B.C.A 5th Sem Exam paper solution [2024]

Page No. 17
Date: / /

Y02: Building Application using PHP

Marks 70.

a-100 Answer the Followings. [06 marks]

1] Define: $_FILES

=> $_FILES is a superglobal variable in PHP that used to collect and handle file upload data submitted through a HTML From with the enctype = "multipart /form-data" attribute.

2] Write any two predefined variable handling functions with syntax and example

=> the two predfined variable handling functions.

1] isset() function

Syntax

bool isset (mixed $ var, mixed $.....)'

2] empty() function

. Syntax

bool empty(mixed $ var)'

// using isset() and empty() together

if (isset($var3) && empty ($var3)){

echo '$var3 is set but empty.
';

}else{

echo '$var3 is either not set or not empty.
';

}

output

$var3 is either not set or not empty.

3] Explain the foreach loop with syntax and example.

=> The 'foreach' loop is PHP is used to iterate over array or objects. It provides a simple and easy to-use syntax for iterating through the elements of an array or the properties of an object.

Syntax

```
foreach ($array as $value) {
  //code to be executed for each iteration
}
```

```
<?php
$fruits = ['Apple', 'Banana', 'orange'];
foreach ($fruits as $fruit) {
    echo $fruit . '<br>';
}
?>
```

Q-1(b) Answer the following (Any two) [12 marks]

1] Differentiate: GET vs POST. Explain $_POST.

GET	POST
1] used for sending small amounts of data	• used for sending large amounts of data.
2] Parameters are visible in the URL.	• Parameters are not visible in the URL (sent in the body of the HTTP request).
3] limited amount of data can be sent (URL length limitation).	• No limitation on the amount of data that can be sent
4] Less secure for sensitive Data, as parameters are visible in the URL.	• More secure for sensitive data as parameters are not visible in the URL.
5] Caching is possible as parameters are part of the URL	• No caching by default
6] Generally used for retrieving data from the server.	• Generally used for submitting data to the server, such as form submissions.

* Explanation of $_POST.

=> $_POST is a PHP superglobal variable that is used to collect from data that is submitted using

the post method. When a form is submitted using the post method the form data is sent to the server in the body of the HTTP request rather than being appended to the URL. like with the GET method.

=> The '$POST' variable is an associative array where the keys are the names of the form fields and the values are the submitted data For example if a form field has the name "username", the submitted value can be accessed using `$_POST['USERNAME'];

Example

Html

```
<form method="post" action="process.php">
    <input type="text" name="username">
    <button type="submit"> submit </button>
</form>
```

Php

```
<?php
if(isset($_POST['username'])){
    $username = $_POST['username'];
    echo "Hello, $username!";
}else{
    echo "No username submitted.";
}
?>
```

2) Define: Array. Explain any five Array functions with syntax and example.

=> An array in PHP is a data structure that can store multiple values under a single variable name. It is a collection of elements, where each element can be accessed by its index or key.

1) count():

Syntax

int count(array $array [, int $mode = COUNT_NORMAL])

=> Returns the number of elements in an array.

Exp

$arr = [1, 2, 3, 4, 5];
echo count($arr); // output 5

2) Push():

Syntax

int array_Push(array & $array, mixed $value [, mixed $ ...]);

=> Pushes one or more elements onto the end of the array.

Exp

$arr = [1, 2, 3];
array_Push($arr, 4, 5);
print_r($arr);

output

Array ([0] => 1 [1] => 2 [2] => 3 [3] => 4 [4] => 5)

3] Pop():
Syntax
'mixed array_pop(array &$array)'
=> Removes and return the last element of an array.
Exp
$arr = [1,2,3];
$last = array_pop($arr);
echo $last; // output: 3

4] Shift():
Syntax
'mixed array_shift(array &$array)'
=> Removes and returns the first element of an array.
exp
$arr = [1,2,3];
$first = array_shift($arr);
echo $first; // output: 1

5] Merge():
Syntax
'array array_merge(array ... &$arrays)'
=> combines two or more arrays into a single array.
exp
$arr1 = [1,2];
$arr2 = [3,4];
$merged = array_merge($arr1, $arr2);
print_r($merged); // output: array([0] => 1[1] => 2[2] => 3[3] => 4)

OR

3] Differentiate: Client side scripting languages and server side scripting languages.

⇒ Client side scripting Language and server side scripting Language are two types of scripting languages used in web development.

Client-side scripting Language	Server side scripting Language
1] Execution: Executed on the client web browser	• Execution: ~~Responsible for generating dynamic content~~ Executed on the server before sending the response to the client's web browser.
2] Responsibility: Responsible for interactions and behaviors on the client-side such as validating form inputs dynamically updating ~~form inputs~~ content and handling user events like mouse clicks and keyboard inputs	• Responsible for generating dynamic content, processing form submissions, accessing databases and performing other server-side tasks.
3] Examples: Javascript HTML, CSS	• PHP, Python (with frameworks like Django) Ruby, Node.js
4] Advantages ⇒ Reduces server load by processing tasks on clients machine	• Advantages Accesses and manipulates server resources, database and files.

=> provides immediate feedback to us on without requiring server interaction	• generates personalized content based on user input or database queries.
=> Enhances user experience by enabling dynamic, and interactive web pages.	Enhances security by controlling access to sensitive data and functionality

=> PHP is a server-side scripting language meaning it is executed on the web server before the HTML is sent to the client's browser.

=> PHP scripts generate dynamic content, interact with database, handle form submissions and perform other server side tasks.

=> Client-side scripting languages like Javascript can be embedded within php scripts to enhance interactivity on the client-side.

Q-2 [A] Attempt the following [05 marks]

1] Explain defining a class and object in php with example.

class :- In php you can define a class using the class keyword followed by the class name. Inside the class you can define properties and methods.

Object :- Once you have defined a class you create objects (instances) of that class using the 'new' keyword followed by the class name.

~~$my car -> brand = 'Toyota';~~

```
class car {
  // properties
  public $brand;
  public function start() {
    return "The $this -> brand starts.";
  }
  public function drive() {
    return "The $this -> brand is driving.";
  }
}

$my car -> brand = 'Toyota';
echo $my car -> start();
echo $my car -> drive();
```

21 How to read a file? Explain it with file handling functions.

⇒ Using file_get_contents():
file_get_contents() reads the entire contents of a file into a string
```
$content = file_get_contents('file.txt');
  echo $content;
```
Using fopen(), fread(), fclose():
- fopen(): opens a file or URL
- fread(): reads from an open file.
- fclose(): closes an open file.

```
$handle = fopen('file.txt', 'r');
if(!$handle){
   while(($line = fread($handle, 4096))
     echo $line;
   }
   fclose($handle);
}
```
fgets() reads a line from an open file:
```
$handle = fopen('file.txt', 'r');
if ($handle){
   while (($line = fgets($handle)) !== false){
     echo $line;
   }
fclose($handle);
}
```

Q-2 [B] Answer the following (Any two) [12 marks]

1] Explain error handling try-catch-block with example.

=> error handling using 'try', 'catch', and 'finally' blocks allows you to gracefully handle exceptions that may occur during the execution of your code.

=> the try block contains the code that you want to monitor for exceptions.

=> if an exception occurs with in the 'try' block, it is caught by the corresponding 'catch' block.

=> the 'catch' block is used to handle the exception by providing appropriate error handling code.

Syntax

```
try {
} catch (Exception $e) {
    // code to handle the exception
}
```

Example

```
try {
    // Division by zero will throw an exception
    $result = 10/0;
    echo "Result: $result";
} catch (Exception $e) {
    // Handle the exception
    echo "An error occurred: " . $e->getMessage();
}
```

```
finally {
// This block is optional and will always be executed
regardless of whether an exception occurred or not
    echo "End of try-catch block.";
}
```

In this example:

- The 'try' block attempts to divide 10 by 0 which will result in a division by zero exception.
- The exception is caught by the 'catch' block, which then echoes the error message.
- The 'finally' block, which is optional, is executed regardless of whether an exception occurred or not. It is typically used for cleanup tasks.
- Output (when exception occurs)

⇒ An error occured: Division by zero End of try catch block.

- Output (when exception does not occurs)

⇒ End of try-catch block.

Using the 'try', 'catch', and 'finally' blocks you can effectively handle errors and exceptions in your php code, ensuring that your application remains stable and resilient even in the face of unexpected issues.

2) Explain $_cookie and $_session with example.

=> Both '$_COOKIE' and '$_SESSION' are superglobal arrays in php used to manage data across multiple pages or request s. However they differ in how they store and manage data.

* $_COOKIE:

- '$_COOKIE is used to store data on the client side as cookies

- cookies are stored in the user's browser and sent with every request to the server.

- They have an expiration time and can be set to persist for a specific duration

- cookies are not secure for sensitive data as they can be manipulated by the user.

Example of $_COOKIE

```
// Setting a cookie
Setcookie("username", "John", time() + 3600, "/");
// Cookie expires in 1 hour

// Retrieving cookie value
echo $_COOKIE["username"]; output: John
```

☆ $_SESSION

- $_SESSION is used to store data on the server side
- Session data is stored on the server and identified by a unique session ID, which is typically stored in a cookie on the client side.
- Sessions are more secure than cookies as the data is stored on the server and cannot be directly manipulated by the user.
- Session data persists until the session is destroyed or expires.

Example of $_SESSION

```
// starting a session
session_start();
// storing data in session
$_SESSION["username"] = "John";
// Retrieving session data
echo $_SESSION["username"];
output : John
```

$_SESSION variables you need to start the session using `session_start();`

OR

3) Explain the php oop inheritance with example.

=> object-oriented programing (oop) inheritance allows a class to inherit properties and methods from another class called the parent or base class. This concept promotes code reusability and enables the creation of hierarchical relationship between classes.

Example program.

```
<?php
class vehicle {
  // Properties
  protected $brand;
  protected $color;
  // constructor
  public function __construct($brand, $color) {
      $this->brand = $brand;
      $this->color = $color;
  }
  // Methods
  public function start() {
      return "starting the " . $this->color . " " . $this->
          brand . "!";
  }
  public function stop() {
      return "stopping the " . $this->color . " " . $this->brand . "!";
  }
}
// Derived class 1: car
class car extends vehicle {
  private $num_doors;
```

```
// constructor
public function __construct($brand, $color,
        $numDoors) {
    parent::__construct($brand, $color);
    $this->numDoors = $numDoors;
}
// Additional method
public function drive() {
    return "Driving the $this->color $this->brand
        car with $this->numDoors doors...";
}
}
// Derived class 2: Motorcycle
class Motorcycle extends Vehicle {
    private $engineSize;
    // constructor
    public function __construct($brand, $color,
        $engineSize) {
        parent::__construct($brand, $color);
        $this->engineSize = $engineSize;
    }
    // Additional method
    public function ride() {
        return "Riding the $this->color $this->brand
            Motorcycle with $this->engineSize cc
            engine...";
    }
}
$myCar = new Car("Toyota", "Blue", 4);
$myMotorcycle = new Motorcycle("Honda", "Red", 250);
```

```
echo $mycar -> start() . "<br>";
echo $mycar -> drive() . "<br>";
echo $mycar -> stop() . "<br>";

echo $myMotorcycle -> start() . "<br>";
echo $myMotorcycle -> ride() . "<br>";
echo $myMotorcycle -> stop() . "<br>";

?>
```

Q-3[A] Attempt the following [06 marks]

1] Define: Web service. Write full form of SOAP, UDDI, WSDL.

=> A web service is a software system designed to support interoperable machine-to-machine interaction over a network it provides a standardized way for different applications to communicate with each other over the internet, regardless of the platform or programing language used.

- SOAP: Simple object Access protocol
- UDDI: Universal Description, Discovery and integration
- WSDL: Web services Description Language

2] Explain sending an email in php.

=> Sending an E-mail in php involves using the mail() function or a library like PhpMailer. using the mail() function

```
$to = "recipient@eaple.com";
$subject = "Test Email";
$message = "This is a test email.";
$headers = "From: sender@example.com";
if(mail($to, $subject, $message, $headers)){
    echo "Email sent successfully.";
}else{
    echo "failed to send email.";
}
```

3] Which function counts return records in php from mysql? Explain it with syntax and example.

[To count the] the function used to count the number of records returned from a MySQL query in php is 'mysqli_num_rows()'.

Syntax
int mysqli_num_rows (mysqli_result $result)
Example
```
$mysqli = new mysqli("localhost", "username",
          "password", "database");
$query = "SELECT * From table_name";
$result = $mysqli->query($query);
$num_rows = mysqli_num_rows($result);
echo "Number of records returned:" $num_rows;
```

Q-3 [B] Answer the following (Any two) [12 marks]

[1] Explain web service Model.

⇒ A web service model in php typically follows the client-server architecture where a server provides services of resources to clients over the web. There are different approaches to implementing web services in php including SOAP (Simple object Access Protocol) REST (Representational state transfer) and Json-RPC (Json Remote procedure call).

(1) SOAP (Simple object Access protocol):

- SOAP is a protocol for exchanging structured information in the implementation of web services.
- It uses XML for message formatting and relies on other protocols like HTTP, SMTP or RTP for message transmission
- SOAP services are described using WSDL (web services Description language)

(2) REST (Representational state transfer):

- REST is an architectural style for designing networked applications.
- It relies on stateless communication and uses standard HTTP methods (GET, POST, PUT, DELETE) for interaction.

- In php RESTfull web services can be implemented using frameworks like slim, lumen, or directly using PHP's built-in features.

(3) JSON-RPC (JSON Remote procedure call).

- JSON-RPC is a lightweight remote procedure call protocol using JSON for data encoding.
- It allows clients to invoke methods on a server using a simple JSON-based request response mechanism.
- PHP provides libraries like jsonRPC client and jsonRPC server for implementing JSON-RPC services.

exp

```
require 'vendor/autoload.php';
$app = new Slim\App();
$app->get('/user/{id}', function ($request,
    $response, $args) {
    $userId = $args['id'];
    $userData = getUserData($userId);
return $response->withJson($userData);
});
function getUserData($userId) {
  // code to fetch user data from
    database or other source */
return array('id' => $userId, 'name'
=> 'John Doe', 'email' => 'john@example.
com');
}
```

$app->run();

21 Explain searching and retrieving records from database? Explain with example.

=> Searching and retrieving records from a database in php typically involves executing a SELECT query and processing the result returned by the query.

Example
Let's assume we have a mysql database named "example" with a table named "user" containing columns 'id', 'name', and 'email'. We want to search for users whose names contain a specific keyword and retrieve their information.

Example program

```
<?php
$mysqli = new mysqli("localhost", "username", "password", "example");

if ($mysqli->connect_error) {
    die("Connection failed: " . $mysqli->connect_error);
}
$searchkeyword = "John";
$query = "SELECT * FROM user WHERE name LIKE '%" . $mysqli->real_escape_string($searchkeyword) . "%'";

$result = $mysqli->query($query);
```

```
if($result){
  if($result->num_rows > 0){
    while($row=$result->fetch_assoc()){
    echo "ID:".$row["id"]."|Name:".$row
      ["name"]."|Email:".$row["email"].
      "<br>";
  }
}else{
  echo "No records found matching the search
    criteria.";
}
}else{
  echo "No records found matching the
search criteria.";
}
}else{
  echo "Error executing query:".$mysqli->error;
}
$mysqli->close();
```

1) Establish connection: connect to the MySQL database using 'Mysqli' object-oriented approach
2) prepare SQL Query: construct a SELECT query to search for records matching the given keyword using the 'LIKE' operator.
3) Execute query:- USE '$mysqli->query()' to execute the query and store the result set in the '$result' variable.
4) close connection: close the database connection to free up resources

OR

3] Explain PHP Mysql_query() and Mysql_fetch_array() with example.

1] 'Mysql_query()' Function:

- This function was used to execute a SQL query on a MySQL database.
- It took a SQL query as a parameter and returned a resource or FALSE on failure.

Example

```
<?php
$link = Mysql_connect("localhost", "username",
        "password");
Mysql_select_db("example", $link);

$query = "SELECT id, name, email FROM users";
$result = mysql_query($query, $link);
if(!$result){
  die("Error in SQL query:".mysql_
      error());
```

2] 'Mysql_fetch_array()' Function

- This function fetched a result row as an associative array, a numeric array, or both from result set.
- It took the result resource returned by 'mysql_query()' as a parameter and returned the next row from the result set as an array.

Example

```
while ($row = mysql_fetch_array ($result,
        MYSQL_ASSOC)) {
  echo "ID:" . $row["id"] . " | Name:" . $row
  ["name"] . " | Email:" . $row["email"] . "<br>";
```

Complete Example

```
<?php
$link = mysql_connect("localhost", "username",
  "password");
mysql_select_db("example", $link);

$query = "SELECT id, name, email FROM users";
$result = mysql_query($query, $link);
if(!$result) {
   die("Error in SQL query:" . mysql_error());
}
while ($row = mysql_fetch_array($result, MYSQL
  _ASSOC)) {
  echo "ID:" . $row["id"] . " | Name:" . $row
  ["name"] . " | Email:" . $row["email"] . "<br>";
}
mysql_free_result($result);
mysql_close($link);
?>
```

Q-4[A] Attempt the following [05 marks]

1] What is Joomla? Explain advantages and Real world Examples of Joomla.

=> the Joomla is a popular open-source content managment system (CMS) that allows users to build websites and powerful online application.

- Advantages
  1] User-Friendly interface
  2] Extensibility
  3] community Support
  4] Security
  5] Multilingual Support.
- Real-world Examples:
  1] Corporate websites
  2] E-commerce websites
  3] community portals
  4] Educational websites
  5] Nonprofit organizations

2] What is template manager? Explain in brief.

=> A template manager is a tool or library used to organize and manage the presentation layer of web applications. It typically involves separating the application's logic from its presentation allowing for easier maintenance scalability

and collaboration among developers and designers.

Features

1) Template inheritance
2) variable substitution
3) conditional logic
4) looping constructs.
5) Partial Templates

engines are smarty and twing.

Q-4[B] Answer the following (any two) [12 mark

1] Explain Joomla Architecture.

⇒ Joomla follows model-view-controller (MVC) architectural pattern, which separates the application logic, data and presentation layers.

1. Model (M):

⇒ The model represents the data and business logic of the Joomla application.

⇒ In Joomla, models are responsible for interacting with the database to retrive update and manipulate data.

⇒ Models are typically located in the 'models' directory of the Joomla extension and contain functions for database operations.

⇒ they encapsulate the data access layer and provide an interface for retrieving and manipulating data.

2] View (V):

⇒ The view represents the presentation layer of the Joomla application.

⇒ Views are responsible for rendering the user interface and displaying data to the user.

⇒ In Joomla, views are implemented as PHP files or templates located in the 'views' directory of the Joomla

=> • Views retrieve data from the model and format it for display using HTML, CSS and Javascript

3) Controller (C):
=> The controller acts as an intermediary between the model and view layers.
=> Controllers handle user input, process request and invoke appropriate actions based on user actions.
=> In Joomla, controller are PHP classes located in the 'controllers' directory of the Joomla extension.

4) Components, Modules, and Plugins:
=> Joomla applications are built using components modules ~~are built using~~ and plugins.
=> components are self-contained units of functionality that provide a specific feature or application within Joomla.
=> Modules are smaller extensions that display content or functionality in specific areas of a Joomla website.
=> Plugins are event-driven extensions that respond to specific event or triggers within the Joomla application.

5) Database Abstraction Layer
=> Joomla includes a Database Abstraction Layer that provides a unified interface for interacting with different database systems (e.g MySQL, PostgreSQL, SQLite).

21 Write steps to create menu in Joomla.

=> Creating a menu in Joomla involves several steps, typically performed through the Joomla administration interface.

(1) Login to Joomla Administration:
=> Open your web browser and navigate to the Joomla administration login page.
=> http://yourdomain.com/administrator).
=> Enter your administrator username and password to login.

(2) Access Menu Manager:
=> Once logged in you'll be directed to the Joomla administration dashboard.
=> Look for the "Menus" option in the top menu or sidebar navigation - click on it to access the menu manager.

(3) Create a New Menu:
=> In the menu manager, click on the "New" button or the "Add New Menu" option to create a new menu.
=> Enter a name for your new menu in the provided field and click "Save & Close" or "Save & New" to create the menu.

(4) Add Menu Items:
=> After creating the menu you'll be redirected to the menu items manager.

=> click on the "New" button or the "Add new menu item" option to add menu items to your newly created menu.
=> Fill in the details for the menu item, including the menu title, type (e.g. article, category, external URL) and any additional setting.
=> Click "Save & Close" or "Save & New" to save the menu item and add another one if needed.

(5) Organize Menu Items:
=> once you've added menu items, you can organize them by dragging and dropping them into the desired order.
=> You can also create submenus by dragging menu items slightly to the right, creating a hierarchy.

(6) Assign Menu to Module Position (optional):
=> If you want to display the menu on your Joomla website, you'll need to assign it to a module position.
=> Navigate to "Extensions" > "Modules" and find the "Menu" module.

(7) Publish Menu Module
=> Finally make sure the menu module is published and visible on the frontend of your Joomla website

=> You can control the visibility of the menu module by adjusting its module assignment menu assignment and other setting.

OR

3] How to create and add a new article? Write step.

=> To create and add a new article in Joomla you typically perform the following step using Joomla administration interface.

① Login to Joomla Administration
=> Enter your administrator username and password to login.

② Access Article Manager:
=> once logged in, you'll be directed to the Joomla administration dashboard.
=> Look for the "content" option in the top menu or sidebar navigation. Hover over it to reveal the sub-menu options

③ Create a New Article:
=> In the Article Manager click on the "New" button or the "Add New Article" option to create a new article.
=> you'll be directed to the article editing page where you can enter the details for your new article.

④ Enter Article Details:
=> Enter a title for your article in the "Title" field.
=> Enter the content of your article in the main editing area. you can use the WYSIWYG editor to format text, add images, links and

other media.

⇒ You can also specify additional settings for your article such as the category, tags, metadata, publishing options, and access permissions.

(5) Add images and media

⇒ To add images or media to your article, click on the "Image" button or the "Add media" button in the WYSIWYG editor.

⇒ Upload or select the desired image or media file from your Joomla media library or your computer.

(6) Save and publish Article:

⇒ Once you've entered all the details and content for your article, click on the "Save & Close" or "Save & New" button to save the article.

⇒ If you're ready to publish the article immediately you can also click on the Save & Close" or "Save & New" button to save and publish the article in one step.

(7) View published Article:

⇒ After saving the Article you can view it on the frontend of your Joomla website by navigating to the appropriate menu item or URL associated with the article.

⇒ Make sure to check the article layout and appearance to ensure it's displaying correctly.