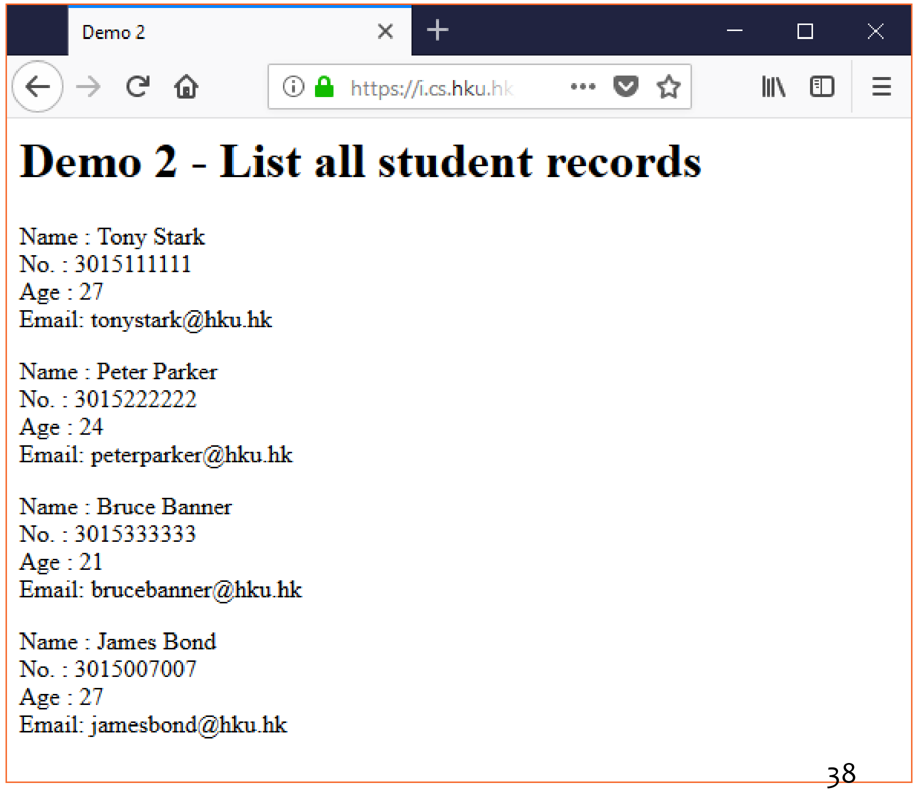
Account Registration Form

Student Name: James Bond

Student No.: 3015007007

Age: 27

Student Email: jamesbond@hku.hk



Demo 2 - List all student records

Name : Tony Stark
No. : 3015111111
Age : 27
Email: tonystark@hku.hk

Name : Peter Parker
No. : 3015222222
Age : 24
Email: peterparker@hku.hk

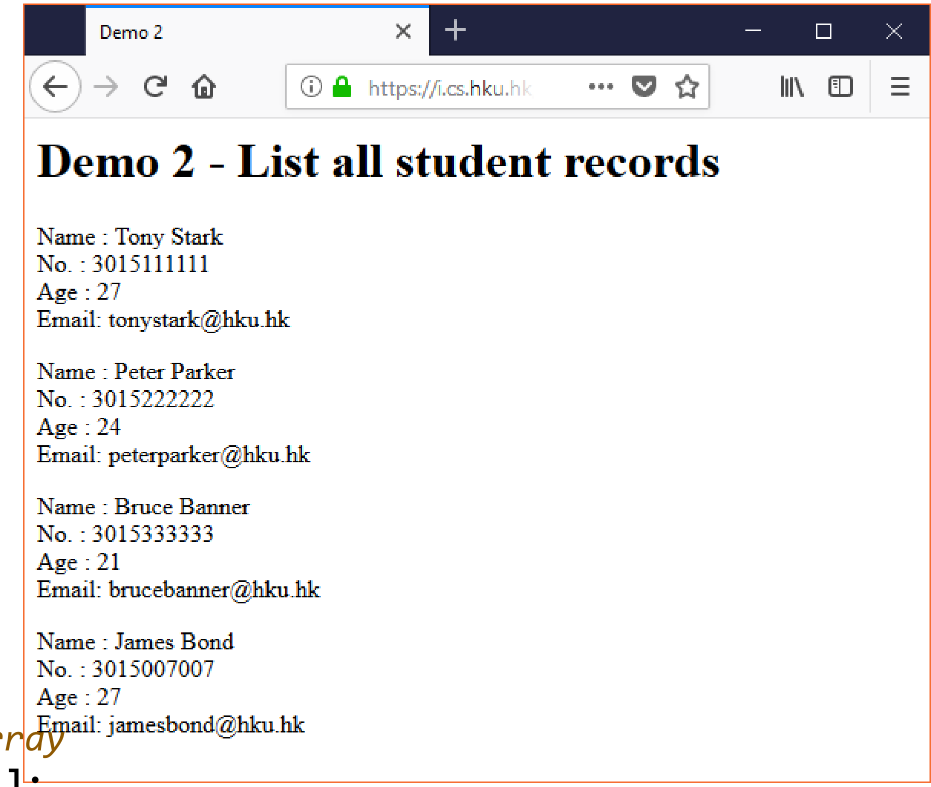
Name : Bruce Banner
No. : 3015333333
Age : 21
Email: brucebanner@hku.hk

Name : James Bond
No. : 3015007007
Age : 27
Email: jamesbond@hku.hk

http://i7.cs.hku.hk/~atctam/c3322/PHP/form-php3.html

Demo 2 – List All Records

```php
<?php
  #Manually create some dummy student records
  $record['name']="Tony Stark";
  $record['number']="3015111111";
  $record['age']=27;
  $record['email']="tonystark@hku.hk";
  $Std_record[0]=$record;
  $record['name']="Peter Parker";
  $record['number']="3015222222";
  $record['age']=24;
  $record['email']="peterparker@hku.hk";
  $Std_record[1]=$record;
  $record['name']="Bruce Banner";
  $record['number']="3015333333";
  $record['age']=21;
  $record['email']="brucebanner@hku.hk";
  $Std_record[2]=$record;

  #Add the new record to the array
  $record['name']=$_POST['name'];
  $record['number']=$_POST['number'];
  $record['age']=intval($_POST['age']);
  $record['email']=$_POST['email'];
  $Std_record[3]=$record;

  for ($i=0; $i < count($Std_record); $i++) {
    echo "<p>Name : ".$Std_record[$i]['name']."<br>";
    echo    "No.  : ".$Std_record[$i]['number']."<br>";
    echo    "Age  : ".$Std_record[$i]['age']."<br>";
    echo    "Email: ".$Std_record[$i]['email']."</p>";
  }

?>
```



Demo 2 - List all student records

Name : Tony Stark
No. : 3015111111
Age : 27
Email: tonystark@hku.hk

Name : Peter Parker
No. : 3015222222
Age : 24
Email: peterparker@hku.hk

Name : Bruce Banner
No. : 3015333333
Age : 21
Email: brucebanner@hku.hk

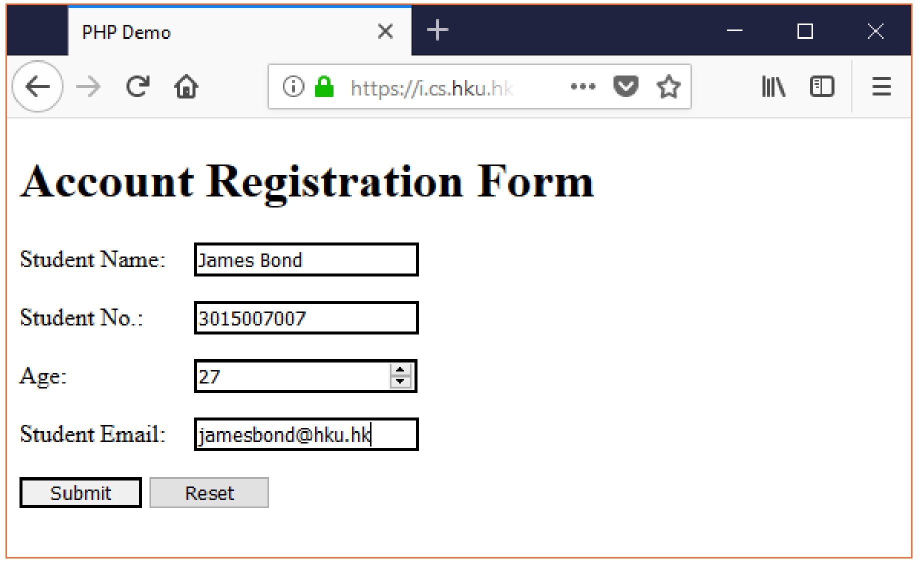
Name : James Bond
No. : 3015007007
Age : 27
Email: jamesbond@hku.hk

Intro to MySQL

Demo 4 – Retrieve Records From DB

```
<body>
 <h1>Account Registration Form</h1>
 <form id="RegForm" action="view5.php" method="post">
  <p>
   <label for="name">Student Name:</label>
   <input type="text" id="name" name="name" maxlength="50" required>
  </p>
     :
     :
```

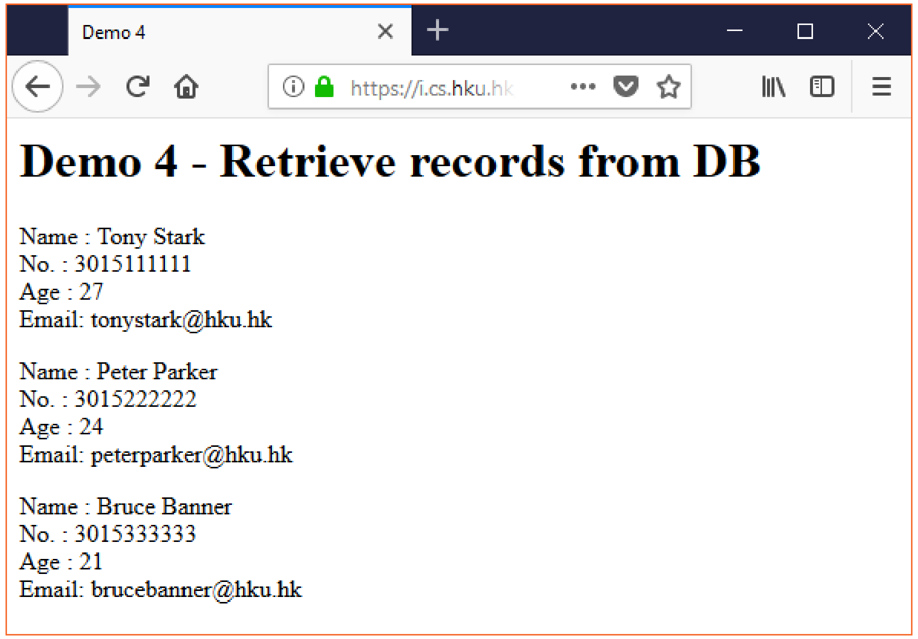
http://i7.cs.hku.hk/~atctam/c3322/PHP/form-php5.html



Account Registration Form

Student Name:	James Bond
Student No.:	3015007007
Age:	27
Student Email:	jamesbond@hku.hk

Submit Reset



Demo 4 - Retrieve records from DB

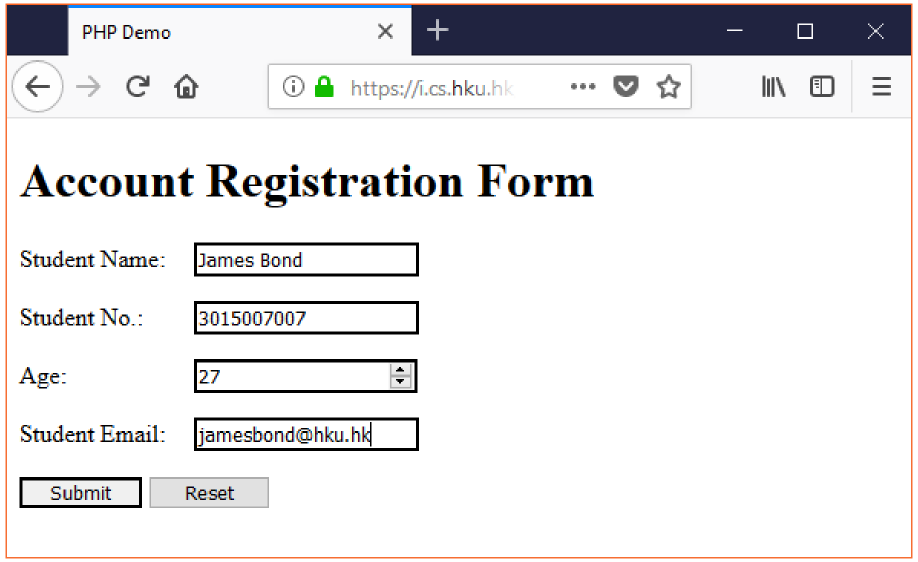
Name : Tony Stark
No. : 3015111111
Age : 27
Email: tonystark@hku.hk

Name : Peter Parker
No. : 3015222222
Age : 24
Email: peterparker@hku.hk

Name : Bruce Banner
No. : 3015333333
Age : 21
Email: brucebanner@hku.hk

Demo 4 – Retrieve Records From DB

```php
<?php
   #Connect to sophia
   $db_conn=mysqli_connect("sophia.cs.hku.hk", "c3322a", “XXXX", "c3322a")
     or die("Connection Error! ".mysqli_connect_error());

   #Retrieve all records from DB
   $query="SELECT * FROM stdRecord";
   $Std_record=mysqli_query($db_conn, $query)
     or die("Query Error!".mysqli_error($db_conn));

   #Display the records
   if (mysqli_num_rows($Std_record) > 0) {
     while ($row=mysqli_fetch_array($Std_record)) {
       echo "<p>Name : ".$row['stdName']."<br>";
       echo    "No.  : ".$row['stdNumber']."<br>";
       echo    "Age  : ".$row['stdAge']."<br>";
       echo    "Email: ".$row['stdEmail']."</p>";
     }
   } else {
     echo "<p>No record!!</p>";
   }
   mysqli_free_result($Std_record);
   mysqli_close($db_conn);

?>
```

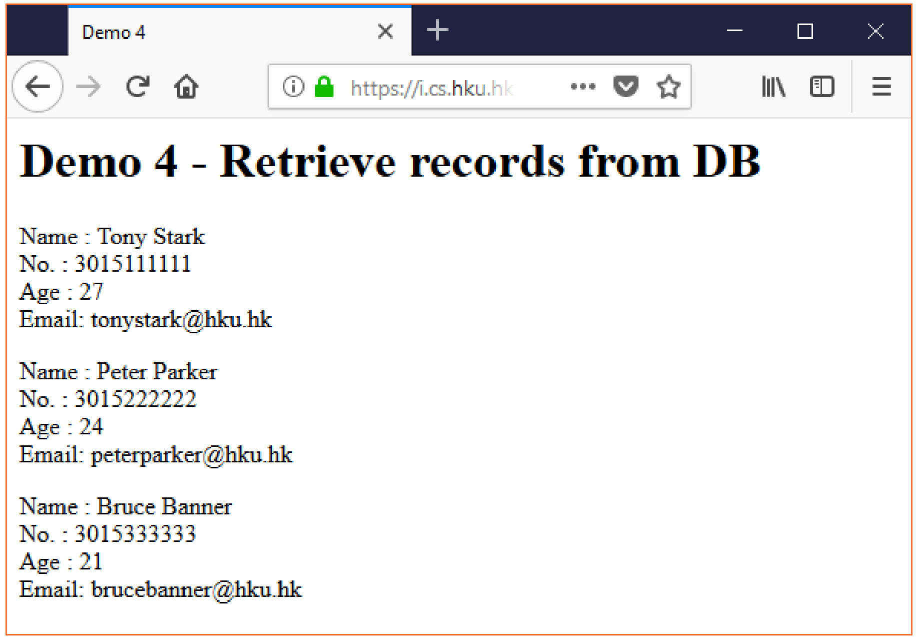

PHP Demo

https://i.cs.hku.hk

Account Registration Form

Student Name:	James Bond
Student No.:	3015007007
Age:	27
Student Email:	jamesbond@hku.hk

Submit Reset


Demo 4

https://i.cs.hku.hk

Demo 4 - Retrieve records from DB

```
Name : Tony Stark
No. : 3015111111
Age : 27
Email: tonystark@hku.hk

Name : Peter Parker
No. : 3015222222
Age : 24
Email: peterparker@hku.hk

Name : Bruce Banner
No. : 3015333333
Age : 21
Email: brucebanner@hku.hk
```

PHP Database Support

- PHP supports many databases
  - MySQL, MongoDB, IBM DB2, Mssql, Ingres, PostgreSQL, etc.
- MySQL is the most popular database system used with PHP.
  - MySQL uses standard SQL
  - MySQL is very fast, reliable, and easy to use
  - MySQL compiles on a number of platforms
- To have a quick overview on PHP + MySQL, please visit: https://www.w3schools.com/php/php_mysql_intro.asp

Database Design

- A database in a Relational DBMS is composed of one or more tables.
- A table is a two-dimensional container for data that consists of **records** (rows);
- Each record has the same number of columns, which are called **fields**, which contain the actual data.
- Each table will have one special field called a **primary key** that is used to uniquely identify each record in a table.

Database of Demo 4

Primary key field

Field names

Record

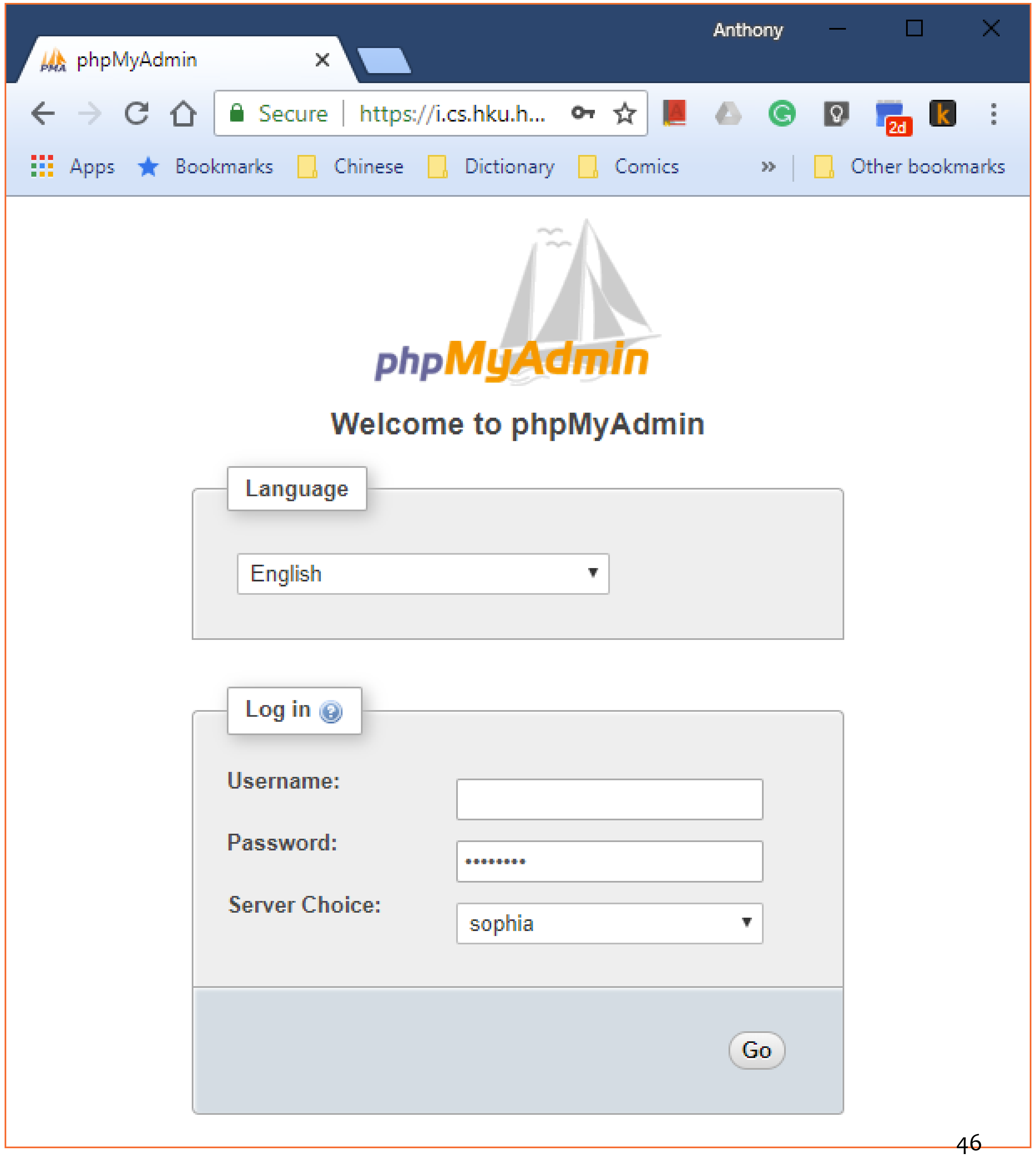
stdName	stdNumber	stdAge	stdEmail
Tony Stark	3015111111	27	tonystark@hku.hk
Peter Parker	3015222222	24	peterparker@hku.hk
Bruce Banner	3015333333	21	brucebanner@hku.hk

phpMyAdmin

https://i.cs.hku.hk/phpmyadmin/index.php

How should I apply for a MySQL database account?

Each user may apply for a MySQL database account using the online form at https://intranet.cs.hku.hk/common/mysqlacct/



Welcome to phpMyAdmin

Language

English

Log in

Username:

Password: ••••••••

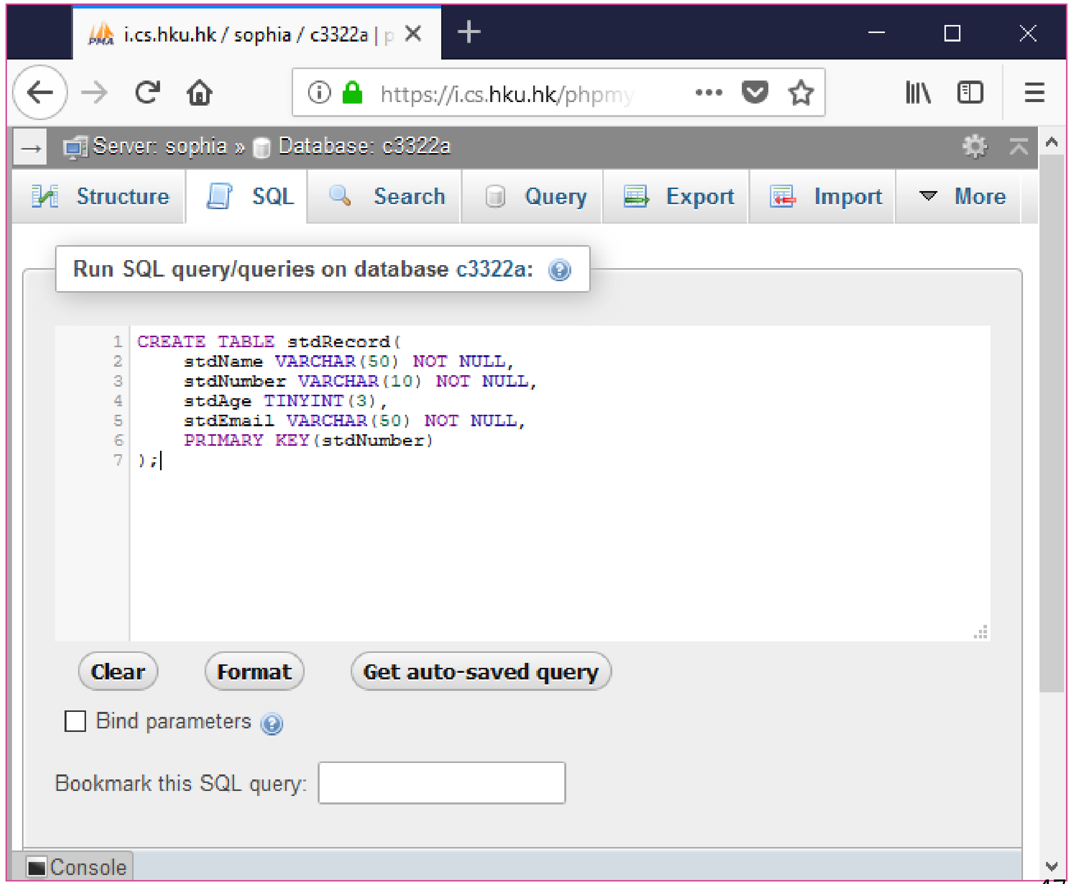
Server Choice: sophia

Go

Create Table

- The CREATE TABLE statement is used to create a new table in a database.

```
CREATE TABLE stdRecord(
     stdName VARCHAR(50) NOT NULL,
     stdNumber VARCHAR(10) NOT NULL,
     stdAge TINYINT(3),
     stdEmail VARCHAR(50) NOT NULL,
     PRIMARY KEY(stdNumber)
);
```

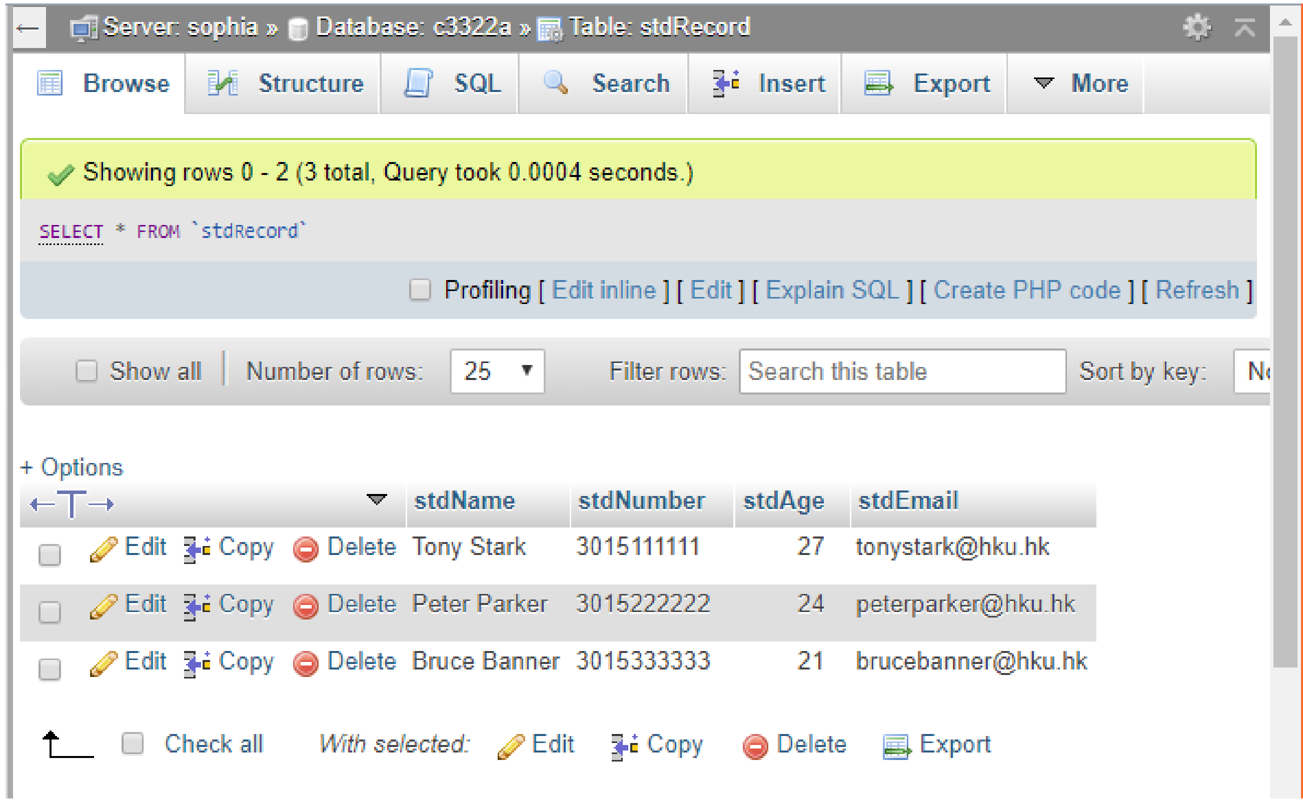

i.cs.hku.hk / sophia / c3322a | p
https://i.cs.hku.hk/phpmy
Server: sophia » Database: c3322a
Structure
SQL
Search
Query
Export
Import
More
Run SQL query/queries on database c3322a:
Clear
Format
Get auto-saved query
Bind parameters
Bookmark this SQL query:
Console

```
CREATE TABLE stdRecord(
    stdName VARCHAR(50) NOT NULL,
    stdNumber VARCHAR(10) NOT NULL,
    stdAge TINYINT(3),
    stdEmail VARCHAR(50) NOT NULL,
    PRIMARY KEY(stdNumber)
);
```

Insert Records

- Insert rows into the table.

```
INSERT INTO stdRecord (stdName, stdNumber,
stdAge, stdEmail) VALUES ("Tony Stark",
"3015111111", 27, "tonystark@hku.hk");

INSERT INTO stdRecord (stdName, stdNumber,
stdAge, stdEmail) VALUES ("Peter Parker",
"3015222222", 24, "peterparker@hku.hk");

INSERT INTO stdRecord (stdName, stdNumber,
stdAge, stdEmail) VALUES ("Bruce Banner",
"3015333333", 21, "brucebanner@hku.hk");
```



Server: sophia » Database: c3322a » Table: stdRecord
Browse Structure SQL Search Insert Export More

Showing rows 0 - 2 (3 total, Query took 0.0004 seconds.)

```
SELECT * FROM `stdRecord`
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all | Number of rows: 25 Filter rows: Search this table Sort by key:
+ Options

	stdName	stdNumber	stdAge	stdEmail
Edit Copy Delete	Tony Stark	3015111111	27	tonystark@hku.hk
Edit Copy Delete	Peter Parker	3015222222	24	peterparker@hku.hk
Edit Copy Delete	Bruce Banner	3015333333	21	brucebanner@hku.hk

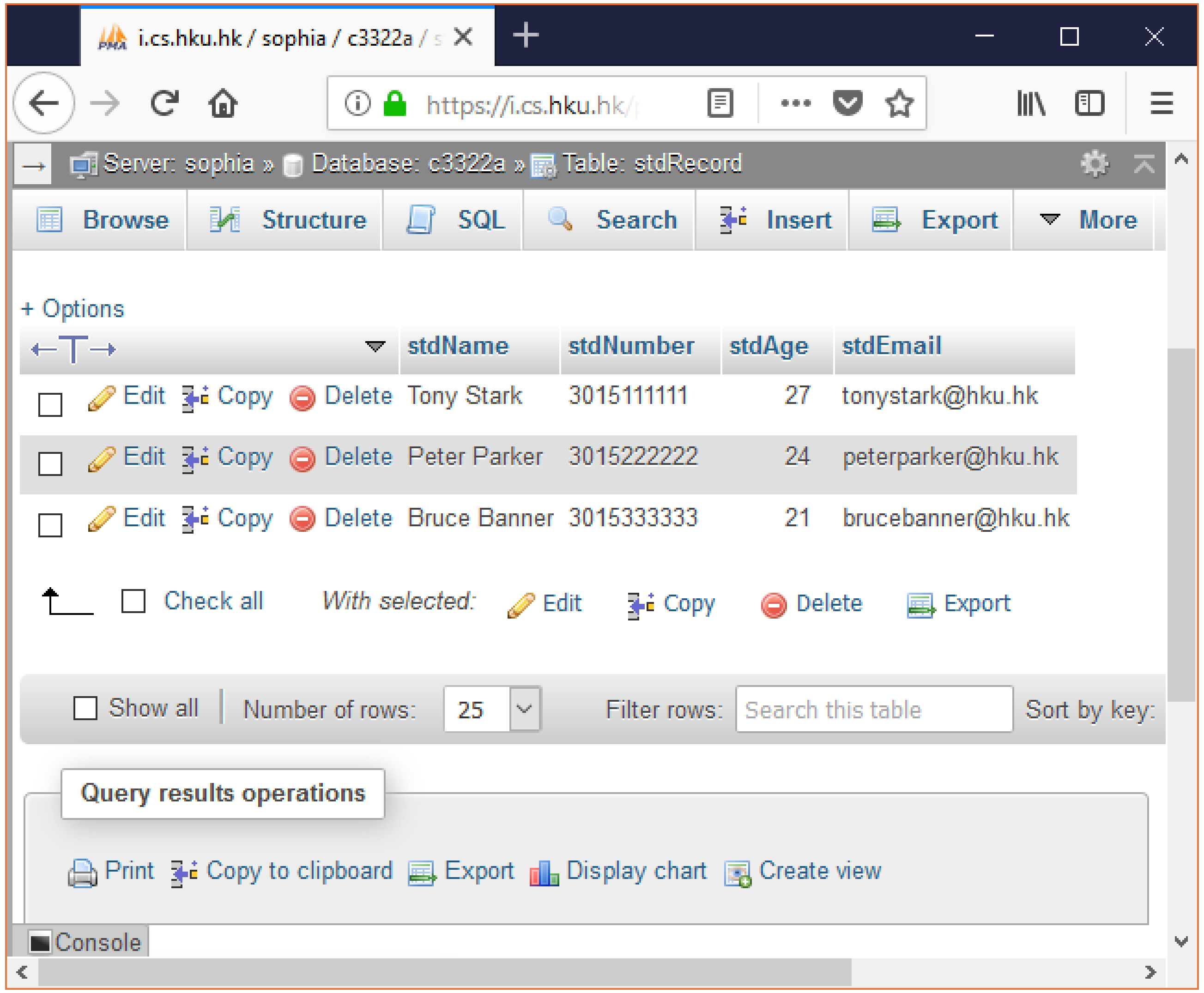
Check all With selected: Edit Copy Delete Export

SELECT

- The SELECT statement is used **to retrieve data** from the database.
- The result of a SELECT statement is a block of data typically called a **result set**.
- You must specify
  - which fields to retrieve and
  - which Table to retrieve from

SELECT

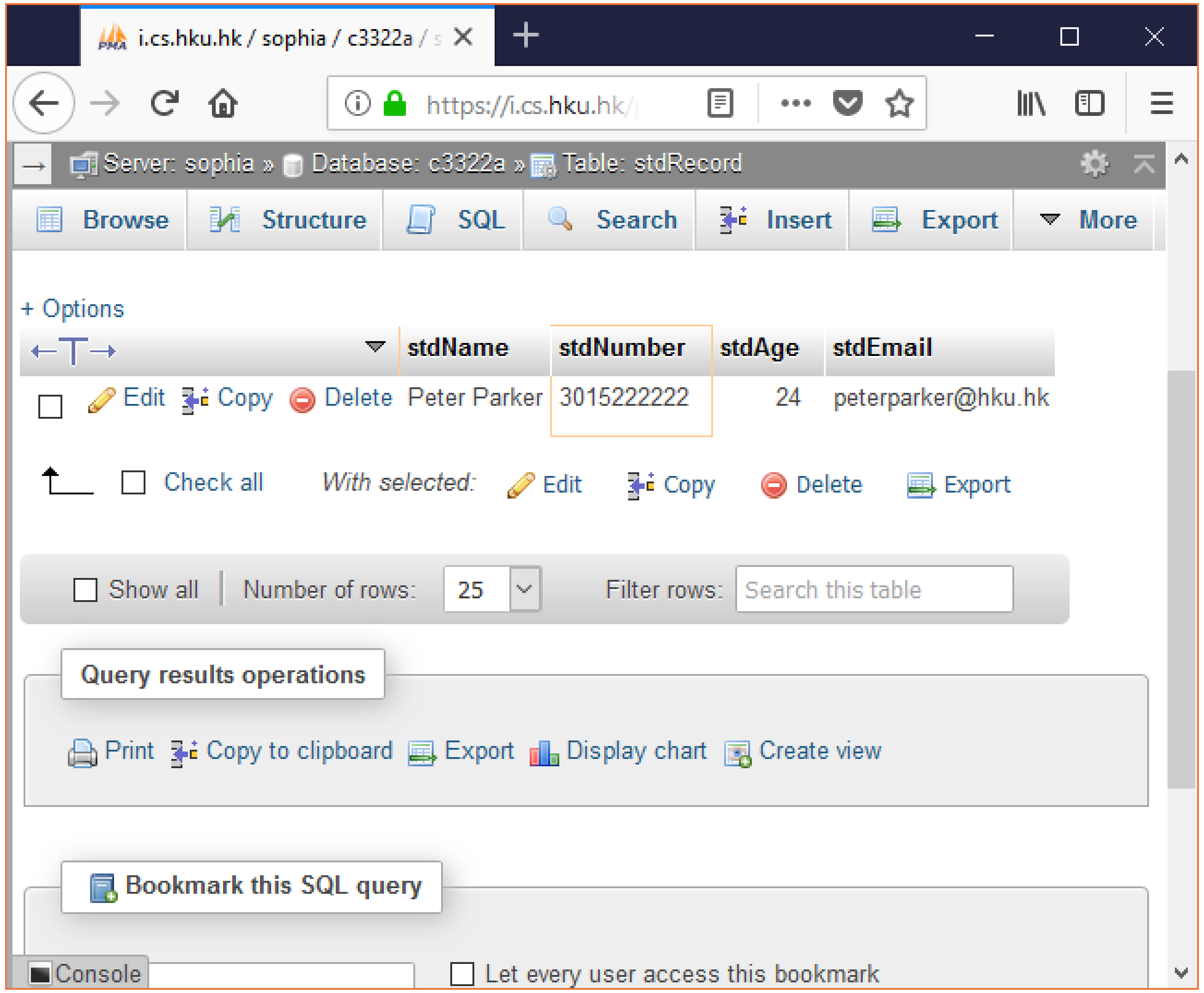
```
SELECT * FROM stdRecord;
```


i.cs.hku.hk / sophia / c3322a / s
https://i.cs.hku.hk/
Server: sophia » Database: c3322a » Table: stdRecord
Browse
Structure
SQL
Search
Insert
Export
More
+ Options

	stdName	stdNumber	stdAge	stdEmail
Edit Copy Delete	Tony Stark	3015111111	27	tonystark@hku.hk
Edit Copy Delete	Peter Parker	3015222222	24	peterparker@hku.hk
Edit Copy Delete	Bruce Banner	3015333333	21	brucebanner@hku.hk

SELECT

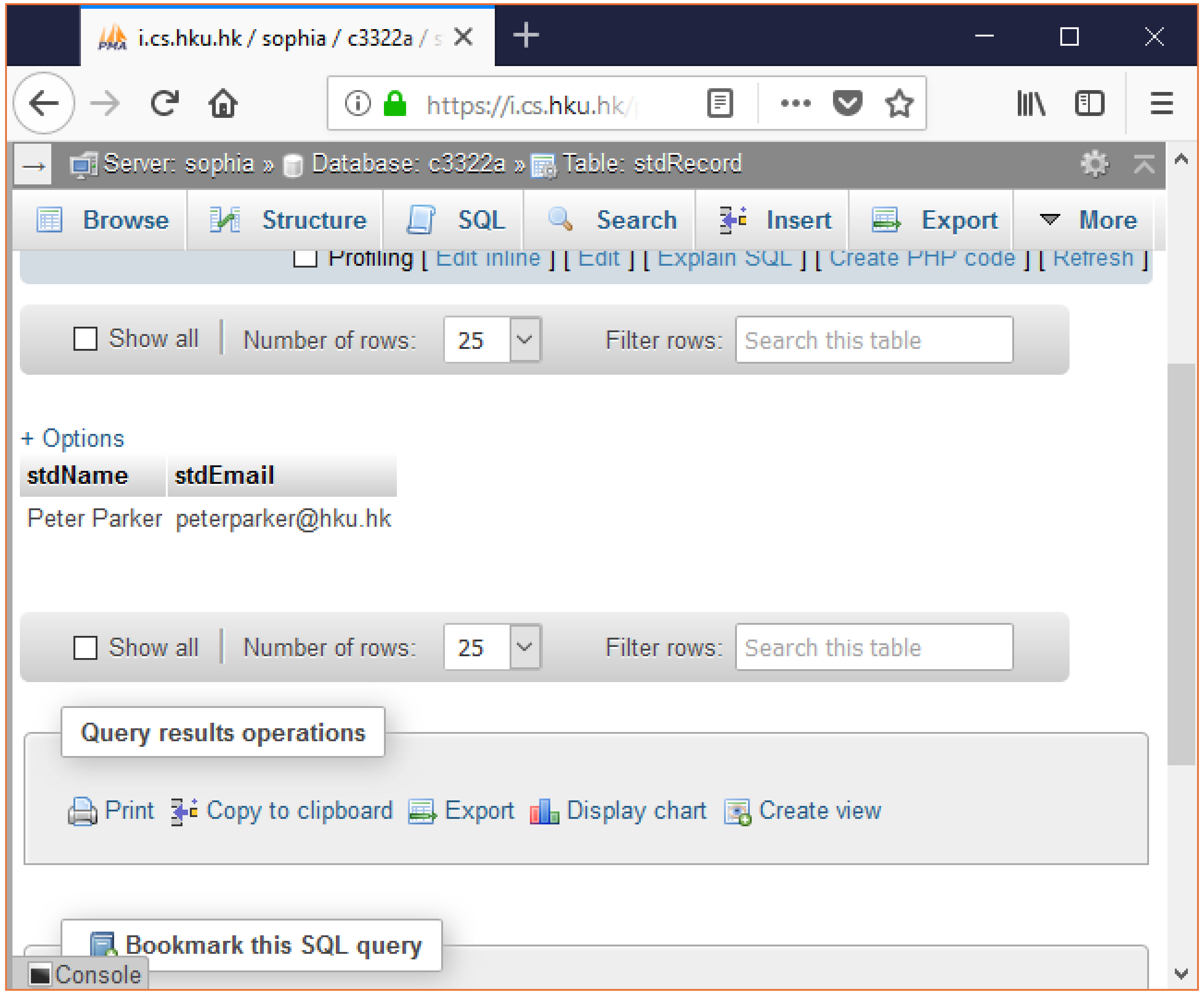
```
SELECT * FROM stdRecord WHERE
stdNumber = "3015222222";
```



	stdName	stdNumber	stdAge	stdEmail
Edit Copy Delete	Peter Parker	3015222222	24	peterparker@hku.hk

SELECT

```
SELECT stdName, stdEmail FROM stdRecord
WHERE stdNumber = "3015222222";
```



stdName	stdEmail
Peter Parker	peterparker@hku.hk

Accessing MySQl in PHP

1. Connect to the database.
2. Handle connection errors.
3. Execute the SQL query.
4. Process the results.
5. Free resources and close connection.

Demo 4

```
<?php
  #Connect to sophia
  $db_conn=mysqli_connect("sophia.cs.hku.hk", "c3322a", “XXXX", "c3322a")
    or die("Connection Error!".mysqli_connect_error());

  #Retrieve all records from DB
  $query="SELECT * FROM stdRecord";
  $Std_record=mysqli_query($db_conn, $query)
    or die("Query Error!".mysqli_error($db_conn));

  #Display the records
  if (mysqli_num_rows($Std_record) > 0) {
    while ($row=mysqli_fetch_array($Std_record)) {
      echo "<p>Name : ".$row['stdName']."<br>";
      echo    "No.  : ".$row['stdNumber']."<br>";
      echo    "Age  : ".$row['stdAge']."<br>";
      echo    "Email: ".$row['stdEmail']."</p>";
    }
  } else {
    echo "<p>No record!!</p>";
  }
```

Connect to the database
mysqli_connect(“db server”, “username”, “password”, “database”)

Handle connection errors
mysqli_connect_error( )

die(“error message”)

- The **mysqli_connect()** function opens a new connection to the MySQL server.
  - Returns the connection object to the MySQL server.

```
mysqli_connect(host,username,password,dbname,port,socket);
```

host	Optional. Specifies a host name or an IP address
username	Optional. Specifies the MySQL username
password	Optional. Specifies the MySQL password
dbname	Optional. Specifies the default database to be used
port	Optional. Specifies the port number.
socket	Optional. Specifies the socket.

- The **mysqli_connect_error()** function returns the error description from the last connection error.

```
mysqli_connect_error();
```

```
Connection Error! Access denied for user
'c3322a'@'i1.cs.hku.hk' (using password: NO)
```

- The **die()** function prints a message and exits the current script.

```
die(message);
```

Demo 4

```
<?php
   #Connect to sophia
   $db_conn=mysqli_connect("sophia.cs.hku.hk", "c3322a", “XXXX", "c3322a")
     or die("Connection Error!".mysqli_connect_error());

   #Retrieve all records from DB
   $query="SELECT * FROM stdRecord";
   $Std_record=mysqli_query($db_conn, $query)
     or die("Query Error!".mysqli_error($db_conn));

   #Display the records
   if (mysqli_num_rows($Std_record) > 0) {
     while ($row=mysqli_fetch_array($Std_record)) {
       echo "<p>Name : ".$row['stdName']."<br>";
       echo    "No.  : ".$row['stdNumber']."<br>";
       echo    "Age  : ".$row['stdAge']."<br>";
       echo    "Email: ".$row['stdEmail']."</p>";
     }
   } else {
     echo "<p>No record!!</p>";
   }
```

Execute the SQL query
mysqli_query(“db connection”, “query string”)

Handle errors
mysqli_error(“db connection”)

Process the results
mysqli_num_rows(“ return result set”)

Process the results
mysqli_fetch_array(“ return result set”)

- The **mysqli_query()** function **performs a query** against the database.
  - For successful SELECT, SHOW, DESCRIBE, or EXPLAIN queries it will **return a mysqli_result object**. For other successful queries it will return TRUE. FALSE on failure.

```
mysqli_query(connection,query,resultmode);
```

connection	Required. Specifies the MySQL connection to use
query	Required. Specifies the query string
resultmode	Optional. Either: * MYSQLI_STORE_RESULT [**default**] * MYSQLI_USE_RESULT (Use unbuffered query; use this if we have to retrieve large amount of data)

- The **mysqli_error()** function **returns the last error description** for the most recent function call.

```
mysqli_error(connection);
```

- The **mysqli_num_rows()** function **returns the number of rows** in a result set.

```
mysqli_num_rows(result);
```

result	Specifies a result set identifier returned by mysqli_query().

- The **mysqli_fetch_array()** function **fetches a result row** as an **associative array**, **a numeric array**, or both.
  - Returns an array of strings that corresponds to the fetched row. NULL if there are no more rows in result-set.

```
mysqli_fetch_array(result,resulttype=MYSQLI_BOTH);
```

result	Specifies a result set identifier returned by mysqli_query().
resulttype	Optional. Specifies what type of array that should be produced. Can be one of the following values: MYSQLI_ASSOC, MYSQLI_NUM, or MYSQLI_BOTH

- How about fetching all result rows in one call?
- The **mysqli_fetch_all()** function **fetches all result rows** and returns the result-set as an **associative array**, **a numeric array**, or both.
  - Returns an **array of** associative or numeric **arrays** holding the result rows

```
mysqli_fetch_all(result,resulttype=MYSQLI_NUM);
```

result	Specifies a result set identifier returned by mysqli_query().
resulttype	Optional. Specifies what type of array that should be produced. Can be one of the following values: MYSQLI_ASSOC, MYSQLI_NUM, or MYSQLI_BOTH

  - Good for directly converting the result array to JSON data

Demo 4

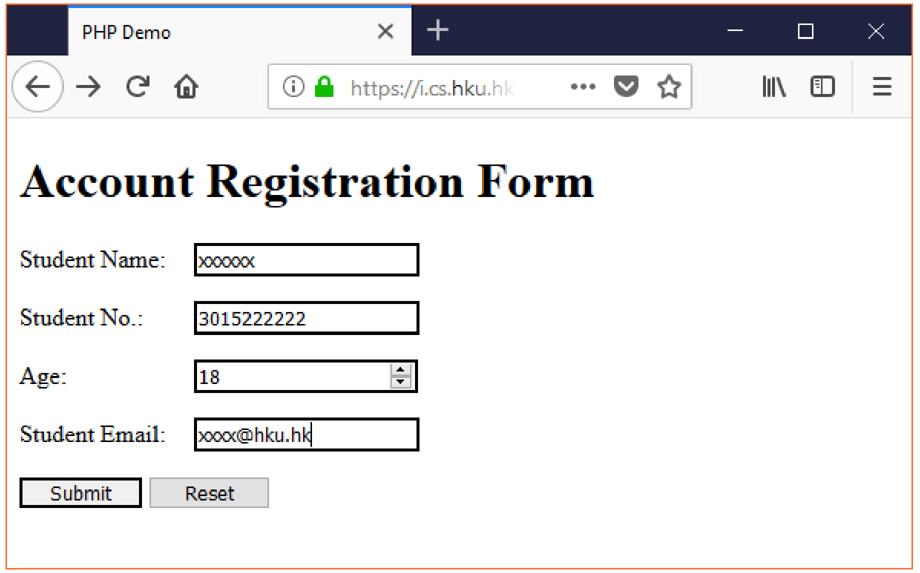
```
<?php
   #Connect to sophia

   #Retrieve all records from DB

   #Display the records
   if (mysqli_num_rows($Std_record) > 0) {
     while ($row=mysqli_fetch_array($Std_record)) {
       echo "<p>Name : ".$row['stdName']."<br>";
       echo    "No.   : ".$row['stdNumber']."<br>";
       echo    "Age   : ".$row['stdAge']."<br>";
       echo    "Email: ".$row['stdEmail']."</p>";
     }
   } else {
     echo "<p>No record!!</p>";
   }
   mysqli_free_result($Std_record);
   mysqli_close($db_conn);

?>
```

Free resources and close connection
mysqli_free_result("result set")

Free resources and close connection
mysqli_close("db connection")

Demo 5 – Check Duplication Before INSERT

```
<body>
 <h1>Account Registration Form</h1>
 <form id="RegForm" action="view6.php" method="post">
  <p>
   <label for="name">Student Name:</label>
   <input type="text" id="name" name="name" maxlength="50" required>
  </p>
   :
   :
```



PHP Demo

https://i.cs.hku.hk

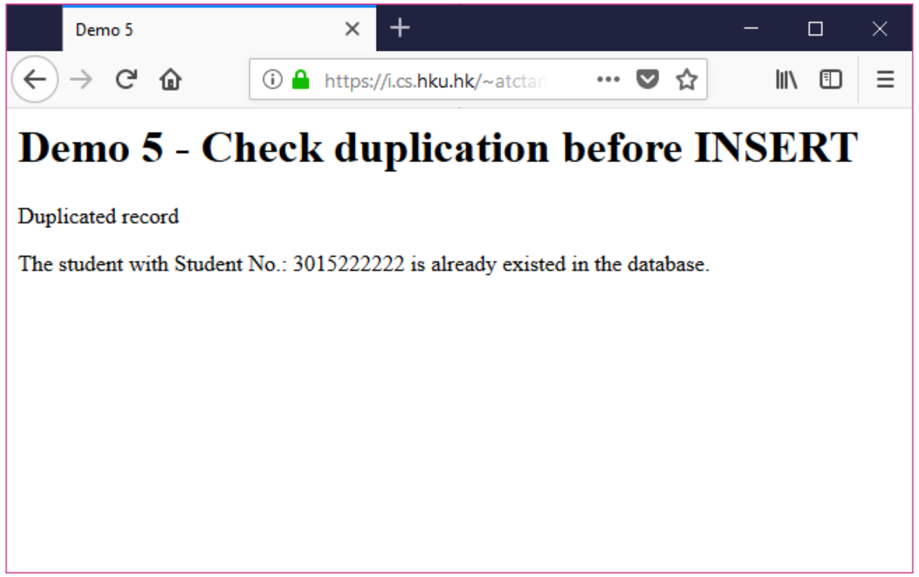
Account Registration Form

Student Name: xxxxxx

Student No.: 3015222222

Age: 18

Student Email: xxxx@hku.hk



Demo 5

https://i.cs.hku.hk/~atcta

Demo 5 - Check duplication before INSERT

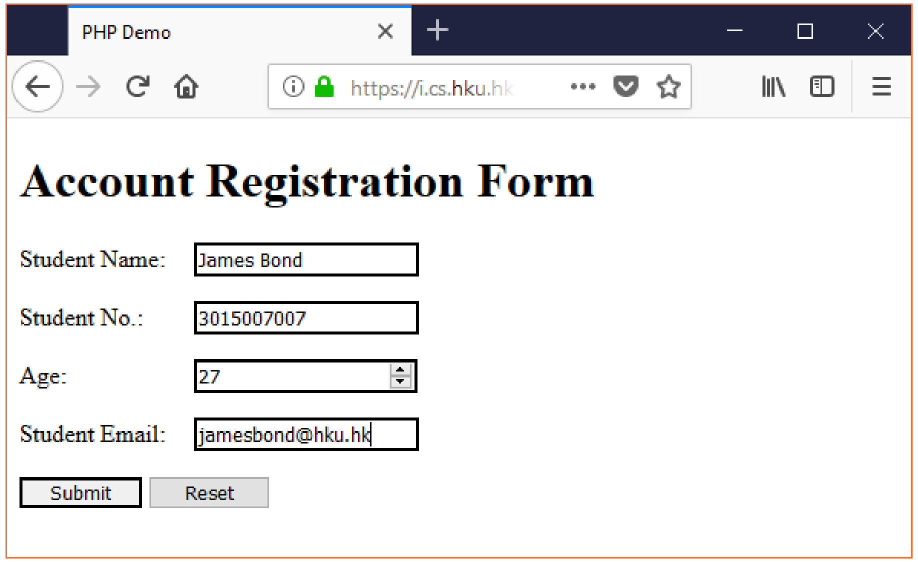
Duplicated record

The student with Student No.: 3015222222 is already existed in the database.

Demo 5 – Check Duplication Before INSERT

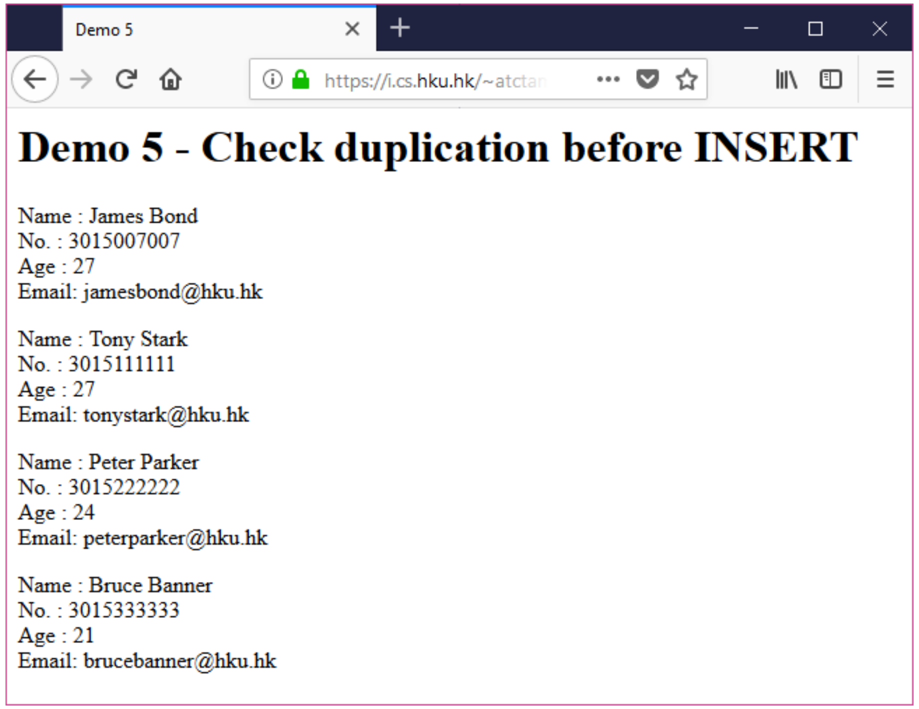
```
<body>
 <h1>Account Registration Form</h1>
 <form id="RegForm" action="view6.php" method="post">
  <p>
   <label for="name">Student Name:</label>
   <input type="text" id="name" name="name" maxlength="50" required>
  </p>
    :
    :
```

http://i7.cs.hku.hk/~atctam/c3322/PHP/form-php6.html



Account Registration Form

Student Name:	James Bond
Student No.:	3015007007
Age:	27
Student Email:	jamesbond@hku.hk



Demo 5 - Check duplication before INSERT

```
Name : James Bond
No. : 3015007007
Age : 27
Email: jamesbond@hku.hk

Name : Tony Stark
No. : 3015111111
Age : 27
Email: tonystark@hku.hk

Name : Peter Parker
No. : 3015222222
Age : 24
Email: peterparker@hku.hk

Name : Bruce Banner
No. : 3015333333
Age : 21
Email: brucebanner@hku.hk
```

Demo 5

```
<?php
   #Connect to sophia
   $db_conn=mysqli_connect("sophia.cs.hku.hk", "c3322a", "xxxx", "c3322a")
     or die("Connection Error!".mysqli_connect_error());

   #Check whether the record in DB
   $name=$_POST['name'];
   $num=$_POST['number'];
   $age=$_POST['age'];
   $email=$_POST['email'];
   $query="SELECT * FROM stdRecord WHERE stdNumber = '$num'";
   $result = mysqli_query($db_conn, $query)
     or die("<p>Query Error!<br>".mysqli_error($db_conn)."</p>");

   if (mysqli_num_rows($result) > 0) {
     echo "<p>Duplicated record</p>";
     echo "<p>The student with Student No.: ".$num." is already existed in the database.";
   } else {
        :
        :
```

Execute the SQL query.
Process the results.

Demo 5

Execute the SQL query.
Process the results.

```
    :
    :
} else {
    $query="INSERT INTO stdRecord (stdName, stdNumber, stdAge, stdEmail)
      VALUES ('$name', '$num', '$age', '$email')";
    if (!mysqli_query($db_conn, $query)) {
      echo "<p>Error insert!!<br>".mysqli_error($db_conn)."</p>";
    }

    #Retrieve all records from DB
    $query="SELECT * FROM stdRecord";
    $Std_record=mysqli_query($db_conn, $query)
      or die("<p>Query Error!<br>".mysqli_error($db_conn)."</p>");

    #Display the records
    if (mysqli_num_rows($Std_record) > 0) {
            :
            :
```

Session & Cookie

Cookies

- Cookies are the key/value (variable/value) pairs maintained by **browsers**.
- How cookie works:
  - When receiving an HTTP request, a server can send a **Set-Cookie header** with the response.
  - Browser stores the cookie.
  - With future requests made to the same server, the browser sends the cookie inside the request in a **Cookie HTTP header**.
    - An **expiration date** or duration can be specified, after which the cookie is no longer sent.
    - Restrictions to a **specific domain and path** can be set, limiting where the cookie is sent.

PHP Cookies

- A cookie is created with the setcookie() function.

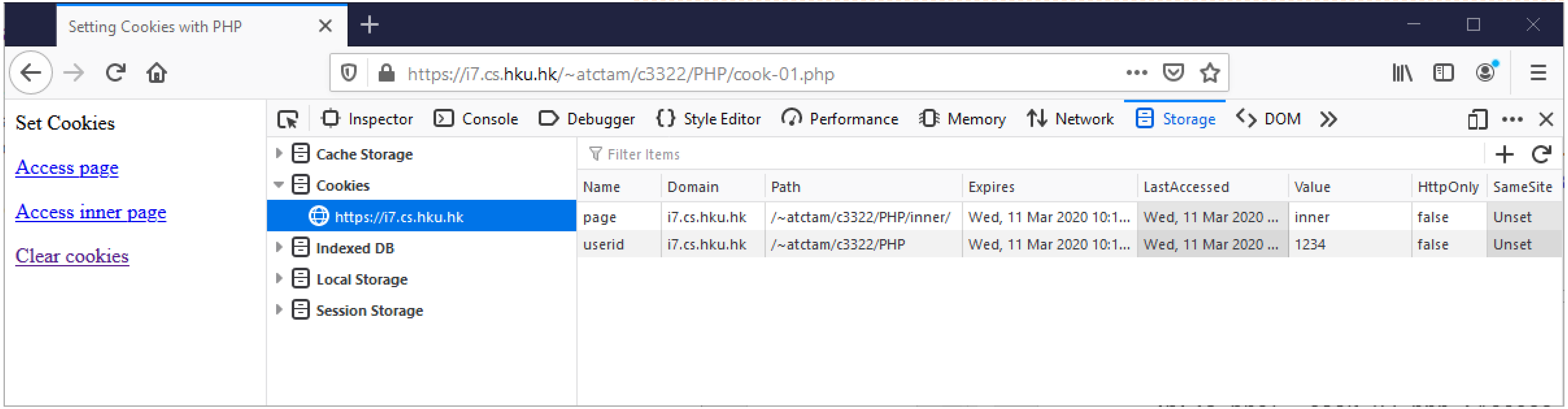
```
setcookie(name, value, expire, path, domain, secure, httponly);
```

name	The cookie name.
value	Optional. Specifies the value.
expire	Optional. Specifies the time the cookie expires.
path	Optional. Specifies the directories for which the cookie is valid.
domain	Optional. Specifies the domain name for which the cookie is valid.
secure	Optional. Specifies whether carries by HTTPS or HTTP.
httponly	Optional. Limits only to HTTP protocol; JavaScript cannot access it.

- Retrieve the value of a cookie using the superglobal variable $_COOKIE.
- Use the isset() function to find out if the cookie is set.
- To delete a cookie variable, just use setcookie() function to **set the cookie expiration time** to be anytime in the past.

Demo 6 - cook-01.php

```
<?php
   setcookie("userid", "1234", time()+3600);
   setcookie("page", "inner", time()+3600, "/~atctam/c3322/PHP/inner");
?>
<html>
   <head>
      <title>Setting Cookies with PHP</title>
   </head>
   <body>
      <?php echo "Set Cookies"?>
      <p><a href="cook-02.php">Access page</a></p>
      <p><a href="inner/cook-02.php">Access inner page</a></p>
      <p><a href="cook-03.php">Clear cookies</a></p>
   </body>
</html>
```


Setting Cookies with PHP

https://i7.cs.hku.hk/~atctam/c3322/PHP/cook-01.php

Set Cookies

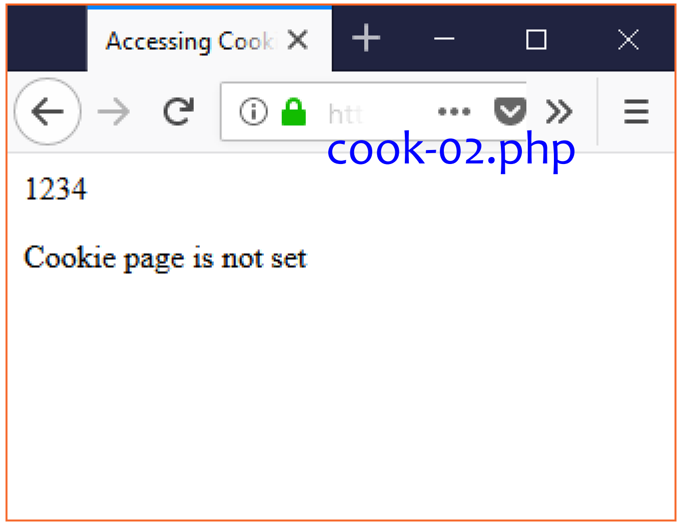
Access page

Access inner page

Clear cookies

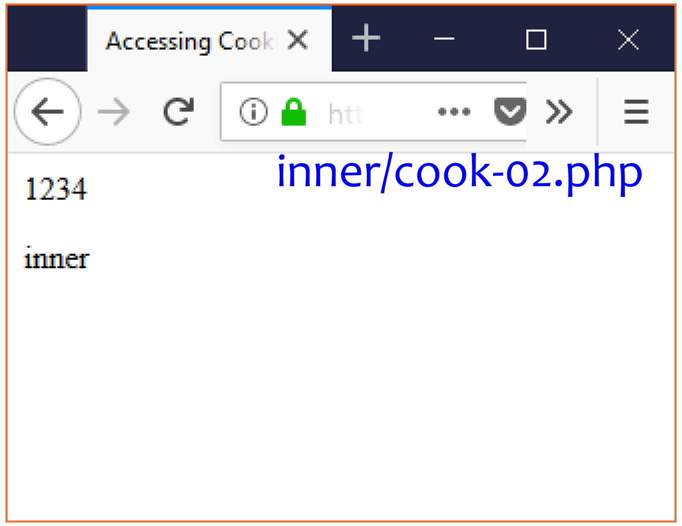
Name	Domain	Path	Expires	LastAccessed	Value	HttpOnly	SameSite
page	i7.cs.hku.hk	/~atctam/c3322/PHP/inner/	Wed, 11 Mar 2020 10:1...	Wed, 11 Mar 2020 ...	inner	false	Unset
userid	i7.cs.hku.hk	/~atctam/c3322/PHP	Wed, 11 Mar 2020 10:1...	Wed, 11 Mar 2020 ...	1234	false	Unset

Demo 6 - cook-02.php



cook-02.php

```
1234

Cookie page is not set
```



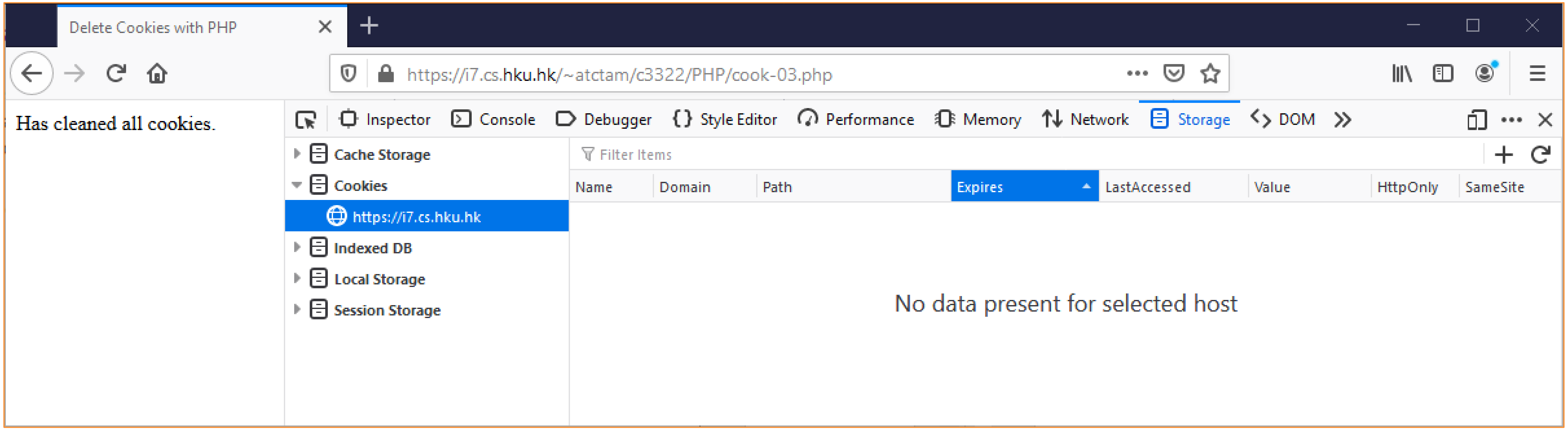
inner/cook-02.php

```
1234

inner
```

```
<html>
   <head>
      <title>Accessing Cookies with PHP</title>
   </head>
   <body>
      <p>
      <?php
        if (isset($_COOKIE["userid"])) {
          echo $_COOKIE["userid"]. "<br>";
        } else {
          echo "Cookie userid is not set\n";
        }
      ?>
      </p>
      <p>
      <?php
        if (isset($_COOKIE["page"])) {
          echo $_COOKIE["page"]. "<br>";
        } else {
          echo "Cookie page is not set\n";
        }
      ?>
      </p>
   </body>
</html>
```

Demo 6 - cook-03.php

```
<?php
   setcookie("userid", "", time()-3600);
   setcookie("page", "", time()-3600, "/~atctam/c3322/PHP/inner/");
?>
<html>
   <head>
      <title>Delete Cookies with PHP</title>
   </head>
   <body>
      <p>
        Has cleaned all cookies.
      </p>
   </body>
</html>
```



Has cleaned all cookies.

No data present for selected host

Sessions

- Server-side Cookies
  - A session is a methods of **storing data** (using variables) **on the server** and the data will be **available to all pages** on the site during that visit.
- How session works:
  - Once connected, server sends a **cookie** that contains the session ID to the browser.
  - In the subsequent requests, the browser sends the **session ID cookie** (together with other cookies from this site) to the server.
  - PHP can **retrieve the data based on the session ID** and make the data available in your PHP script.
  - The session **ends** once the window or tab in which the webpage was loaded, is closed or the server explicitly destroys all session variables.

Open a New Session

- A session is **started with** the session_start() function.
- This function first checks if a session is already started and if none is started then it starts a new session.
  - If a new session is started, a **cryptographic session ID** is created.
- Session data is stored on the server in text file or even database.
- The session ID is **associated with** saved session data, in this way providing a method for tying a particular **user** to this data.
- Important Note:
  - The session_start() function **must be the very first thing** in your PHP file before any HTML tags.
  - ***All PHP files*** must include the session_start() function to access the session data.

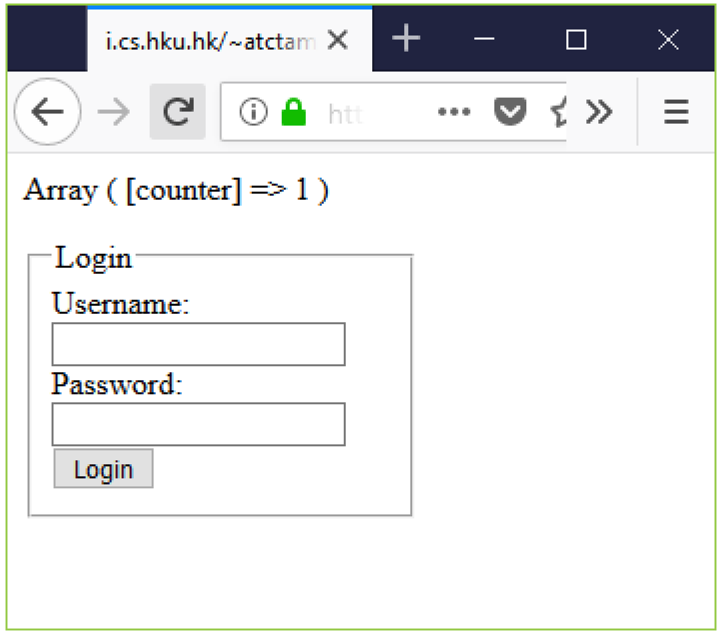
Propagation of Session ID

- There are two methods to propagate a session ID:
  - Cookie [default]
    - Send the session ID to the browser in form of a cookie named PHPSESSID.
      - PHPSESSID=9hjtvg980cakoblsloa4mag75u
  - URL parameter
    - Propagated by the URL as part of the query string
      - <a href="login.php?PHPSESSID=9hjtvg980cakoblsloa4mag75u">
    - PHP is capable of transforming links transparently. If the run-time option **session.use_trans_sid** is enabled, relative URLs will be changed to contain the session id automatically.
- Session IDs are propagated across different HTTP requests by cookies or by appending to each URL as query string.

Access Session Data

- Session data can be accessed via the **$_SESSION** superglobal array variable.
- Use a session variable (no declarations needed).
  - $_SESSION["something"] = "somevalue";
- Use **isset()** to check whether a session variable is set.
- Use **unset()** to remove a session variable.
- To free all session variables, use **session_unset()**.
- To destroy all of the data associated with the current session that is stored in the session storage, use **session_destroy()**.
- To remove session cookie, use **setcookie()** to set the session ID to expire.

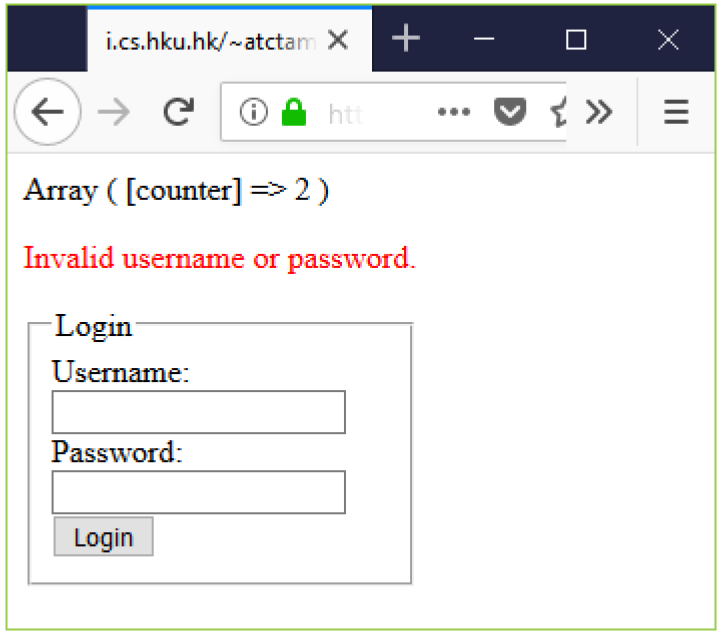
Demo 7 – login.php



Array ([counter] => 1)

Login
Username:
Password:
Login

Before log on
or
after log out

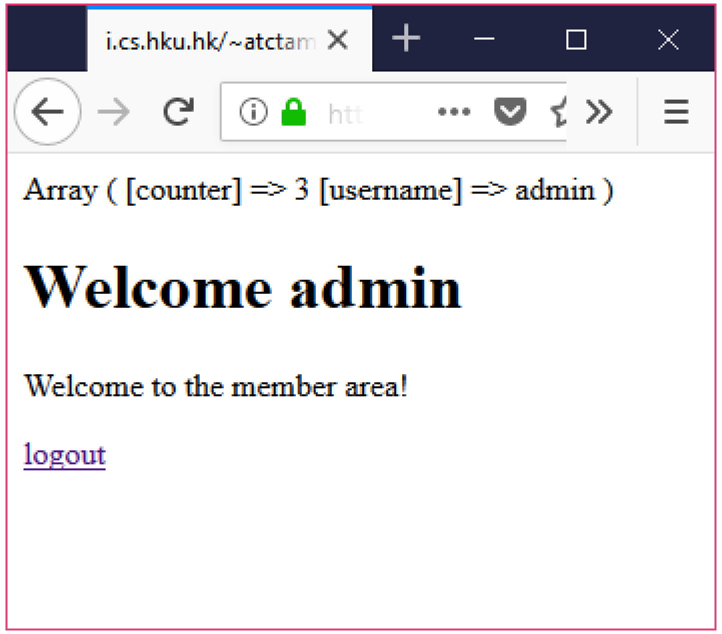


Array ([counter] => 2)

Invalid username or password.

Login
Username:
Password:
Login

Fail authentication



Array ([counter] => 3 [username] => admin)

Welcome admin

Welcome to the member area!

logout

After authenticated

Demo 7

Get session data

Just for demo purpose

Debug: show all session data

Start here

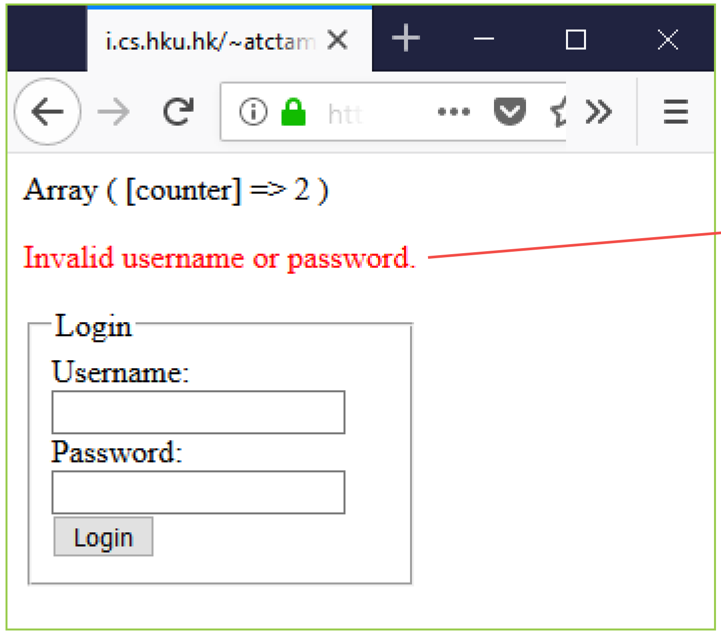
```php
<?php
#Source: www.zentut.com/php-tutorial/php-session/
#A simple login example

session_start();
#Set the access counter
if (isset($_SESSION['counter'])) {
  $_SESSION['counter'] += 1;
} else {
   $_SESSION['counter'] = 1;
}

#Set a predefined account
define("SYSUSER",'admin');
define("SYSPASSWORD",'secret');
#Debug: show current session data
print_r($_SESSION);

#Our main() function
start();
```

Demo 7

```
function start() {
  if(isset($_POST['login'])) { //if is a POST request
    if (authenticate()) {
      // display secured content if user logged in successfully
      display_secured_content();
    } else {
      // display login form again with message
      display_login_form('Invalid username or password.');
    }
  } else if(isset($_GET['action']) && $_GET['action'] == 'Logout') {
    // obtain a GET request with query string action=logout
    logout();
  } else {
    // is a GET request
    if (authenticate()) {
      display_secured_content();
    } else {
    // default: display the login form
      display_login_form();
    }
  }
}
```

Already authenticated.
Has session cookie.

Demo 7



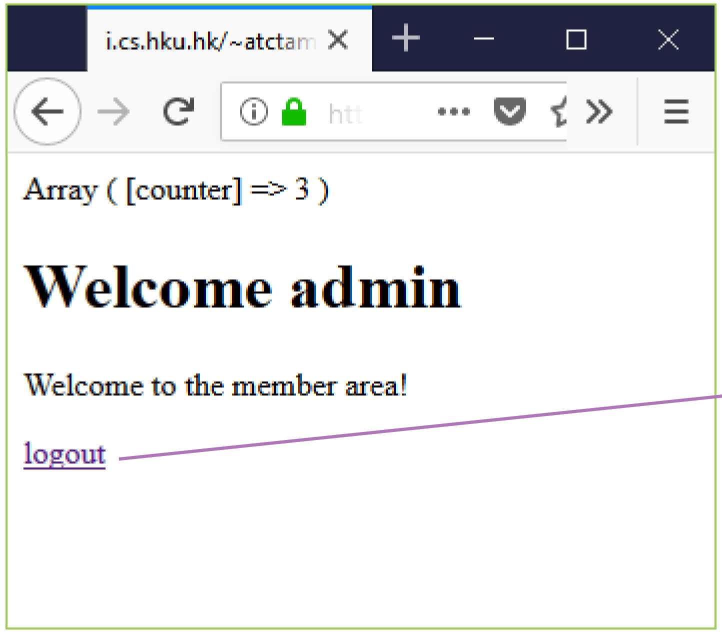
i.cs.hku.hk/~atctam

Array ([counter] => 2)

Invalid username or password.

```
function display_login_form($msg='') {
  ?>
  <style>
    .error {color: #ff0000;}
    fieldset {width: 300px;}
  </style>
  <form action="login.php" method="post">
    <p class="error">
      <?php echo $msg; ?>
    </p>
    <fieldset name="logininfo">
      <legend>Login</legend>
      <label for="username">Username:</label>
      <input type="text" name="username" id="username"><br>
      <label for="password">Password:</label>
      <input type="password" name="password" id="password"><br>
      <input type="submit" name="login" value="Login">
    </fieldset>
  </form>
  <?php
}
```

Demo 7



Array ([counter] => 3)

Welcome admin

Welcome to the member area!

logout

```
function display_secured_content() {
  ?>
  <h1>Welcome <?php echo $_SESSION['username'];?></h1>
  <p>Welcome to the member area!</p>
  <p>
    <a href="login.php?action=Logout">logout</a>
  </p>
  <?php
}
```

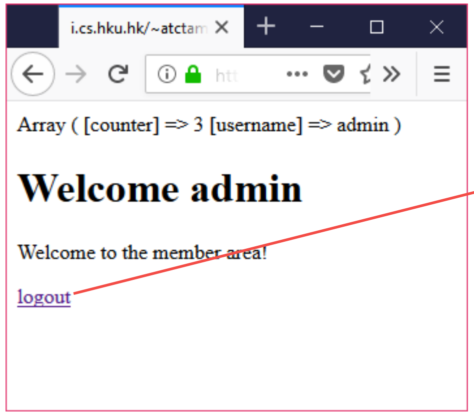
Demo 7

Authenticate user

Store session cookie if matched

```
function authenticate() {
  if (isset($_SESSION['username'])) { //if already authenticated
    return true;
  }
  if (isset($_POST['username']) && isset($_POST['password'])) {
    $username = $_POST['username'];
    $password = $_POST['password'];
    #Check username & password
    if ($username == SYSUSER  && $password == SYSPASSWORD) {
      #Matched username & password
      $_SESSION['username'] = SYSUSER;  //Store authenticated variable
      session_write_close();  //free session lock
      return true;
    } else {  //Wrong credential
      return false;
    }
  }
}
```

Demo 7

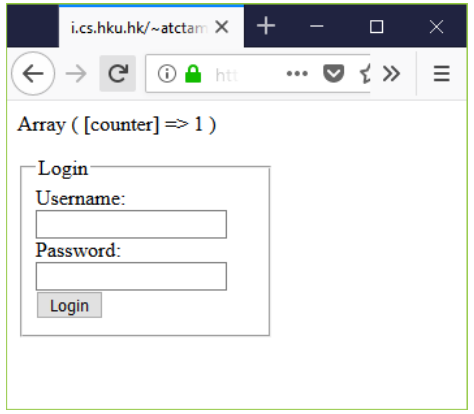

i.cs.hku.hk/~atctan

Array ([counter] => 3 [username] => admin)

Welcome admin

Welcome to the member area!

logout


i.cs.hku.hk/~atctan

Array ([counter] => 1)

```
function logout() {
  #set SESSION cookie to expire ==> delete cookie
  if (isset($_COOKIE[session_name()])) {
    setcookie(session_name(),'',time()-3600, '/');
  }
  session_unset();
  session_destroy();
  #Set redirection
  header('location: login.php');
}
?>
```

References

- PHP Tutorial – Tutorialspoint.com
  - https://www.tutorialspoint.com/php/index.htm
- PHP 5 Tutorial – W3school.com
  - https://www.w3schools.com/php/default.asp
- PHP Manual
  - https://secure.php.net/manual/en/index.php
- PHP The Right Way
  - https://phptherightway.com/
- PHP MySQL – W3school.com
  - https://www.w3schools.com/php/php_mysql_intro.asp
- PHP Session – zentut.com
  - http://www.zentut.com/php-tutorial/php-session/