HNDIT3022
Web Programming

Week1:- Introduction to web architectures

Learning Outcomes (LO)

- After successful completion of this course, the student should be able to:
  - LO1. Implement a dynamic webpage using server-side scripting.
  - LO2. Use server-side scripting language constructs, conditions statements, loops, arrays, strings, functions, files, IO, and objects, in programming a webpage.
  - LO3. Integrate SQL database with the webpage programmed in server-side scripts.
  - LO4. Describe the three-tier architecture and MVC framework from the perspective of a web programmer.
  - LO5. Use web programming libraries and frameworks

Course Aims

- To prepare the students to design and develop web-based applications and dynamic websites using server site scripts and libraries. Creating multitier applications using the MVC framework

References

- Required Textbook and Resources

  Learning PHP, MySQL, JavaScript, and CSS: A Step-by-Step Guide to Creating Dynamic Websites, by Robin Nixon, O'Reilly Media

- Recommended Additional Resources

  W3School web site

World Wide Web

- The World Wide Web (WWW) is an Internet-based hypertext project that allows global information sharing.
- It is a collection of interlinked multimedia documents that are stored on the Internet.
- A protocol (HTTP) is used to access the documents. Also, it is one way of sharing information via the Internet

WWW Components

- **Structural Components**
  - Clients/browsers – to access and operate on the network
  - Servers – run on sophisticated hardware and keep shared resources
  - Internet – the global infrastructure which facilitates data transfer
- **Syntactic Components**
  - Hyper Text Transfer Protocol (HTTP)
  - Hyper Text Markup Language (HTML)
    - extensible Markup Language (XML)
  - Uniform Resource Identifiers (URIs)

Activity 1.1

- Search the differences between the internet and WWW.

Activity 1.2

- Discuss about the servers and browsers.

HTTP

- Hypertext Transfer Protocol (HTTP) is an application-layer protocol.
- HTTP is a communication standard governing the requests and responses that take place between the browser running on the end user's computer and the web server.

URI

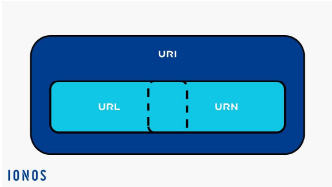
- The Uniform Resource Identifier (URI) is intended to identify abstract or physical resources on the Internet.
- A Uniform Resource Locator (URL), or web address, is the most common form of URI.
  - It is used for unambiguously identifying and locating websites or other web-connected resources.

HTML

- HTML is the standard markup language for creating Web pages and describes the structure of a Web page.
- HTML consists of a series of elements.
- HTML elements tell the browser how to display the content.
- HTML elements label pieces of content such as "this is a heading", "this is a paragraph", "this is a link", etc.

Activity 1.3

- Search URI vs. URL vs. URN

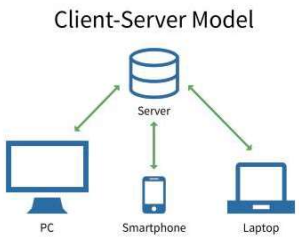


Introduction to web architectures

Client-Server Architecture

- A software architecture model consisting of two parts,
  - client systems,
  - and server systems

both communicating over a computer network or on the same computer.



Web Architecture

- The architecture of an application refers to the overall structure of an application, including the way it is designed, implemented, and deployed.
- There are many different approaches, strategies, and patterns of web application architecture.

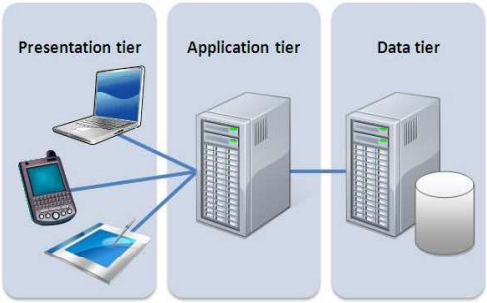
The Client: Client-server architecture

- The requester of services and the server is defined as the provider of services.
- Characteristics of a client
  - **Initiates requests(Master)**
  - **Waits for and receives replies**
  - **Usually connects to a small number of servers at one time**
  - **Typically interacts directly with end-users using a graphical user interface**
- Examples: web browsers, email clients, and online chat clients

The Server: Client-server architecture

- Host that is running one or more server programs that can share information or resources.
- Characteristics of a server
  - **Passive (slave)**
  - **Waits for requests from clients**
  - **Upon receipt of requests, processes them and then serves replies**
  - **Usually accepts connections from a large number of clients**
  - **Typically does not interact directly with end users**
- Examples: web servers, database servers, and mail servers

Three-tier architecture



Multi-tier Architecture

- A client–server architecture in which presentation, application processing, and data management functions are logically separated.
- Developers can create flexible and reusable applications.
- Most widespread use of multi-tier architecture is the three-tier architecture.

Benefits of three-tier architecture

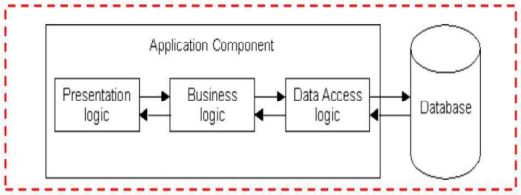
- Services of each tier can be customized and optimized without impacting the other tiers.
- Faster development
- Improved scalability
- Improved reliability
- Improved security

Web Application

- Any application that uses Web Technologies including web browsers, web servers, and Internet protocols is called a Web Application.
- A web application can be divided into three different layers.

1-tier web architecture

- All 3 layers are on the same machine.

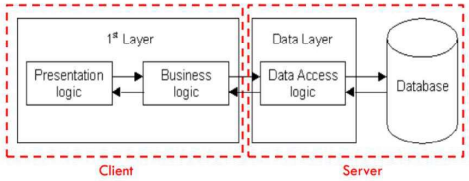


Web Oriented three-tier architecture

- Presentation tier
  - Browser/custom client
  - Client Side Scripting (JavaScript)
  - Applets.
- Logical tier
  - Web Server (Apache, IIS, WebSphere etc.)
  - Scripting Languages (PHP, Perl, etc.)
  - Programming Languages (Java, C, C#, etc.)
  - Application Frameworks (Ruby on Rails etc.)
- Data tier
  - Database Management System (DBMS) (Oracle, MySQL, SQL Server, DB2, etc.)
  - XMLDB

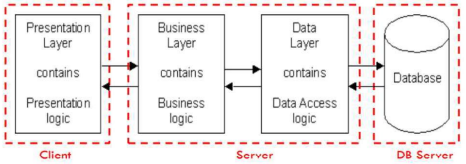
2-tier web architecture

- The database runs on a Server.
- Separated from the client and easy to switch to a different database.
- The presentation and logic layers are still tightly connected (coupled).



3-tier web architecture

- Each layer can potentially run on a different machine.



Thank You !

Questions ?

1. Introduction to web architectures

Terminology

World Wide Web (WWW)

The World Wide Web (WWW) is an Internet-based hypertext project that allows global information sharing. It is a collection of interlinked multimedia documents that are stored on the Internet. A protocol (HTTP) is used to access the documents. Also, it is one way of sharing information via the Internet.

Documents and downloadable media are made available to the network through web servers and can be accessed by programs such as web browsers. Servers and resources on the World Wide Web are identified and located through character strings called uniform resource locators (URLs).

WWW Components

Structural Components

- Clients/browsers – to access and operate on the network
- Servers – run on sophisticated hardware and keep shared resources
- Internet – the global infrastructure which facilitates data transfer

Syntactic Components

- Hyper Text Transfer Protocol (HTTP)
- Hyper Text Markup Language (HTML)
- Uniform Resource Identifiers (URIs)

Client / Web browser

A client is a requesting program in a client/server relationship, e.g., the user of a Web browser is effectively making client requests for pages from servers all over the Web. A web **browser** is a software application for retrieving, presenting, and traversing information resources on the World Wide Web. When it needs to view information from the Internet, First it requires opening a web browser. A web browser is software that is used to access the Internet. A web browser provides access to websites and does activities within them like login, viewing multimedia, linking from one site to another, visiting one page from another, printing, sending, and receiving email, among many other activities.

Search Engine: is also a Computer program that allows users to search information on the web.

Server

In general, a server is a computer program that provides services to other computer programs in the same or other computers. This can also be defined as a network server that manages access to files, folders, and other resources over the Internet or local intranet via HTTP. Web servers handle access permission, execute programs, keep track of directories and files, and communicate with client computers.

Internet

The Internet is the global network that defines how the hardware can be connected or it can be seen as a medium for collaboration and interaction between individuals and their computers regardless of geographic location. That includes everything from the cables that carry terabits of data/information every second to inter-connected computers. All this hardware wouldn't work without the **protocols**. Protocols are sets of rules that machines (devices) follow to complete tasks. There are several protocols on the Internet.

Visit History of the internet to watch about the evolution of internet.

HTTP

Hypertext Transfer Protocol (HTTP) is an application-layer protocol. HTTP is a communication standard governing the requests and responses that take place between the browser running on the end user's computer and the web server.

HTML

HTML is the standard markup language for creating Web pages and describes the structure of a Web page. HTML consists of a series of elements. It tells the browser how to display the content. Label pieces of HTML content such as "this is a heading", "this is a paragraph", "this is a link", etc.

URI

The Uniform Resource Identifier (URI) is intended to identify abstract or physical resources on the Internet. A Uniform Resource Locator (URL), or web address, is the most common form of URI. It is used for unambiguously identifying and locating websites or other web-connected resources.

URL

This is a universal addressing mechanism and it would be impossible to navigate to a site or a page linking would not be feasible without a URL.

URLs have 3 components:

- A Prefix (usually http:// or https:// -- Protocol)
- A Hostname: (such as www.wikipeda.com)
- A Path: (such as /web/itfac/index.htm for the document)

IP address

Every device on a network has a unique identification. As an address of a letter to send in the posted mail, computer devices use a unique identifier to send data to a specific device on a network.

There are two versions of IP addresses:

- IP Version 4 (IPv4)

- IP Version 6 (IPv6).

Numeric IP addresses work very well for computers, but humans find it difficult to use long patterns of numbers. Therefore, humans identify computers using Uniform Resource Locators (URLs) or Web Addresses.

There is a translator to translate the URL to an IP address as computers need to use IP addresses to access websites. This translation happens through the Domain Name System (DNS).

Domain Name System (DNS)

Domain name: The specific address of a computer on the Internet

– microsoft.com

A DNS server is a high-end computer that contains a database of public IP addresses and their associated hostnames (URLs), it serves to resolve or translate, those common names to IP addresses. It is also called as name server.

If you are using an Internet Service Provider (ISP), your DNS server is at your ISP.

Click Here **for further details.**

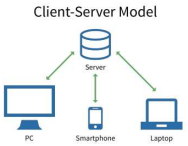
Introduction to Web Architecture

The architecture of an application refers to the overall structure of an application, including the way it is designed, implemented, and deployed. The quality of the architecture often directly impacts the quality of the finished product. Ideally, a completed project allows users to access information easily and understand how to navigate its content.

Web architecture is important because it plays a crucial role in the performance, scalability, and maintenance of a website or web application. Additionally, effective web architecture can ensure adaptability if an employer requests major changes to a project that's already underway. There are many different approaches, strategies, and patterns of web application architecture.

Client Server Architecture

It is a software architecture model consisting of two parts, client systems, and server systems both communicating over a computer network or on the same computer.



- **The client** is defined as the requester of services and the server is defined as the provider of services

  Characteristics of a client

  - **Initiates requests(Master)**
  - **Waits for and receives replies**
  - **Usually connects to a small number of servers at one time**
  - **Typically interacts directly with end-users using a graphical user interface**

  Examples: web browsers, email clients, and online chat clients

- **A Server** is a host that is running one or more server programs that can share information or resources.

  Characteristics of a server

  - **Passive (slave)**
  - **Waits for requests from clients**
  - **Upon receipt of requests, processes them and then serves replies**
  - **Usually accepts connections from a large number of clients**
  - **Typically does not interact directly with end-users**

  Examples: web servers, database servers, and mail servers

Multi-tier Architecture

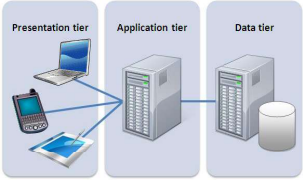
In software engineering, multi-tier architecture (often referred to as n-tier architecture) is a client–server architecture in which presentation, application processing, and data management functions are logically separated.

The most widespread use of multi-tier architecture is the three-tier architecture.

N-tier application architecture provides a model by which developers can create flexible and reusable applications. By segregating an application into tiers, developers acquire the option of modifying or adding a specific layer, instead of reworking the entire application.

Three tier architecture

Three-tier architectures typically comprise a presentation tier, a business or data access [logic] tier, and a data tier.



Presentation tier

The presentation tier is the user interface and communication layer of the application, where the end-user interacts with the application. Its main purpose is to display information to and collect information from the user. This top-level tier can run on a web browser, as a desktop application, or as a graphical user interface (GUI). Web presentation tiers are usually developed using HTML, CSS, and JavaScript. Desktop applications can be written in a variety of languages depending on the platform.

Application Tier

The application tier, also known as the logic tier or middle tier, is the heart of the application. In this tier, information collected in the presentation tier is processed - sometimes against other information in the data tier - using business logic, a specific set of business rules. The application tier can also add, delete, or modify data in the data tier. The application tier is typically developed using Python, Java, Perl, PHP, or Ruby, and communicates with the data tier using API calls.

Data Tier

The data tier, sometimes called database tier, data access tier, or back-end, is where the information processed by the application is stored and managed. This can be a relational database management system such as PostgreSQL, MySQL, Maria DB, Oracle, DB2, Informix, or Microsoft SQL Server, or a NoSQL Database server such as Cassandra, Couch DB, or MongoDB.

Benefits of three-tier architecture

Each tier can run on a separate operating system and server platform - e.g., web server, application server, database server - that best fits its functional requirements. Each tier runs on at least one dedicated server hardware or virtual server, so the services of each tier can be customized and optimized without impacting the other tiers.

Other benefits (compared to single- or two-tier architecture) include:

1. **Faster development**: Because each tier can be developed simultaneously by different teams, an organization can bring the application to market faster, and programmers can use the latest and best languages and tools for each tier.
2. **Improved scalability**: Any tier can be scaled independently of the others as needed.
3. **Improved reliability**: An outage in one tier is less likely to impact the availability or performance of the other tiers.

4. **Improved security**: Because the presentation tier and data tier can't communicate directly, a well-designed application tier can function as a sort of internal firewall, preventing SQL injections and other malicious exploits.

Web-oriented three-tier architecture

Any application that uses Web Technologies including web browsers, web servers, and Internet protocols is called a **Web Application**. Web application architecture defines the interactions between applications, middleware systems, and databases. A web application can be divided into three different layers.

Presentation tier

- Browser/custom client
- Client Side Scripting (JavaScript)
- Applets.

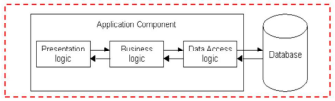
Logical tier

- Web Server (Apache, IIS, WebSphere etc.)
- Scripting Languages (PHP, Perl, etc.)
- Programming Languages (Java, C, C# etc.)
- Application Frameworks (Ruby on Rails etc.)

Data tier

- Database Management System (DBMS) (Oracle, MySQL, SQL Server, DB2, etc.)
- XMLD

1-tier architecture



All 3 layers are on the same machine. All code and processing are kept on a single machine. Presentation, Logic, and Data layers are tightly connected.

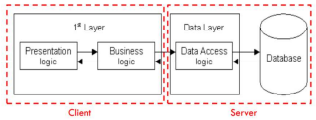
Limitations

Scalability issue. A single processor means hard to increase the volume of processing

Portability is less. Moving to a new machine may mean rewriting everything.

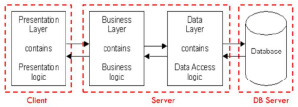
Hard to maintain. Changing one layer requires changing other layers

2-tier architecture



The database runs on a Server. Separated from the client and easy to switch to a different database. The presentation and logic layers are still tightly connected (coupled). Heavy load on the server and hence potential congestion on the network. The presentation is still tied to business logic.

3-tier architecture



Each layer can potentially run on a different machine. Presentation, logic, data layers disconnected.

HNDIT3022
Web Programming

Week2:- Introduction to server-side scripts

Scripting Language Vs Programming Language

- Programming language
  - Has all the features needed to develop complete applications.
  - The code has to be compiled before it can be executed
- Scripting language
  - Mostly used for routine tasks
  - The code is usually executed without compiling
  - Is usually embedded into other software environments

Scripting Language

- A script is a set of programming instructions that is interpreted at runtime.
- They do not require the compilation step and are rather interpreted.
- A scripting language is a programming language designed for integrating and communicating with other programming languages.
- The purpose of the scripts is usually to enhance the performance or perform routine tasks for an application.

Activity 2.1

- Discuss Different Scripting Languages.
- Ex:-
  - Active Server Pages (ASP)
  - Perl
  - PHP (Hypertext Pre-Processor)
  - JSP(Java Server Pages)
  - JavaScript
  - Etc..
- Click here for more details

What is Client-Side Scripting Language?

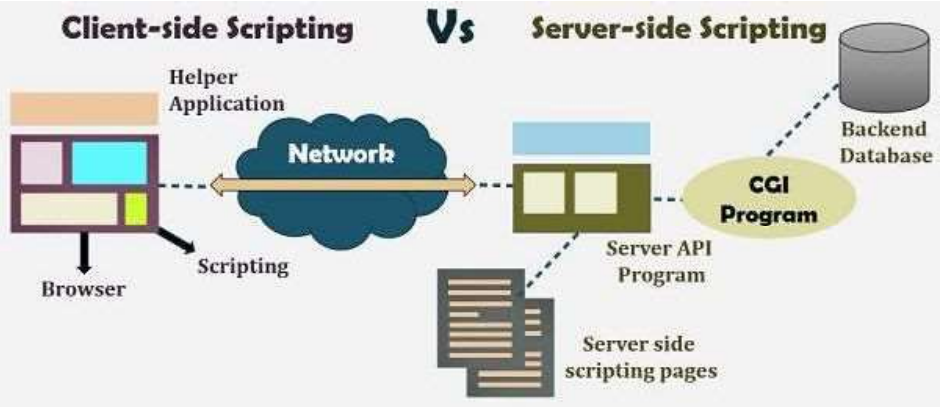
- Web browsers execute client-side scripting.
- Source code is used to transfer from the web server to the user's computer over the internet and run directly on browsers.
- It cannot be basically used to connect to databases on a web server.
- It allows for more interactivity.
- It is also used for validations and functionality for user events.
- For example
  - Client-side scripting could check the user's form for errors before submitting it.
- Example Languages:
  - Java script

Server-Side Scripting Languages[2]

- Server-Side Scripting Languages
  - Customize a web page and dynamically change its contents.
  - Respond to queries from users or from HTML forms.
  - Access the database and send the information back to the browser.

What is Server-Side Scripting Language

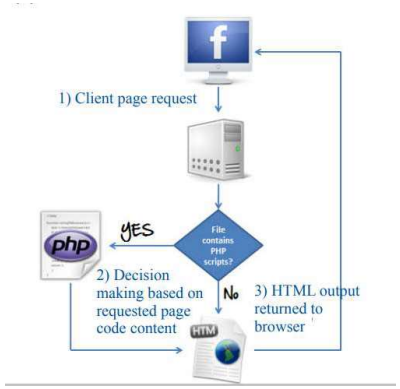
- Server-side scripting is a method of programming for the web that runs software on the server rather than the browser or installed plugins to create dynamic web pages.
- Example Languages
  - Perl, PHP, JSP, Ruby, ColdFusion, and Python



PHP

Server Side scripting Language

How PHP works?



What is PHP?

- PHP is an acronym for "PHP: Hypertext Preprocessor".
- PHP is a widely used open-source general-purpose server-side scripting language.
- PHP scripts can only be interpreted on a server that has PHP installed.
- It is especially suited for web development and can be embedded into HTML.
- PHP is free to download and use

Setting Up a Development Server

- In order to develop and run PHP Web pages three vital components need to be installed on your computer system.
  - Web Server – The most often used is the freely available Apache Server.
  - PHP Parser – In order to process PHP script instructions a parser must be installed to generate HTML output that can be sent to the Web Browser.
  - Database(Optional) – The most commonly used is the freely available MySQL database.

Setting Up a Development Server[2]

- There are several software come in the form of a package that binds the bundled programs together.
- So that you don't have to install and set them up separately.
- Example:
  - WAMP (Windows, Apache, MySQL, and PHP)
  - XAMPP(Cross Platform, Apache, Maria DB/MySQL, PHP, Perl)
  - LAMP(Linux, Apache, MySQL, and PHP)
  - MAMP (Mac, Apache, MySQL, and PHP)

PHP Syntax

- PHP is a case-sensitive language, "VAR" is not the same as "var".
- The PHP tags themselves are not case-sensitive, but it is strongly recommended that we use lowercase letters.
  - Ex:- <?php … ?>

  The opening <?php tells the webserver to allow the PHP program to interpret all the following code up to the ?> tag
- PHP file that contains PHP tags and ends with the extension ".php".

Activity 2.2

- Set up your own development environment using the suitable software application discussed above. (WAPM, XAMPP, etc..)

PHP Syntax[2]

- A PHP script starts with <?php and ends with ?>:
- A PHP script can be placed anywhere in the document.
- A PHP file can also contain tags such as HTML and client-side scripts such as JavaScript.

Hello World in PHP

- The program shown below is a basic PHP application that outputs the words “Hello World!” When viewed in a web browser.

```
<?php
    echo “Hello World”;
?>
```

echo and print

- echo and print are both used to output data to the screen.
- The echo statement can be used with or without parentheses:

  echo or echo()

- The print statement can be used with or without parentheses:

  print or print()

PHP Example

```
<!DOCTYPE html>
<html>
<body>

<h1>My first PHP page</h1>

<?php
echo "Hello World!”;
print "Hello World”;
?>

</body>
</html>
```

HTML Code

PHP Code

HTML Code

```
Output -
Hello World!
Hello World
```

echo vs print

- The differences are small:
  - echo has no return value while print has a return value of 1 so it can be used in expressions.
  - echo can take multiple parameters (although such usage is rare) while print can take one argument.
  - echo is marginally faster than print.

Activity 2.3

- Get the output of following code segment.

```php
<?php

    echo "This is", "How","multiple arguments", "works", "in PHP echo</br>";

    print "Print is not supportng for multle parameters";
?>
```

Questions ?

Comments

- // or /*.....*/ can be used to get a comment.

```php
<?php

        // this s a comment
        /*this is also a comment
        runs on multiple lines */

?>
```

Thank You !

HNDIT3022
Web Programming

Week3:- Basics of PHP

PHP variables

- You must place a $ in front of all variables.
- PHP automatically determines variable type by the context in which it is being used.
- PHP supports a number of different variable types:
  - integers , floating point numbers, strings and arrays
- Variable names in PHP are case-sensitive.
- Example
  - $variable_name;

Variables

Variable naming conventions

- There are a few rules that you need to follow when choosing a name for your PHP variables.
  - PHP variables must start with a letter or underscore “_".
  - PHP variables may only be comprised of alpha-numeric characters and underscores. a-z, A-Z, 0-9, or _ .
  - Variables with more than one word should be separated with underscores
    - $my_variable
  - Variables with more than one word can also be distinguished with capitalization.
    - $myVariable
    - $name
    - $INCOME
    - $_123

PHP Data types

- PHP is a loosely typed language; it does not have explicitly defined data types.
- PHP determines the data types by analyzing the attributes of the data supplied.
  - Alphanumeric characters are classified as strings
  - Whole numbers are classified as integers
  - Numbers with decimal points are classified as floating points.
  - True or false values are classified as Boolean.

Constants

- A constant is a variable whose value cannot be changed at runtime.
- Suppose we are developing a program that uses the value of PI 3.14, we can use a constant to store its value.
  - define('PI',3.14);
- define is a predefined function that is used to create constants in PHP.

Variable assignment

- The syntax to assign a value to a variable is always

  variable = value.

- Or, to reassign the value to another variable, it is other

  variable = variable

Operators

Operators

- Operators are the mathematical, string, comparison, and logical commands such as plus, minus, multiply, and divide.

- Types of operators
  - Arithmetic operators
  - Assignment operators
  - Comparison operators
  - Logical operators

Assignment Operators

Operator	Example	Equivalent to
=	$a=10	$a=10
+=	$a+=5	$a =$a+5
-=	$a-=2	$a=$a-2
*=	$a*=10	$a=$a*10
/=	$a/=5	$a=$a/5
%=	$a%=3	$a=$a%3
.=	$a.="A"	$a=$a . "A"

Arithmetic Operators

Operator	Description	Example
+	Additon	$a+1
-	Substraction	$a-6
*	Multiplicaton	$a*40
/	Division	$a/5
%	Modulus(Division Reminder)	$a%2
++	Increment	$a++ or ++$a
--	Decrement	$a– or --$a

Comparison operators

Operator	Description	Example
==	Is equal to	$a == 5
!=	Is not equal to	$a!=5
>	Is greater than	$a>10
<	Is less than	$a<10
>=	Is greater than or equal to	$a>=10
<=	Is less than or equal to	$a<=10

Logical Operators

Operator	Description	Example
&&	And	$a==3 && $b==2
and	Low-precedence and	$a==3 and $b==2
||	Or	$a==3 || $b==2
or	Low-precedence or	$a==3 or $b==2
!	Not	!($a==$b)
xor	Exclusive or	$a xor $b

Thank You !

Questions ?

HNDIT3022
Web Programming

Week4:- Expressions & Control flow in PHP

Example

```php
<?php

    $a=10; //is an expression

    $b=12;
    $c=5;

    $a=$b+$c; //is an expression

    $ans=sqrt(100); // is an expression

?>
```

Expressions

- An expression is a combination of values, variables, operators, and functions that results in a value.
- The simplest form of an expression is a literal, which simply means something that evaluates to itself.
  - Example:- 73 , "Hello".
- An expression could also simply be a variable, which evaluates to the value that has been assigned to it.
- They are both types of expressions, because they return a value.

Conditionals

Control Structures

Conditional

- Conditionals alter program flow. They enable you to ask questions about certain things and respond to the answers you get in different ways.
- There are three types of non-looping conditionals:
  - The **if** statement
  - The **switch** statement
  - The **?** operator.

The if statement[2]

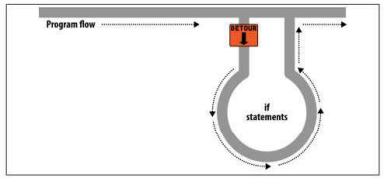
- The if statement executes some code if one condition is true.

- Syntax

  if(*condition*){
  code to be executed if condition is true;
  }

The if statement

- In the case of an if statement, you could imagine coming across a detour sign that you have to follow.
- If a certain condition is TRUE you drive off and follow the detour until you return to where it started and then continue on your way in your original direction.
- Or,
- If the condition isn't FALSE, you ignore the detour and carry on driving



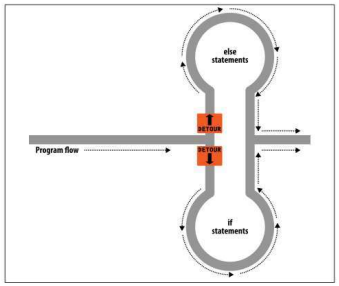
Example

```php
<?php

    $t = date("H");

    if ($t < "20") {
        echo "Have a good day!";
    }
?>
```

date() is a predefined function which formats a local date and time, and returns the formatted date string.

else statement

- Sometimes when a conditional is not TRUE, you may not want to continue on to the main program code immediately but might wish to do something else instead.
- This is where the else statement comes in



Example

```php
<?php
        $num1=10;
        $num2=20;

        if($num1>$num2){
            echo "$num1 is greater than $num2";
        }else{
            echo "$num2 is greater than $num1";
        }
?>
```

else statement[2]

- With an if ... else statement, the first conditional statement is executed if the condition is TRUE.
- But if it's FALSE, the second one is executed

- Syntax

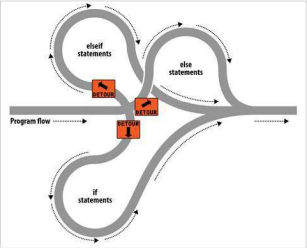
  if(condition){
  }
  else{
  }

Activity 4.1

- Write down a PHP code segment to display whether a given number is odd or even.

elseif Statement

- There are also times when you want a number of different possibilities to occur, based upon a sequence of conditions.
- You can achieve this using the elseif statement.



Example

```php
<?php
        $num1=10;
        $num2=20;

        if($num1>$num2){
            echo "$num1 is greater than $num2";
        }elseif($num1==$num2){
            echo "$num1 is equal to $num";
        }else{
            echo "$num2 is greater than $num1";
        }
?>
```

elseif Statement[2]

- You may have as many elseif statements as you like.
- But as the number of elseif statements increases, you would probably be better advised to consider a switch statement if it fits your needs.
- Syntax

```
if(condition){

}elseif(condition){

}else{

}
```

The switch Statement

- The switch statement is useful in cases in which one variable or the result of an expression can have multiple values, which should each trigger a different function.
- Syntax

```
switch(value to check){
    case value1:
        //statement1;
        break;
    ...........
    ...........
    ............
    default:
        //statement default;
        break;
}
```

OR

```
switch(value to check):
    case value1:
        //statement1;
        break;
    ...........
    ...........
    ............
    default:
        //statement default;
        break;
endswitch
```

Example

```php
<?php
  switch ($page)
  {
    case "Home":
        echo "You selected Home";
        break;
    case "About":
        echo "You selected About";
        break;
    case "News":
        echo "You selected News";
        break;
    case "Login":
        echo "You selected Login";
        break;
    case "Links":
        echo "You selected Links";
        break;
  }
?>
```

The ? Operator(ternary operator)

- Compact version of if…else
- The ? operator is passed an expression that it must evaluate, along with two statements to execute:
  - one for when the expression evaluates to TRUE,
  - the other for when it is FALSE.

Activity 4.2

- Write down a PHP code segment to print the weekday name when day number is given. (use date("N") to get the day number

Example

```php
<?php

    $enough= $fuel <= 1 ? "Fill tank now" : "There's enough fuel";

?>
```

Activity 4.3

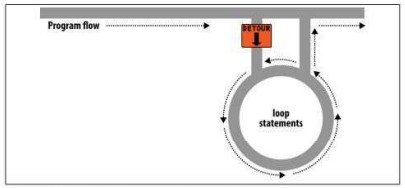
- Write down a PHP code to print a given number is Odd or Even using ternary operator.

Looping

- One of the great things about computers is that they can repeat calculating tasks quickly and tirelessly.

- Often you may want a program to repeat the same sequence of code again and again until something happens, such as a user inputting a value or reaching a natural end.

- PHP's various loop structures provide the perfect way to do this



Types of loops

- While loop
- Do….while loop
- For loop

Looping

While Loop

- Syntax

```
while(condition){

    //statement
    increment/decrement

}
```

Example

```
<?php

    $num=1;

    while($num<=5){
        echo "Hello<br/>";
        $num++;
    }
?>
```

Activity 4.4

- Write down a PHP code segment to display the numbers from 10-1.

do……while loop

- Syntax

```
do{
    //statement
    increment/decrement
}while(condition);
```

Example

```php
<?php

    $num=1;

    do{
        echo "Hello<br/>";
        $num++;

    }while($num<5);
?>
```

For loop

- The for loop, is also the most powerful, as it combines the abilities to set up variables as you enter the loop, test for conditions while iterating loops, and modify variables after each iteration.

- Syntax

  for(initialization; condition; increment/decrement){
  //statement
  }

Activity 4.6

- Write down a PHP code to display the numbers from 1-10

Example

```php
<?php

    for($num=1; $num<=5; $num++){
        echo "Hello<br/>";
    }
?>
```

Activity 4.6

- Write a small PHP code to print multiplication table of 5.

5 times 1= 5

5 times 2=10

……

……

……

Break

- Just as you saw how to break out of a switch statement, you can also break out of a for loop using the same break command.

```
<?php

    for($i=1; $i<10; $i++){
        if($i==5){
            break;
        }
        echo $i."<br/>";
    }
?>
```

For loop[2]

- A more complex form of the for statement even lets you perform multiple operations in each of the three parameters

- Example

```
<?php

    for ($i = 3, $j = 2 ; $i + $j < 10 ; $i++ , $j++){
        echo $i." ";
        echo $j."<br/>";
    }
?>
```

Continue

- The continue statement is a little like a break statement, except that it instructs PHP to stop processing the current loop and to move right to its next iteration.

```
<?php

    for($i=1; $i<10; $i++){
        if($i==5){
            continue;
        }
        echo $i."<br/>";
    }
?>
```

Questions ?

Thank You !

HNDIT3022
Web Programming

Week5:- PHP Arrays

Arrays[2]

- Create an array

```php
<?php
    $student_list[]="Kamal";
    $student_list[]="Sanduni";
    $student_list[]="Kusum";
    $student_list[]="Namal";
    $student_list[]="Geetha";
?>
```

Arrays

- An array is a special variable, which can hold more than one value at a time.
- They are like bead strings because each element has its own location and (with the exception of the first and last ones) each has other elements on either side.
- Some arrays are referenced by numeric indices; others allow alphanumeric identifiers.
- Built-in functions let you sort them, add or remove sections, and walk through them to handle each item through a special kind of loop.
- By placing one or more arrays inside another, you can create arrays of two, three, or any number of dimensions.

Arrays[3]

- Another way to create an by using array() function

```php
<?php
    $student_list=array("Kamal","Sanduni","Kusum","Namal","Geetha");
?>
```

Numerically Indexed Arrays

- Arrays which has numerical indices called as Numeric Arrays.

indexes	0	1	2	3	4
	20	50	34	56	100

Activity 5.1

```
<?php
    $student_list=array("Kamal","Sanduni","Kusum","Namal","Geetha");

    echo $student_list[0];
?>
```

- Output?

Example

```
<?php
    $student_list=array("Kamal","Sanduni","Kusum","Namal","Geetha");
    print_r($student_list);
?>
```

The print_r() function prints the information about a variable in a more human-readable way.

Output:

Array ([0] => Kamal [1] => Sanduni [2] => Kusum [3] => Namal [4] => Geetha)

Activity 5.2

```
<?php
    $student_list=array("Kamal","Sanduni","Kusum","Namal","Geetha");
    $student_list[2]="Shanthi";

    echo $student_list[2];
?>
```

- Output?

Activity 5.3

```php
<?php
    $student_list=array("Kamal","Sanduni","Kusum","Namal","Geetha");
    $student_list[]="Shanthi";

    print_r($student_list);
?>
```

- Output?

Associative Arrays

- Arrays which can contain alphanumeric identifiers.

indexes	m1	m2	m3	m4	m5
	20	50	34	56	100

- You can reference the items in an array by name rather than by number.

foreach…as loop

```php
<?php
    $student_list=array("Kamal","Sanduni","Kusum","Namal","Geetha");

    foreach($student_list as $student){
        echo $student."<br/>";
    }
?>
```

- Output

Kamal
Sanduni
Kusum
Namal
Geetha

Example

```php
<?php
    $student_list=array("Name"=>"Kamal", "Age"=>58);

    foreach($student_list as $index=>$value){
        echo $index." ".$value."<br/>";
    }
?>
```

Activity 5.4

- Create an associative array to store marks for three subjects named English, Maths, and Science of a student.

2D Array

- A two-dimensional array is an array of arrays.
- Let's assume you want to create an array to store the marks of two students for three subjects.

```php
<?php
    $student_list=array(
                        array("Kamal", 75, 89,98),
                        array("Shantha",70, 65,67)
                        );

?>
```

Multidimensional arrays

- A simple design feature in PHP's array syntax makes it possible to create arrays of more than one dimension.
- That feature makes it possible to include an entire array as a part of another one and to be able to keep doing so.
- Array is an array containing one or more arrays.
- The dimension of an array indicates the number of indices you need to select an element.
  - For a two-dimensional array you need two indices to select an element
  - For a three-dimensional array you need three indices to select an element

Example

```php
<?php
    $student_list=array(
                        array("Kamal", 75, 89,98),
                        array("Shantha",70, 65,67)
                        );

    foreach($student_list as $student){
        foreach($student as $value){
            echo $value." ";
        }
        echo "<br/>";
    }

?>
```

- Output

Kamal 75 89 98
Shantha 70 65 67

2D Array[2]

- Let's modify the code to use as an Associative array.

```php
<?php
    $student_list=array(
                        "Kamal"=> array("Name"=>"Kamal", "Science"=>75, "Maths"=>89, "English"=>98),
                        "Nimal"=> array("Name"=>"Nimal", "Science"=>70, "Maths"=>65, "English"=>67)
                        );

    echo $student_list["Kamal"]["Science"];
?>
```

- You can directly access a particular element of the array using square brackets, like above example shows.

Answer

```php
<?php
    $student_list=array(
                        "Kamal"=> array("Name"=>"Kamal", "Science"=>75, "Maths"=>89, "English"=>98),
                        "Nimal"=> array("Name"=>"Nimal", "Science"=>70, "Maths"=>65, "English"=>67)
                        );

    echo "<table border=1>";
    echo "<tr><th>Name</th><th>Science</th><th>Maths</th><th>English</th></tr>";
    foreach($student_list as $student_index=>$student){
        echo "<tr>";
        foreach($student as $index=>$value){
            echo "<td>$value</td>";
        }
        echo "</tr>";
    }
?>
```

Activity 5.4

- Modify the above code to get the following output.

Name	Science	Maths	English
Kamal	75	89	98
Nimal	70	65	67

Array functions

- is_array()
- count()
- sizeof()
- sort()
- rsort()
- explode()
- implode()
- array_merge()
- array_slice()
- array_fill()

Activity 5.5

- Discuss about the usage of the functions listed above.

Thank You !

Questions ?

HNDIT3022
Web Programming

Week6:- File Handling

Checking Whether a File Exists

- To determine whether a file already exists, you can use the file_exists function, which returns either TRUE or FALSE.

```php
<?php
    if (file_exists("testfile.txt"))
        echo "yes";
    else{
        echo "No";
    }

?>
```

- We have not created any file yet. So above code will display No.

Introduction

- Powerful as it is, MySQL is not the only (or necessarily the best) way to store all data on a web server.
- Sometimes it can be quicker and more convenient to directly access files on the hard disk.
- Cases in which you might need to do this are modifying images such as uploaded user avatars, or log files that you wish to process

fopen()

- fopen(filepath, mode) – Create a file if the file is not exist otherwise open in given mode.

Mode	Action	Descripton
'r'	Read from file start.	Open for reading only; place the file pointer at the beginning of the file. Return FALSE if the file doesn't already exist.
'r+'	Read from file start and allow writing.	Open for reading and writing; place the file pointer at the beginning of the file. Return FALSE if the file doesn't already exist.
'w'	Write from file start and truncate file.	Open for writing only; place the file pointer at the beginning of the file and truncate the file to zero length. If the file doesn't exist, attempt to create it.
'w+'	Write from file start, truncate file, and allow reading.	Open for reading and writing; place the file pointer at the beginning of the file and truncate the file to zero length. If the file doesn't exist, attempt to create it.
'a'	Append to file end.	Open for writing only; place the file pointer at the end of the file. If the file doesn't exist, attempt to create it.
'a+'	Append to file end and allow reading	Open for reading and writing; place the file pointer at the end of the file. If the file doesn't exist, attempt to create it

Activity 6.1

- let's create it and write a few lines to it.

```
<?php
    $file=fopen("textfile.txt", "w")
        or die("Couldn't open the file");
    fwrite($file, "This is a sample text")
        or die("couldn't write into fle");
    fclose($file);
    echo "written successfully";
?>
```

File reading[2]

- You can retrieve multiple lines or portions of lines through the fread() function
- The fread() function is commonly used with binary data.
- If you use it on text data that spans more than one line, remember to count newline characters.

File reading

- The easiest way to read from a text file is to grab a whole line through fgets().

```
<?php
    $file=fopen("textfile.txt", "r")
        or die("Couldn't open the file");
    while(!feof($file)){ //if not in the endof file
        $line=fgets($file);
        echo $line;
    }
    fclose($file);
?>
```

Activity 6.2

- Let us try to read a file using fread().

```
<?php
    $file=fopen("textfile.txt", "r")
        or die("Couldn't open the file");
    $text=fread($file, 4);
    fclose($file);
    echo $text;
?>
```

Updating file

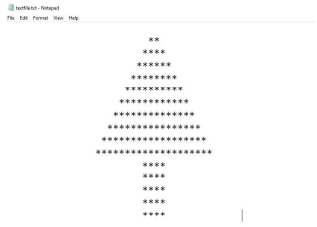
- Often, you will want to add more data to a saved file, which you can do in many ways.
- You can use one of the append write modes

or

- you can simply open a file for reading and writing with one of the other modes that supports writing, and move the file pointer to the correct place within the file that you wish to write to or read from.

Activity 6.3

- Create a file with the following content.



- Display the pattern as it is in a web page.

Updating file[2]

- Lets use the append write mode.

```
<?php
    $file=fopen("textfile.txt", "a")
        or die("Couldn't open the file");
    fwrite($file, "This text is appending");
    fclose($file);
?>
```

Answer

- Using fgets()

```
<pre>
<?php
    $file=fopen("textfile.txt", "r")
        or die("Couldn't open the file");

    while(!feof($file)){
        echo fgets($file);
    }
    fclose($file);
?>
</pre>
```

- Using file_get_content()

```
<pre>
    <?php
        echo file_get_contents("textfile.txt");
    ?>
</pre>
```

File uploading[1]

- $_FILES[][] is an associative array which used to access the files.
- Different indexes we can use in $_FILES as follows.

$_FILES['the_file']['name']	This is the name of the actual file
$_FILES['the_file']['size']	This is the size of the file in bytes
$_FILES['the_file']['tmp_name']	This is the a temporary file that resides in the tmp directory of the server
$_FILES['the_file']['type']	This gets the file extension from the file name

Activity 6.4

- Let's get the details of a file to be upload.

```php
<?php
    $fileName=$_FILES["addImages"]["name"];
    $fileType=$_FILES["addImages"]["type"];
    $tmpName=$_FILES["addImages"]["tmp_name"];
    $fileSize=$_FILES["addImages"]["size"];

    echo $fileName."</br>";
    echo $fileType."</br>";
    echo $tmpName."</br>";
    echo $fileSize."</br>";
?>
```

Choose File No file chosen Insert

WhatsApp Image 2023-10-07 at 23.20.02.jpeg
image/jpeg
C:\xampp\tmp\phpAB0D.tmp
151268

File uploading[2]

- You need to upload a file from a form chosen by a special type of encoding called multipart/form-data.

```php
<form action="fileUpload.php" method="post" enctype="multipart/form-data">
    <input type="file" name="addImages">

    <input type="submit" name="submit" value="Insert">

</form>
```

- Use post as a method.

File uploading[3]

- Syntax

  move_uploaded_file(Temporary file, Target_file)

*Target_file specifies the path of the file to be uploaded

```php
<?php
    $fileName=$_FILES["addImages"]["name"];
    $fileType=$_FILES["addImages"]["type"];
    $tmpName=$_FILES["addImages"]["tmp_name"];
    $fileSize=$_FILES["addImages"]["size"];

    move_uploaded_file($tmpName, "Images/".$fileName);

?>
```

Activity 6.5

- Modify the above code to check whether the file already exists in the given location before uploading the file.

File uploading[4]

- Now Let's try to limit the file types.

```php
<?php
    $fileName=$_FILES["addImages"]["name"];
    $fileType=$_FILES["addImages"]["type"];
    $tmpName=$_FILES["addImages"]["tmp_name"];
    $fileSize=$_FILES["addImages"]["size"];

    $allowed_types = array("jpg" => "image/jpg",
                           "jpeg" => "image/jpeg",
                           "gif" => "image/gif",
                           "png" => "image/png");

    $targetFile="Images/".$fileName;

    if(in_array($fileType, $allowed_types )){
        move_uploaded_file($tmpName, $targetFile);
        echo "File upload successfully";
    }else{
        echo "File type is Not Accepted";
    }

?>
```

File uploading[3]

- Let's limit the file size to be uploaded.

```php
<?php
    $fileName=$_FILES["addImages"]["name"];
    $fileType=$_FILES["addImages"]["type"];
    $tmpName=$_FILES["addImages"]["tmp_name"];
    $fileSize=$_FILES["addImages"]["size"];

    $fileLimit=2*1024*1024; //2MB
    $targetFile="Images/".$fileName;


    if($fileSize<=$fileLimit){
        move_uploaded_file($tmpName, $targetFile);
        echo "File upload successfully";
    }else{
        echo "File size exceeded";
    }

?>
```

Activity 6.6

- Now let us write down the complete code segment to upload files when the following conditions are satisfied.
  - Not exist currently in the folder
  - File size should be less than 2MB
  - Allowed only image type files.

Answer

```php
<?php
    $fileName=$_FILES["addImages"]["name"];
    $fileType=$_FILES["addImages"]["type"];
    $tmpName=$_FILES["addImages"]["tmp_name"];
    $fileSize=$_FILES["addImages"]["size"];

    $allowed_types = array("jpg" => "image/jpg",
                           "jpeg" => "image/jpeg",
                           "gif" => "image/gif",
                           "png" => "image/png");

    $fileLimit=2*1024*1024; //2MB
    $targetFile="Images/".$fileName;

    if(!file_exists($targetFile)){
        if(in_array($fileType, $allowed_types )){
            if($fileSize<=$fileLimit){
                move_uploaded_file($tmpName, $targetFile);
                echo "File upload successfully";
            }else{
                echo "File size exceeded";
            }
        }else{
            echo "File type is Not Accepted";
        }
    }else{
        echo "File is already Uploded";
    }

?>
```

Thank You !

Questions ?

HNDIT3022
Web Programming

Week7:-PHP functions & Superglobals

PHP functions

Introduction

- A function is a set of statements that perform a particular function and optionally returns a value.
- You can pull out a section of code that you have used more than once, place it into a function, and call the function by name when you want the code.

PHP functions

- PHP comes with hundreds of ready-made, built-in functions.
- To use a function, call it by name.
- For example, you can see the print function in action here:

```
print("print is a pseudo-function");
```

- The parentheses tell PHP that you're referring to a function.
- Besides the built-in PHP functions, it is possible to create your own functions.(User defined fuctions)

PHP functions[2]

- The general syntax for a function is:

```
function function_name([parameter [, ...]])
 {
     // Statements
 }
```

- A definition starts with the word function.
- A name follows, which must start with a letter or underscore, followed by any number of letters, numbers, or underscores.
- The parentheses are required.
- One or more parameters, separated by commas, are optional

Activity 7.2

- Write down a PHP function to print the sum of two numbers when two numbers are passed as parameters.

Activity 7.1

- Write down a PHP function to print “Hello world”

PHP function[3]

- Returning values

```
function function_name([parameter [, ...]])
 {
     // Statements
    return value;
 }
```

- A definition starts with the word function.
- A name follows, which must start with a letter or underscore, followed by any number of letters, numbers, or underscores.
- The parentheses are required.
- One or more parameters, separated by commas, are optional
- Return keyword with the return value

Activity 7.3

- Write down a PHP function to return the sum of two numbers when two numbers are passed as parameters.

Superglobals

Introduction

- Superglobals were introduced in PHP 4.1.0, and are built-in variables that are always available in all scopes.
- The PHP superglobal variables are:
  - $GLOBALS
  - $_SERVER
  - $_REQUEST
  - $_POST
  - $_GET
  - $_FILES
  - $_ENV
  - $_COOKIE
  - $_SESSION

$_SERVER

- $_SERVER is a PHP super global variable which holds information about headers, paths, and script locations.
- Example
  - $_SERVER['PHP_SELF'] Returns the filename of the currently executing script
  - $_SERVER['SERVER_NAME'] Returns the name of the host server.
  - $_SERVER['REQUEST_METHOD'] Returns the request method used to access the page (such as POST)
  - $_SERVER['SERVER_PORT'] Returns the port on the server machine being used by the web server for communication (such as 80)

Activity 7.4

```php
<?php
    echo $_SERVER["PHP_SELF"];
    echo "</br>";
    echo $_SERVER["SERVER_NAME"];
    echo "</br>";
    echo $_SERVER["SERVER_PORT"];
    echo "</br>";
    echo $_SERVER["REQUEST_METHOD"];
    echo "</br>";
?>
```

/Examples/index.php
localhost
80
GET

Activity 7.5

- Try out the following code segment.

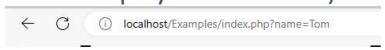
```html
<html>
    <head>

    </head>
    <body>
        <a href="index.php?name=Tom">Click Here</a>
        </br>
        <?php
        echo $_GET["name"];
        ?>
    </body>
</html>
```

$_GET

- $_GET is used to collect form data after submitting an HTML form with method="get".
- $_GET can also collect data sent in the URL.

Cont..

- Information sent from a form with the GET method is **visible to everyone** (all variable names and values are displayed in the URL).



- GET also has limits on the amount of information to send. The limitation is about 2000 characters.
- However, because the variables are displayed in the URL, it is possible to bookmark the page. This can be useful in some cases.
- GET may be used for sending non-sensitive data.

$_POST

- $_POST is used to collect form data after submitting an HTML form with method="post“.

- Example

```
<html>
    <head>
    </head>
    <body>
        <form action="<?php echo $_SERVER['PHP_SELF']?>" method="post">
            Enter Your Name:- <input type="text" name="username">
            <input type="submit" value="Submit">
        </form>
        <?php
            echo $_POST["username"];
        ?>
    </body>
</html>
```

Enter Your Name:- Submit

Tom

Cont..

- Information sent from a form with the POST method is invisible to others (all names/values are embedded within the body of the HTTP request) and has no limits on the amount of information to send.

- POST supports advanced functionality such as support for multi-part binary input while uploading files to the server.

- However, because the variables are not displayed in the URL, it is not possible to bookmark the page.

- **Developers prefer POST for sending form data.**

Activity 7.6

- Create a login form using HTML.
- Write a PHP code segment to capture the user-entered username & password.

Cookies & Session

Cookies

- A cookie is an item of data that a web server saves to your computer's hard disk via a web browser.
- It can contain almost any alphanumeric information (as long as it's under 4 KB)
- Can be retrieved from your computer and returned to the server.
- Because of their privacy implications, cookies can be read only from the issuing domain.

Cookies[2]

- Common uses include
  - session tracking
  - maintaining data across multiple visits
  - holding shopping cart contents
  - storing login details

Cookies[3]

- You can call the setcookie function

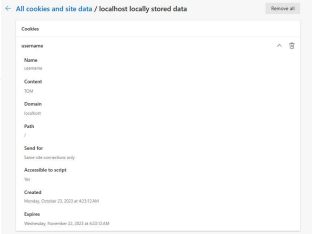
  setcookie(name, value, expire, path, domain, secure, httponly);

Parameter	Description	Example
name	The name of the cookie. This is the name that your server will use to access the cookie on subsequent browser requests.	username
value	The value of the cookie, or the cookie's contents. This can contain up to 4 KB of alphanumeric text.	Tom
expire	(Optional.) Unix timestamp of the expiration date. Generally, you will probably use time() plus a number of seconds. If not set, the cookie expires when the browser closes.	time() + 2592000
path	(Optional.) The path of the cookie on the server. If this is a / (forward slash), the cookie is available over the entire domain, such as www.webserver.com. If it is a subdirectory, the cookie is available only within that subdirectory. The default is the current directory that the cookie is being set in, and this is the setting you will normally use.	/
domain	(Optional.) The Internet domain of the cookie. If this is .webserver.com, the cookie is available to all of webserver.com and its subdomains, such as www.webserver.com and images.webserver.com. If it is images.webserver.com, the cookie is available only to images.webserver.com and its subdomains such as sub.images.webserver.com, but not, say, to www.webserver.com.	.webserver.com
secure	(Optional). Specifies whether or not the cookie should only be transmitted over a secure HTTPS connection. TRUE indicates that the cookie will only be set if a secure connection exists. Default is FALSE	
httponly	(Optional.) If set to TRUE the cookie will be accessible only through the HTTP protocol (the cookie will not be accessible by scripting languages). This setting can help to reduce identity theft through XSS attacks. Default is FALSE	

Setting up a cookie

```php
<?php
    $cookie_name="username";
    $cooke_value="TOM";

    setcookie($cookie_name, $cooke_value, time() + (86400 * 30), "/"); //86400 -> 1day
?>
```

- Run the above code
- Now check for cookies in your browser settings



$_COOKIE

- Reading the value of a cookie is as simple as accessing the $_COOKIE (superglobal) system array.

```php
<?php
    $cookie_name="username";
    $cooke_value="TOM";

    setcookie($cookie_name, $cooke_value, time() + (86400 * 30), "/"); //86400 -> 1day


    if (isset($_COOKIE['username']))
        $username = $_COOKIE['username'];
?>
```

Sessions[2]

- A session is started with the session_start() function.
- Session variables are set with the PHP global variable: $_SESSION.
- To remove all global session variables and destroy the session, use session_unset() and session_destroy():

Sessions

- Because your program can't tell what variables were set in other programs or even what values the same program set the previous time it ran.
- You'll sometimes want to track what your users are doing from one web page to another.
- PHP provides a much more powerful and simpler solution in the form of sessions.
- Session variables solve this problem by storing user information to be used across multiple pages (e.g. username, favorite color, etc). By default, session variables last until the user closes the browser.

Example

index.php

```php
<?php
    session_start();
    $_SESSION["ID"]="IT/2021/F/001";
    $_SESSION["email"]="exampleuser.gmail.com";
?>
```

example.php

```php
<?php
    session_start();
?>
<html>
    <body>
        <?php
        // Echo session variables that were set on previous page
        echo "Favorite color is " . $_SESSION["ID"] . ".<br>";
        echo "Favorite animal is " . $_SESSION["email"] . ".";
        ?>
    </body>
</html>
```

Activity 7.7

- Use the login form created in Activity 7.6.
- Set the login credentials(Username) into a session variable.
- Check the session value set in the above in another page.

Thank You !

Questions ?

HNDIT3022
Web Programming

Week8:- Working with Databases-Part1

Introduction to My SQL

Lesson outline

- Introduction to MY SQL
- Connecting MY SQL databases using PHP
- Insert, Update, and Delete data
- Insert values using HTML form

Introduction

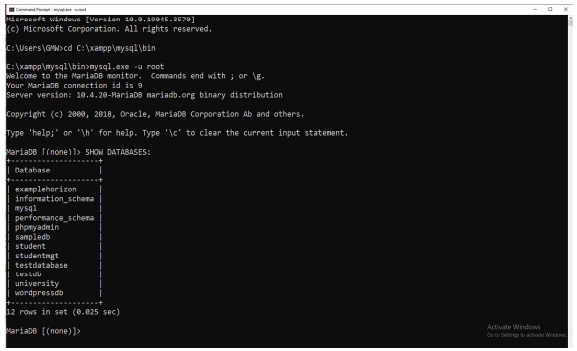
- MySQL is a popular database management system for web servers.
- One reason for its success must be the fact that, like PHP, it's free to use.
- It's also extremely powerful and exceptionally fast.
- MySQL is also highly scalable, which means that it can grow with your website.

My SQL

- A MySQL database contains one or more tables, each of which contains records or rows.
- Within these rows are various columns or fields that contain the data itself.

Starting My SQL

- Example in the XAMPP CLI

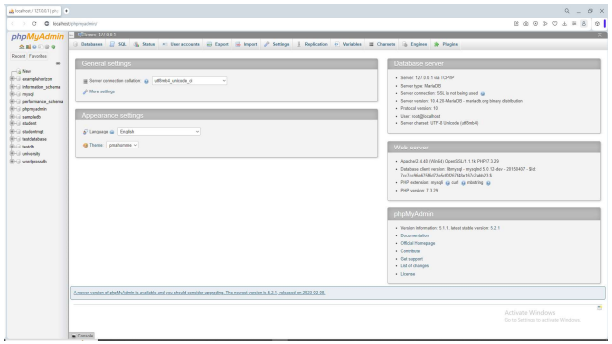


Terminology

- Database
  - The overall container for a collection of MySQL data
- Table
  - A sub-container within a database that stores the actual data
- Row
  - A single record within a table, which may contain several fields
- Column
  - The name of a field within a row

Starting My SQL[2]

- Example in XAMPP GUI(phpmyadmin)



Activity 8.1

- Let's create a new database via phpmyadmin. (See the video)

Connecting to My SQL

- mysql_connect() - Open a connection to a MySQL Server.

- Parameters
  - Server
    - The MySQL server. It can also include a port number
    - Ex:- localhost
  - Username
    - The username is given to your My SQL server
    - Ex:- root
  - Password
    - The password is given to your My SQL server

Accessing My SQL using PHP

Activity 8.2

- Let's establish the connection to My SQL server.

```php
<?php
    $server="localhost";
    $username="root";
    $password="";

    $connection= mysql_connect($server, $username, $password);

    if(!$connection){
        die("Connection failed: " . mysql_error());
    }
    echo "Successfully connected";
?>
```

- However mysql_connect was deprecated in PHP 5.5.0, and it was removed in PHP 7.0.0. Instead, the MySQLi or PDO_MySQL extension should be used.

Let's modify the code

- Use myqli_connect() instead mysql_connect

```php
<?php
    $server="localhost";
    $username="root";
    $password="";

    $connection= mysqli_connect($server, $username, $password);

    if(!$connection){
        die("Connection failed: " . mysqli_connect_error());
    }
    echo "Successfully connected";
?>
```

Selecting a database[2]

- At the same time you establish the connection you can select the database also.

```php
<?php
    $server="localhost";
    $username="root";
    $password="";
    $dbname="university";

    $connection= mysqli_connect($server, $username, $password, $dbname);

    if(!$connection){
        die("Connection failed: " . mysqli_connect_error());
    }
    echo "Successfully connected";
?>
```

Selecting a database

- Using mysqli_select_db() we can select the database we want.

```php
<?php
    $server="localhost";
    $username="root";
    $password="";
    $dbname="university";

    $connection= mysqli_connect($server, $username, $password);

    if(!$connection){
        die("Connection failed: " . mysqli_connect_error());
    }
    echo "Successfully connected";


    mysqli_select_db($connection,$dbname);// pass the connecton and db name as parameters
?>
```

Building and executing a query

- mysqli_query() is the function used to execute a query.
- It will take the connection created above and a SQL query as the parameters.
- When you pass SQL query it should be given as a string.
- Ex:-

```php
$query="INSERT INTO student(studentID,studentName,email)
        VALUES ('KEG/IT/2021/F/002','Nimal','nmal@gmail.com')";

$result=mysqli_query($connection, $query);
```

Inserting data

```
$query="INSERT INTO student(studentID,studentName,email)
        VALUES ('KEG/IT/2021/F/002','Nimal','nmal@gmail.com')";

$result=mysqli_query($connection, $query);

if(!$result){
    die("Values not inserted");
}
echo "Inserted Successfully";
```

Activity 8.3

- Modify the above code to delete a given record.

Updating data

- We can use the same method used in insertion.

```
$updateQuery="UPDATE student SET email='kamla90@gmail.com'
              WHERE studentID='KEG/IT/2021/F/001'";
$result=mysqli_query($connection, $updateQuery);
if(!$result){
    die("Data cannot be updated");
}
echo "Updated successfully";
```

Activity 8.3 -Answer

```
$deletQuery="DELETE FROM student WHERE studentID='KEG/IT/2021/F/001'";
$result=mysqli_query($connection, $deletQuery);
if(!$result){
    die("Data is not deleted");
}
echo "Deleted one record";
```

Questions ?

Thank You !

HNDIT3022
Web Programming

Week9:-Working with Databases – Part2

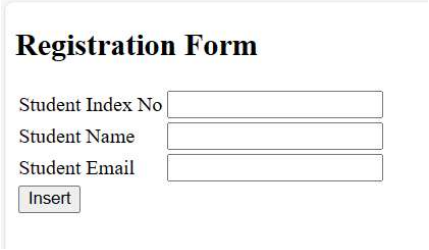
Introduction

- The main way that website users interact with PHP and MySQL is through the use of HTML forms.
- Handling forms is a multipart process.
- First a form is created, into which a user can enter the required details.
- This data is then sent to the web server, where it is interpreted, often with some error checking.
- When the code is satisfied with the accuracy of the input, it takes some action that usually involves the database.

Working with Forms

Activity 9.1

- Create the following HTML form



Activity 9.1- Answer

```
<form action="<?php echo $_SERVER["PHP_SELF"]?>" method="post">
    <tr><td><label>Student Index No</label></td>
        <td><input type="text" name="studentID"></td></tr>

    <tr><td><label>Student Name</label></td>
        <td><input type="text" name="studentName"></td></tr>

    <tr><td><label>Student Email</label></td>
        <td><input type="text" name="email"></td></tr>

    <tr><td colspan=2><input type="submit" value="Insert" name="insert"></td></tr>
</form>
```

```
<?php
    $server="localhost";
    $username="root";
    $password="";
    $dbname="universty";

    $connection= mysqli_connect($server, $username, $password, $dbname);

    if(!$connection){
        die("Connection failed: " . mysqli_connect_error());
    }


    if(!empty($_POST["insert"])){
        $IndexNo=$_POST["studentID"];
        $name=$_POST["studentName"];
        $email=$_POST["email"];

        $query="INSERT INTO student(studentID,studentName,email)
                VALUES ('$IndexNo','$name','$email')";

        $result=mysqli_query($connection, $query);

        if(!$result){
            die("Values not inserted");
        }
        echo "Inserted Successfully";
    }
?>
```

Inserting form values

Step 1:- Establish the connection(discussed in Lesson 8).

Step 2:- Collect the user-entered value using the post(discussed in Lesson 7).

Step 3:- pass the collected values into an SQL INSERT query.

Step 4:- Execute the query in PHP

Fetching Data

Fetching data

- When you want to fetch data from a database you need to run a SELECT query.
- It will return a result set.
- Using mysqli_fetch_row(), mysqli_fetch_assoc() or mysqli_fetch_array() you can create an array of each row of the result by passing that in to a loop.

Fetching data[3]

```php
<?php
    $server="localhost";
    $username="root";
    $password="";
    $dbname="university";

    $connection= mysqli_connect($server, $username, $password, $dbname);

    if(!$connection){
        die("Connection failed: " . mysqli_connect_error());
    }

    $selectQuery="SELECT * FROM student";
    $result=mysqli_query($connection, $selectQuery);

    if(mysqli_num_rows($result)>0){
        while($record=mysqli_fetch_row($result)){
            echo $record[0]. " ". $record[1]. " ". $record[2]. "</br>";
        }
    }

?>
```

Fetching data[2]

- mysqli_fetch_row()
  - Fetch a result row as a numeric array.
- mysqli_fetch_array()
  - Fetch a result row as a numeric array and as an associative array. (Use field names as array indexes for associative array)
- mysqli_fetch_assoc()
  - Fetch a result row as an associative array. (Use field names as array indexes for associative array)
- mysqli_num_rows()
  - Return the number of rows in a result set.

Activity 9.2

- Display the fetched values in an HTML table. Add a delete option for each record.

Activity 9.2- Answer

```php
$selectQuery="SELECT * FROM student";
    $result=mysqli_query($connection, $selectQuery);

    if(mysqli_num_rows($result)>0){
        while($record=mysqli_fetch_row($result)){
            echo "<tr>";
            echo "<td>".$record[0]."</td>";
            echo "<td>".$record[1]."</td>";
            echo "<td>".$record[2]."</td>";
            echo "<td><a href='".$_SERVER["PHP_SELF"]."?ID=$record[0]'>DELETE</a></td>";
            echo "</tr>";
        }
    }

if(!empty($_GET["ID"])){
    $deleteQuery="DELETE FROM student WHERE studentID='".$_GET["ID"]."'";
    $result=mysqli_query($connection, $deleteQuery);
    if(!$result){
        die("Record is not deleted");
    }
    echo "One record deleted";
    header("Location:index.php");
}
```

Questions ?

Activity 9.3

- Try to modify the code segment as it can update values.(Refer the php file given in LMS)

Thank You !

HNDIT3022
Web Programming

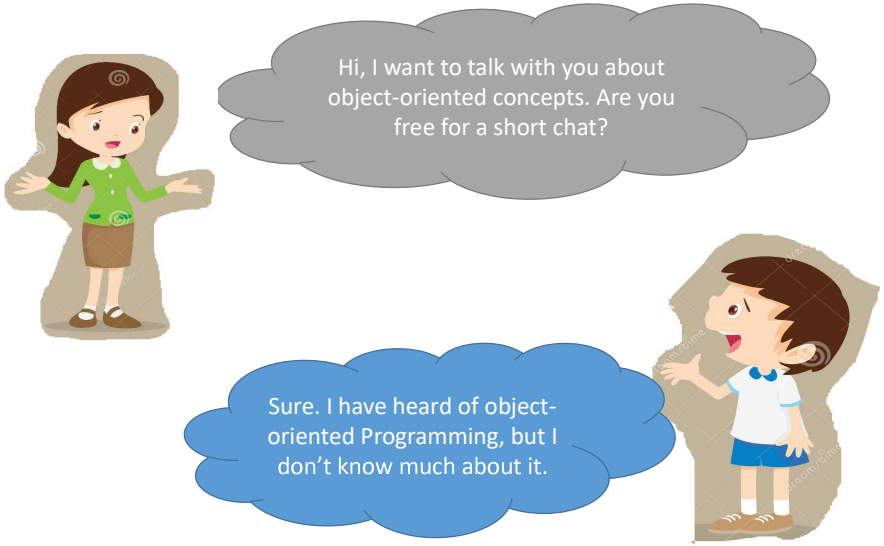
Week10:-Object and Operations in PHP

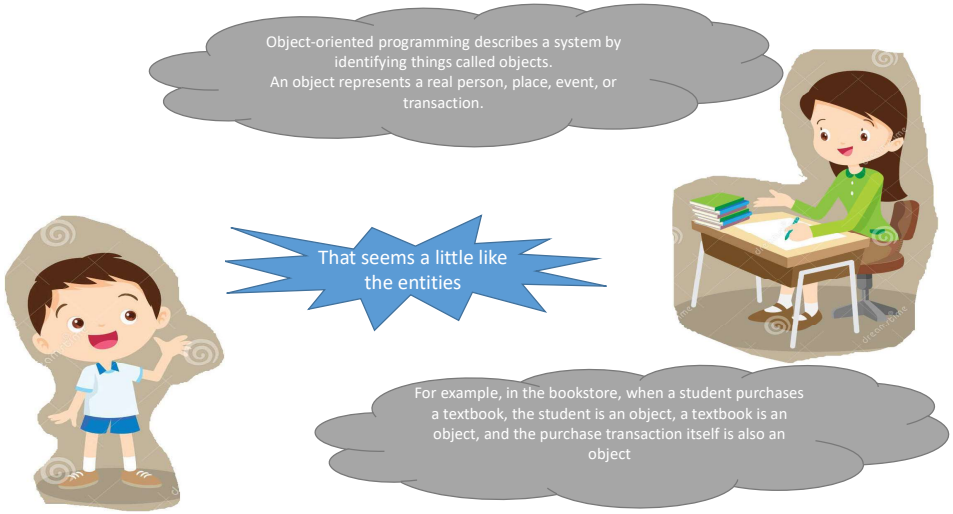
Lesson Outline

- Introduction to OOP
- Creating classes in PHP
- Creating objects in PHP
- Constructor and destructor
- Inheritance

Introduction to OOP

- OOP stands for Object-Oriented Programming.
- Procedural programming is about writing procedures or functions that perform operations on the data, while object-oriented programming is about creating objects that contain both data and functions.

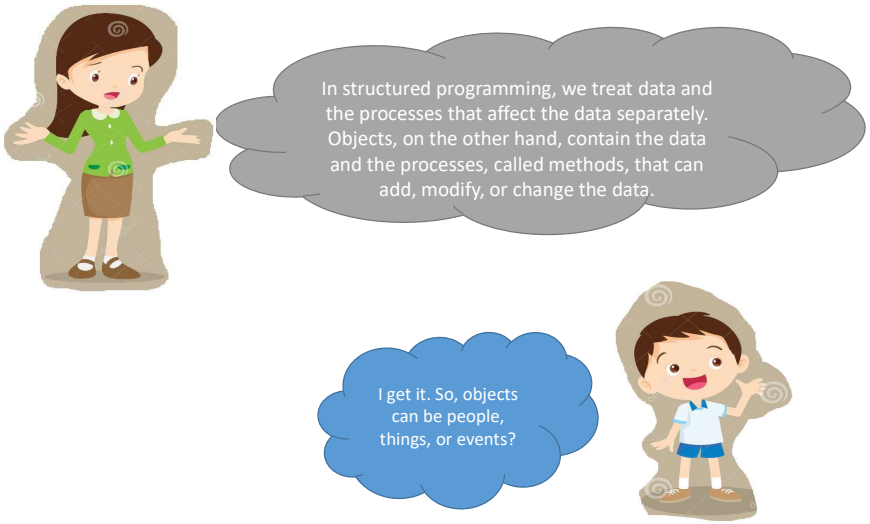




Object


- Attributes
  - It's the characteristics or the properties of the Object
- Methods
  - Methods are the things that the Object can do



Class

- A class represents a collection of objects having same characteristic properties that exhibit common behavior.
- It gives the blueprint or description of the objects that can be created from it.
- Creation of an object as a member of a class is called instantiation.
- Thus, object is an instance of a class.

Activity 10.1

- Discuss the four concepts in Object object-oriented programming briefly.
  - Encapsulation
  - Abstraction
  - Inheritance
  - Polymorphism

Declaring properties and methods

- Let's add one property
- Two functions
  - Setter and a getter
- The $this keyword refers to the current object, and is only available inside methods.

```php
<?php

    class student{
        public $studentID;

        function setID($studentID){
            $this->studentID=$studentID;
        }

        function getID(){
            return $this->studentID;
        }
    }
?>
```

Define a class

- A class is defined by using the class keyword, followed by the name of the class.

```php
<?php

    class student{

    }

?>
```

Defining objects

- Creating the object using new keyword

```php
$st1=new student();  //create an object
```

- Invoking functions.

```php
$st1->setID("KEG/IT/2021/F/001");//Invoking the setID()
echo $st1->getID();// Invoking getID()
```

Activity 10.2

- Create a class named a circle.
  - Add one constant to store the Pi value.
  - Add one variable to hold the radius.
  - Create a function to return the area of the circle.
- Create an object
  - Assign suitable radius
  - Print the area

Constructor

- A constructor allows you to initialize an object's properties upon creation of the object.
- __construct() is the function used in constructors.

```
function __construct(){

}
```

Activity 10.2-Answer

```
<?php

    class circle{
        const PI=3.14; //class constant can be create using const keyword
        public $radius;

        function calArea(){
            $area=self::PI*$this->radius*$this->radius;
            return $area;
        }
    }

    $c1=new circle();
    $c1->radius=10;
    echo "Area is ". $c1->calArea();
?>
```

Activity 10.3

- Let’s modify the code in Activity 10.2.
- The radius should be given at the same time when you create the circle.

Activity 10.3-Answer

```php
<?php

    class circle{
        const PI=3.14;
        public $radius;

        function __construct($radius){
            $this->radius=$radius;
        }

        function calArea(){
            $area=self::PI*$this->radius*$this->radius;
            return $area;
        }
    }

    $c1=new circle(10);
    echo "Area is ". $c1->calArea();
?>
```

Example

```php
<?php

    class circle{
        const PI=3.14;
        public $radius;

        function __construct($radius){
            $this->radius=$radius;
        }

        function __destruct(){
            echo "</br>End of the execution";
        }

        function calArea(){
            $area=self::PI*$this->radius*$this->radius;
            return $area;
        }
    }

    $c1=new circle(10);
    echo "Area is ". $c1->calArea();
?>
```

Output

Area is 314
End of the execution

Destructors

- A destructor is called when the object is destructed or the script is stopped or exited.
- __destruct() function that is automatically called at the end of the script.
- Constructors and destructors help reduce the amount of code.

Access Modifiers

- There are three access modifiers:
  - public - the property or method can be accessed from everywhere. This is default
  - protected - the property or method can be accessed within the class and by classes derived from that class
  - private - the property or method can ONLY be accessed within the class

Inheritance

- Deriving classes using an existing class.
- The child class will inherit all the public and protected properties and methods from the parent class.
- In addition, it can have its own properties and methods.
- An inherited class is defined by using the extends keyword.

Example

```php
<?php

    class vehicle{
        protected $brand;

        function honk(){
            echo "Tuuut, Tuut!";
        }
    }

    class car extends vehicle{
        private $modelname;

        function setModel($modelname){
            $this->modelname=$modelname;
        }
    }

    $cl=new car();
    $cl->setmodel("Mustang");
    $cl >honk();

?>
```

Questions ?

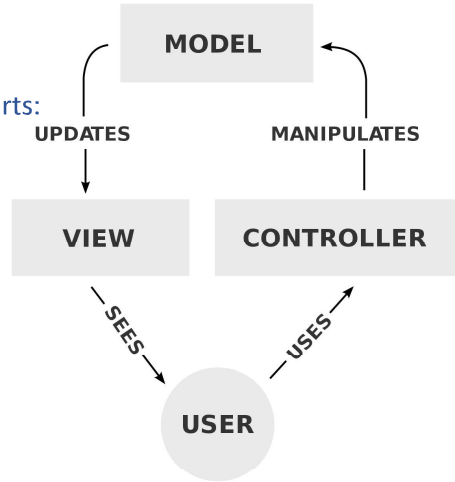
Thank You !

HNDIT3022
Web Programming

Week11:- Use MVC frameworks for Web Applications

MVC

- MVC is a collation of its three core parts:
  - Model
  - View
  - Controller



MVC Architecture

Model

- The Model is the name given to the permanent storage of the data used in the overall design.
- It must allow access for the data to be viewed, or collected and written to, and is the bridge between the View component and the Controller component in the overall pattern.
- The model has no connection or knowledge of what happens to the data when it is passed to the View or Controller components.

Model[2]

- Its sole purpose is to process data into its permanent storage or seek and prepare data to be passed along to the other parts
- The Model, however, cannot simply be summed up as a database, or a gateway to another system that handles the data process.
- The Model must act as a gatekeeper to the data itself, asking no questions but accepting all requests that come its way.

View[2]

- For example, many mistake the View as having no connection whatsoever to the Model and that all of the data displayed by the View is passed from the Controller. In reality, this flow disregards the theory behind the MVC pattern completely.
- It's also important to remember that the View part is never given data by the Controller.
- There is no direct relationship between the View and the Controller without the Model in between them

View

- The View is where data, requested from the Model, is viewed and its final output is determined.
- Traditionally in web apps built using MVC, the View is the part of the system where the HTML is generated and displayed.
- The basic example of this is a button generated by a View, which a user clicks and triggers an action in the Controller.

Controller

- Its job is to handle data that the user inputs or submits and update the Model accordingly.
- It is the only part of the pattern the user should be interacting with.
- The Controller can be summed up simply as a collector of information, which then passes it on to the Model to be organized for storage, and does not contain any logic other than that needed to collect the input.

Controller[2]

- The Controller is also only connected to a single View and to a single Model, making it a one way data flow system, with handshakes and sign-offs at each point of data exchange.

- It's important to remember the Controller is only given tasks to perform when the user interacts with the View first, and that each Controller function is a trigger, set off by the user's interaction with the View

MVC Frameworks in PHP

Framework

- A framework is a prepackaged set of software libraries that provide generic functionality and can be customized by user code
- A framework is like a structure that provides a base for the application development process.
- With the help of a framework, you can avoid writing everything from scratch.
- Frameworks provide a set of tools and elements that help in the speedy development process.
- It acts like a template that can be used and even modified to meet the project requirements.

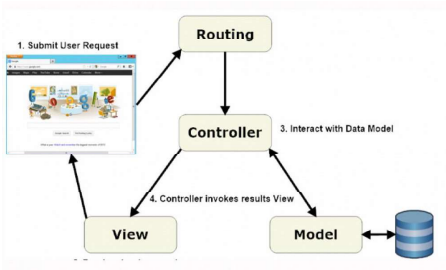
Popular PHP frameworks

- Laravel: Known for its expressive syntax and provides features like routing, authentication, and database management.
- CodeIgniter: One of the most popular PHP frameworks with MVC support, great for creating lightweight web applications.
- Symfony: A modular PHP framework with a built-in debugging system and extensive documentation.
- Zend: A robust PHP framework that is highly scalable and secure.
- CakePHP: The first MVC-based PHP framework with the most thorough set of libraries.

Laravel

Laravel[2]

- Additionally, routes are used to map URLs to designated controller actions, as shown below.
- A request is made say, when a user enters a URL associated with your application.
  - A route associated with that URL maps the URL to a controller action.
  - That controller action leverages the necessary model(s) to retrieve information from the database and then passes that data off to a view.
  - And that view renders the final page



Laravel

- Larval is a web application framework with expressive, elegant syntax.
- It already laid the foundation freeing you to create without sweating the small things in MVC architecture.
- Laravel attempts to take the pain out of development by easing common tasks used in most web projects.
- Laravel applications follow the Model-View-Controller design pattern, where it uses:
  - **Controllers**: handle user requests and retrieve data, by leveraging Models
  - **Models**: interact with the database and retrieve your objects' information
  - **Views**: render pages

Setting up your first web site

Laravel

Initial settings

- Download and Install composer using the given link below
Composer (getcomposer.org)

- Open cmd and direct to the composer folder

- Run the following code in cmd.

```
C:\composer> composer global require laravel/installer
Changed current directory to C:/Users/GMW/AppData/Roaming/Composer
```

- Set the path.
  - Go to C:\Users\username\AppData\Roaming\Composer\vendor\bin
  - Copy the path
  - Go to control panel->system->advanced system settings->environment variables->path edit path add new
  - Paste the copied path there.

Run your application

- Direct to the application via cmd and run php artisan serve

```
C:\xampp\htdocs\WebApplication>php artisan serve
Starting Laravel development server: http://127.0.0.1:8000
```

- Once you run artisan serve you will be get a message as above. Now your web is running on http://127.0.0.1:8000 .(localhost:8000)

- Open your web browser and run localhost:8000 you will get the default welcome page in Laravel.

Creating a new project

- Open cmd and direct to the place where you want to save project.

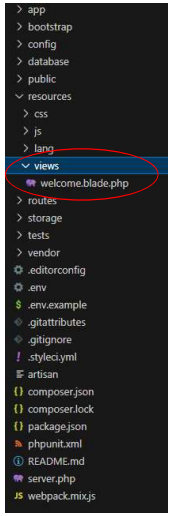
```
C:\Users\GMW>cd    Folder name
```

- Create new project using Laravel new
  - Give a suitable name for your application

```
C:\  Folder path  >laravel new WebApp

  _                             _
 | |                           | |
 | |   _  _  _  _  _  __   ___| |
 | |  / _ | | | | | |/ _ \ / _ \ |
 | |_| (_| | | (_| | | (_) | (_) |
 |____\__,_|_| \__,_|_|\__/ \___|_|

  Creating a "laravel/laravel" project at "./WebApp"
```

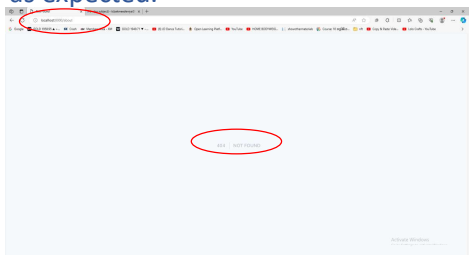
Modifying view

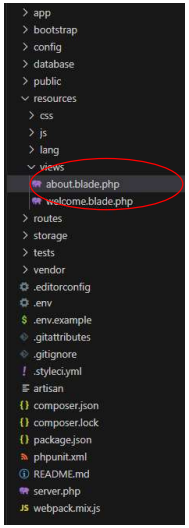
- The web page you see on the localhost:8000 is rendered from the welcome page in the views.
  - Resources->views
- Open welcome.blade.php.
  - Blade is the simple, yet powerful templating engine that is included with Laravel.
  - Unlike some PHP templating engines, Blade does not restrict you from using plain PHP code in your templates.
- Modify the content and refresh the browser.



Modifying view[2]

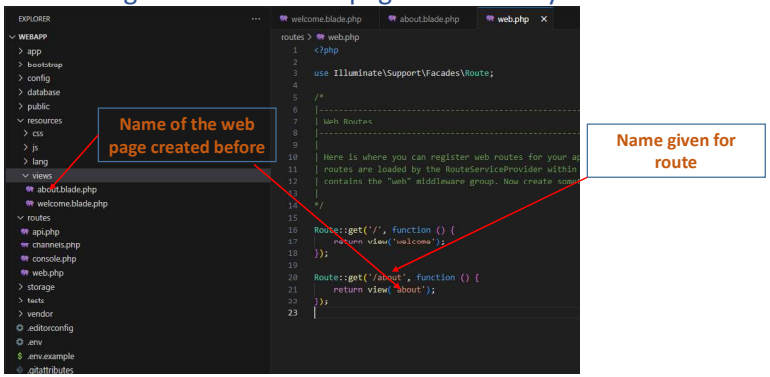
- Let's add a new page to our view. (about.blade.php)
- Add suitable content there.
- Now try to access via the web browser. You will not be able to get the view as expected.





Routes[2]

- Copy the route given for welcome page. And modify as follows.

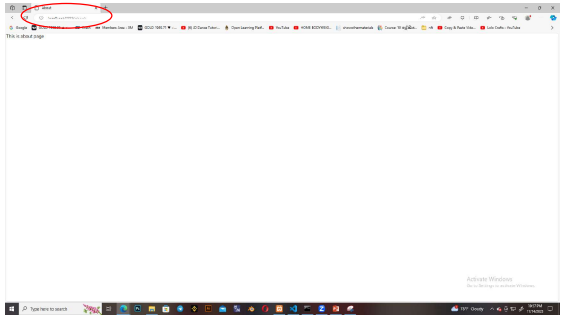


Routes in Laravel

- Routes are actually the web URLs that you can visit in your web application.
  - For example */home, /profile, /dashboard*, etc are all different routes that one can create in a Laravel Application.
- Keep in mind that, routes are case sensitive thus */profile* is different than */Profile*.
- if you are adding a new page (navigation)you are supposed to make a relevant route.
- In Laravel, all of our routes are going to be written in *routes/web.php* file
- This directory is made standard for all our web-related routes. Open this file and let's create our first route with Laravel, write to the end of this file.

Routes[3]

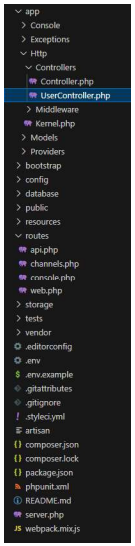
- Once you enter the given route name via the browser, you'll be able to get the view given.



Controllers

- Controllers are there in the app/Http/controllers
- php artisan make: controller ControllerName command will create a controller with the given name.

```
php artisan make:controller UserController
Controller created successfully.
```

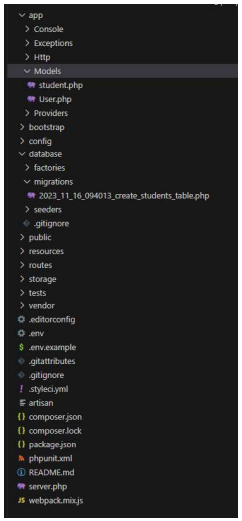


Migrations

- Migrate includes the database tables files. Once executed it will create the tables in there.
- Remove the unnecessary migrations.
- Let us create a new migration.
- Run the following code in the terminal.

```
> php artisan make:model student -m
```

- It will create a migration named as a student.
- And also a model named a student in app/Models



Controllers[2]

- Now lets modify the controller and the routes.

app > Http > Controllers > UserController.php

```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class UserController extends Controller
{
    function welcome(){
        return view("welcome");
    }
}
```

routes > web.php

```php
<?php

use Illuminate\Support\Facades\Route;
use App\Http\Controllers\UserController;



/*
|--------------------------------------------------------------------------
| Web Routes
|--------------------------------------------------------------------------
|
| Here is where you can register web routes for your application. These
| routes are loaded by the RouteServiceProvider within a group which
| contains the "web" middleware group. Now create something great!
|
*/

Route::get('/', [UserController::class, "welcome"]);
```

Path of the controller

Name of the function

- Modify the migration file. Include the fields you want to add

database > migrations > 2023_11_16_094013_create_students_table.php

```php
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreateStudentsTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('students', function (Blueprint $table) {
            $table->id();
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('students');
    }
}
```

database > migrations > 2023_11_16_094013_create_students_table.php

```php
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreateStudentsTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('students', function (Blueprint $table) {
            $table->id();
            $table->string("studentName");
            $table->integer("age");
            $table->string("email");
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('students');
    }
}
```

Migrations

- Create a database in your MySQL server.
  - Ex:- Let us name the database as university.
- Open .env file
- Set the database name, username and password.
- Now run the following command to create the table in your MySQL server.

```
php artisan migrate
```

```env
APP_NAME=Laravel
APP_ENV=local
APP_KEY=base64:gHfNUk9Qn0yLGOX/Ny84kxUoZmuCmzsZXK0CxnV2JMo=
APP_DEBUG=true
APP_URL=http://localhost

LOG_CHANNEL=stack
LOG_DEPRECATIONS_CHANNEL=null
LOG_LEVEL=debug

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=university
DB_USERNAME=root
DB_PASSWORD=
```

Display Data[2]

- Add following code in the welcome page. To display the details.

```blade
@foreach($student as $st)
    <p>{{$st->studentName}} {{$st->age}} {{$st->email}}</p>
@endforeach
```

Display data

- Let us create a new controller named StudentController. Change the routes also

app > Http > Controllers > StudentController.php

```php
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\student;

class StudentController extends Controller
{
    public function index(){
        $student=student::all();

        return view("welcome", ["student"=>$student]);
    }
}
```

routes > web.php

```php
<?php

use Illuminate\Support\Facades\Route;

use App\Http\Controllers\StudentController;


/*
|--------------------------------------------------------------------------
| Web Routes
|--------------------------------------------------------------------------
|
| Here is where you can register web routes for your application. These
| routes are loaded by the RouteServiceProvider within a group which
| contains the "web" middleware group. Now create something great!
|
*/



Route::get('/', [StudentController::class, "index"]);
```

Questions ?

Thank You !