B.C.A. Semester – 4

BCA-402A
Building Application Using PHP

UNIT - 4

Object oriented Programming with PHP and Error Handling

Session and Cookies

Introduction

Session and cookies are important concepts in web development, and they are commonly used in PHP to store user-specific data. In this PDF content, we will cover the basics of sessions and cookies in PHP, including how to create, set, and retrieve session and cookie data. We will also discuss some best practices for using sessions and cookies in your PHP applications.

Sessions in PHP

- A session is a way to store information about a user across multiple page requests.
- When a user logs in to your website, for example, you can create a session that stores their user ID, username, or other information.
- This information can then be accessed and used throughout the user's session on your website.

Creating a session in PHP is easy. You simply call the session_start() function at the beginning of your PHP script. This function creates a new session or resumes an existing session, depending on whether a session cookie already exists. Here is an example:

```
<?php
session_start();
// Set session variables
$_SESSION["user_id"] = 12345;
$_SESSION["username"] = "john_doe";
?>
```

In this example, we are calling the session_start() function at the beginning of our PHP script, and then setting two session variables using the $_SESSION superglobal array.

To retrieve the session data in another PHP script, you simply call the session_start() function again, and then access the session variables using the $_SESSION superglobal array. Here is an example:

```
<?php
session_start();

// Retrieve session variables
$user_id = $_SESSION["user_id"];
$username = $_SESSION["username"];
?>
```

In this example, we are calling the session_start() function again, and then retrieving the session variables we set in the previous script.

Cookies in PHP

A cookie is a way to store information on a user's computer or device. Cookies are often used to store user preferences, such as language or layout settings. Cookies can also be used to store user-specific data, similar to sessions.

Creating a cookie in PHP is done using the setcookie() function. This function takes three parameters: the name of the cookie, the value of the cookie, and the expiration time of the cookie. Here is an example:

```
<?php
// Set a cookie
setcookie("username", "john_doe", time() + 3600);
?>
```

In this example, we are setting a cookie with the name "username" and the value "john_doe". The expiration time of the cookie is set to one hour from the current time, using the time() function.

To retrieve the cookie data in another PHP script, you simply access the cookie using the **$_COOKIE** superglobal array. Here is an example:

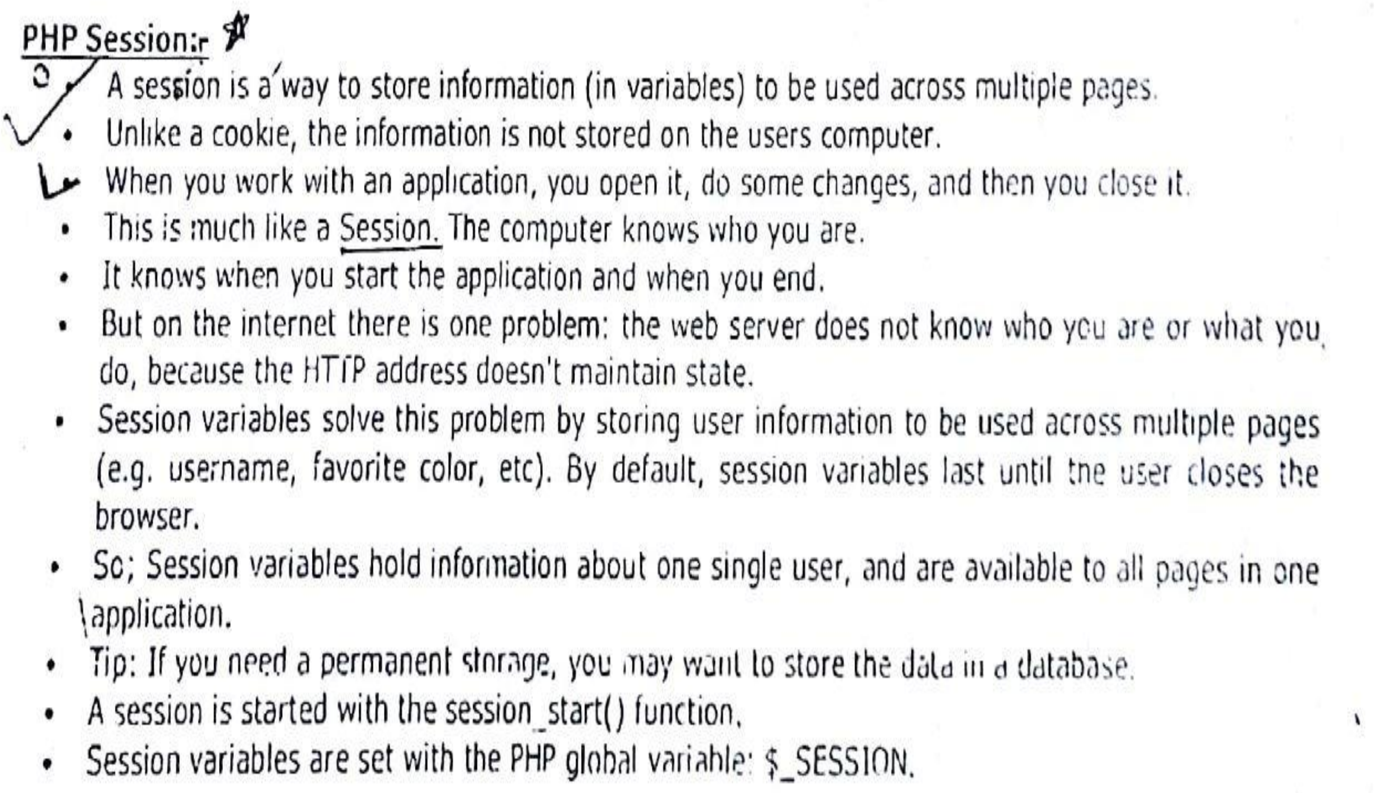
```
<?php
// Retrieve a cookie
$username = $_COOKIE["username"];
?>
```

In this example, we are retrieving the value of the "username" cookie we set in the previous script.

Best Practices for Using Sessions and Cookies

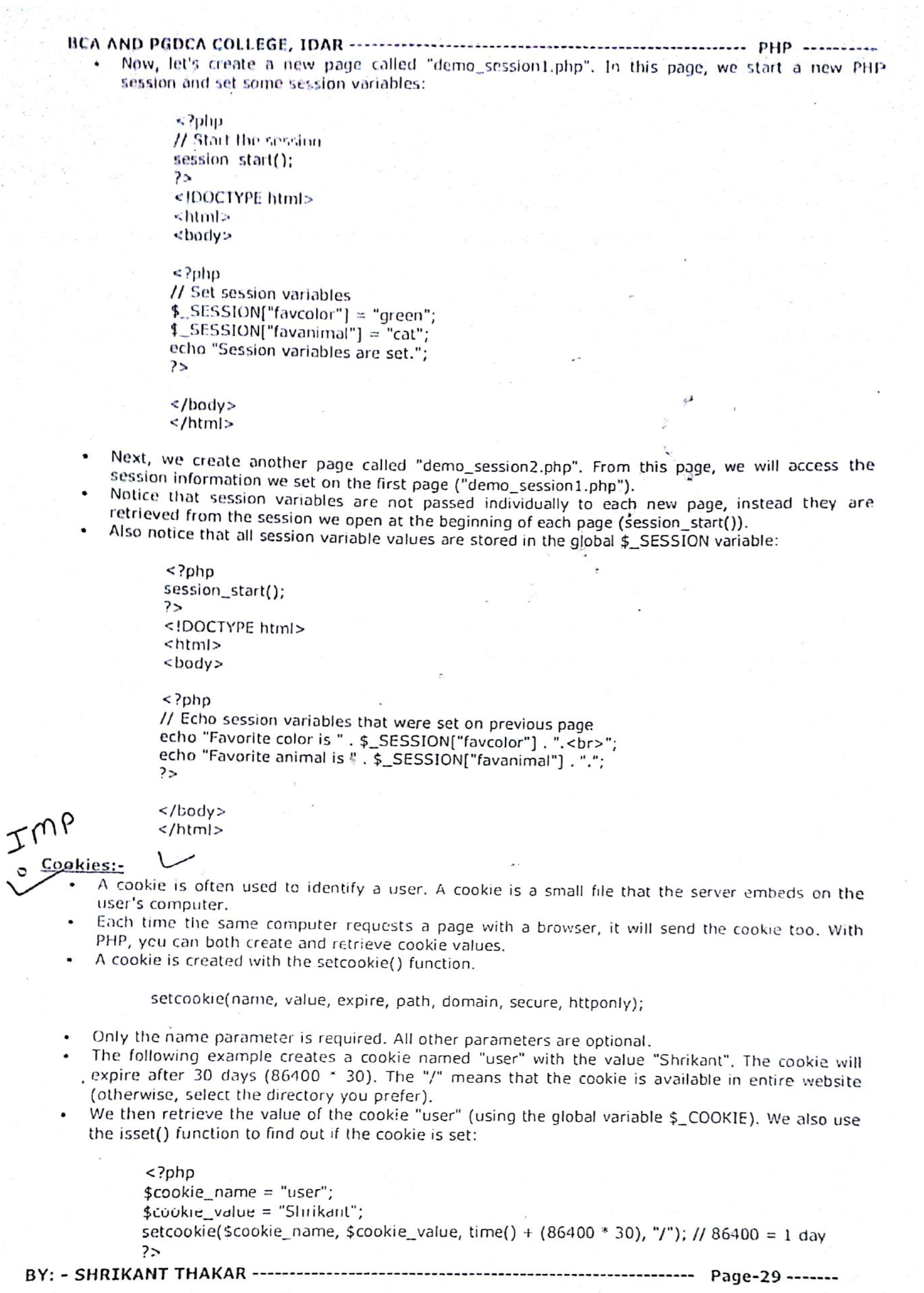
- When using sessions and cookies in your PHP applications, it is important to follow some best practices to ensure the security and reliability of your application. Here are some best practices to consider:
- Only store necessary data: Avoid storing sensitive data such as passwords or credit card numbers in sessions or cookies.
- Use secure cookies: When creating cookies, set the secure and httponly flags to ensure that the cookie is only transmitted over a secure connection and cannot be accessed by JavaScript.
- Use session_regenerate_id() function: Call the session_regenerate_id() function periodically to prevent session fixation attacks.
- Set session and cookie timeouts: Set a reasonable timeout period for your sessions

and cookies to ensure that they expire after a certain amount of time.



PHP Session:

- A session is a way to store information (in variables) to be used across multiple pages.
- Unlike a cookie, the information is not stored on the users computer.
- When you work with an application, you open it, do some changes, and then you close it.
- This is much like a Session. The computer knows who you are.
- It knows when you start the application and when you end.
- But on the internet there is one problem: the web server does not know who you are or what you do, because the HTTP address doesn't maintain state.
- 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.
- So; Session variables hold information about one single user, and are available to all pages in one application.
- Tip: If you need a permanent storage, you may want to store the data in a database.
- A session is started with the session_start() function.
- Session variables are set with the PHP global variable: $_SESSION.



- Now, let's create a new page called "demo_session1.php". In this page, we start a new PHP session and set some session variables:

```
<?php
// Start the session
session_start();
?>
<!DOCTYPE html>
<html>
<body>

<?php
// Set session variables
$_SESSION["favcolor"] = "green";
$_SESSION["favanimal"] = "cat";
echo "Session variables are set.";
?>

</body>
</html>
```

- Next, we create another page called "demo_session2.php". From this page, we will access the session information we set on the first page ("demo_session1.php").
- Notice that session variables are not passed individually to each new page, instead they are retrieved from the session we open at the beginning of each page (session_start()).
- Also notice that all session variable values are stored in the global $_SESSION variable:

```
<?php
session_start();
?>
<!DOCTYPE html>
<html>
<body>

<?php
// Echo session variables that were set on previous page
echo "Favorite color is " . $_SESSION["favcolor"] . ".<br>";
echo "Favorite animal is " . $_SESSION["favanimal"] . ".";
?>

</body>
</html>
```

IMP

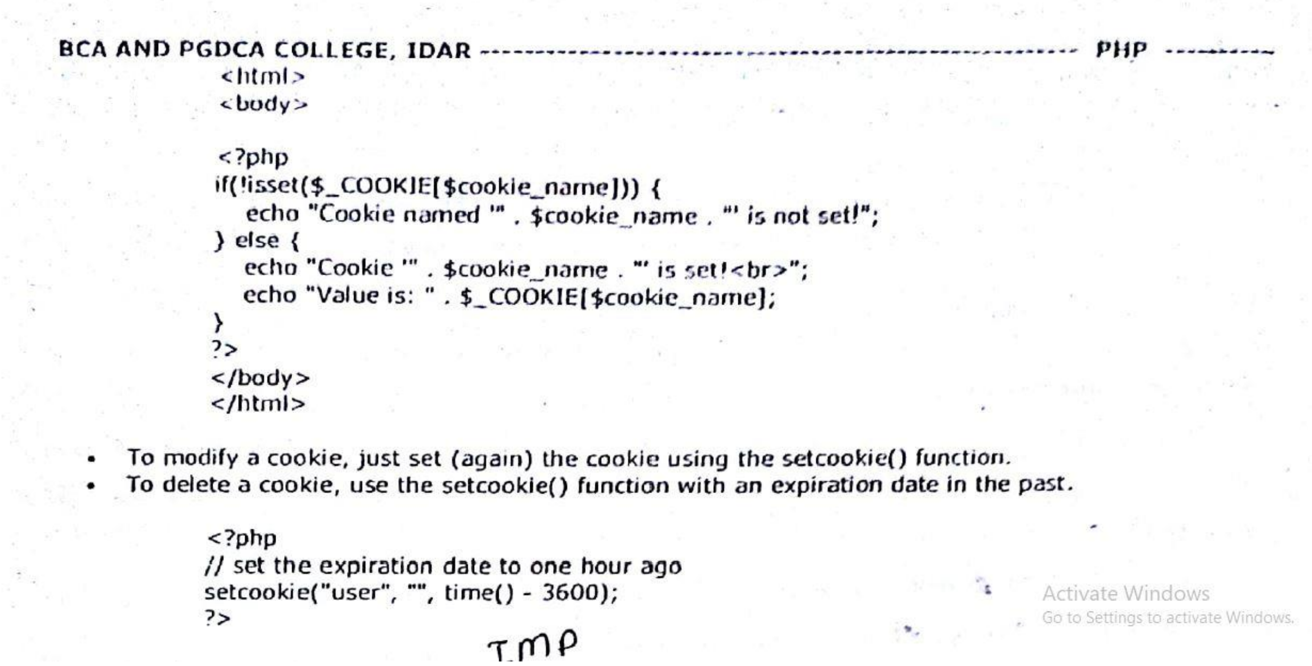
o **Cookies:-**

- A cookie is often used to identify a user. A cookie is a small file that the server embeds on the user's computer.
- Each time the same computer requests a page with a browser, it will send the cookie too. With PHP, you can both create and retrieve cookie values.
- A cookie is created with the setcookie() function.

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

- Only the name parameter is required. All other parameters are optional.
- The following example creates a cookie named "user" with the value "Shrikant". The cookie will expire after 30 days (86400 * 30). The "/" means that the cookie is available in entire website (otherwise, select the directory you prefer).
- We then retrieve the value of the cookie "user" (using the global variable $_COOKIE). We also use the isset() function to find out if the cookie is set:

```
<?php
$cookie_name = "user";
$cookie_value = "Shrikant";
setcookie($cookie_name, $cookie_value, time() + (86400 * 30), "/"); // 86400 = 1 day
?>
```



BCA AND PGDCA COLLEGE, IDAR -- PHP ---------

```
<html>
<body>

<?php
if(!isset($_COOKIE[$cookie_name])) {
   echo "Cookie named '" . $cookie_name . "' is not set!";
} else {
   echo "Cookie '" . $cookie_name . "' is set!<br>";
   echo "Value is: " . $_COOKIE[$cookie_name];
}
?>
</body>
</html>
```

- To modify a cookie, just set (again) the cookie using the setcookie() function.
- To delete a cookie, use the setcookie() function with an expiration date in the past.

```
<?php
// set the expiration date to one hour ago
setcookie("user", "", time() - 3600);
?>
```

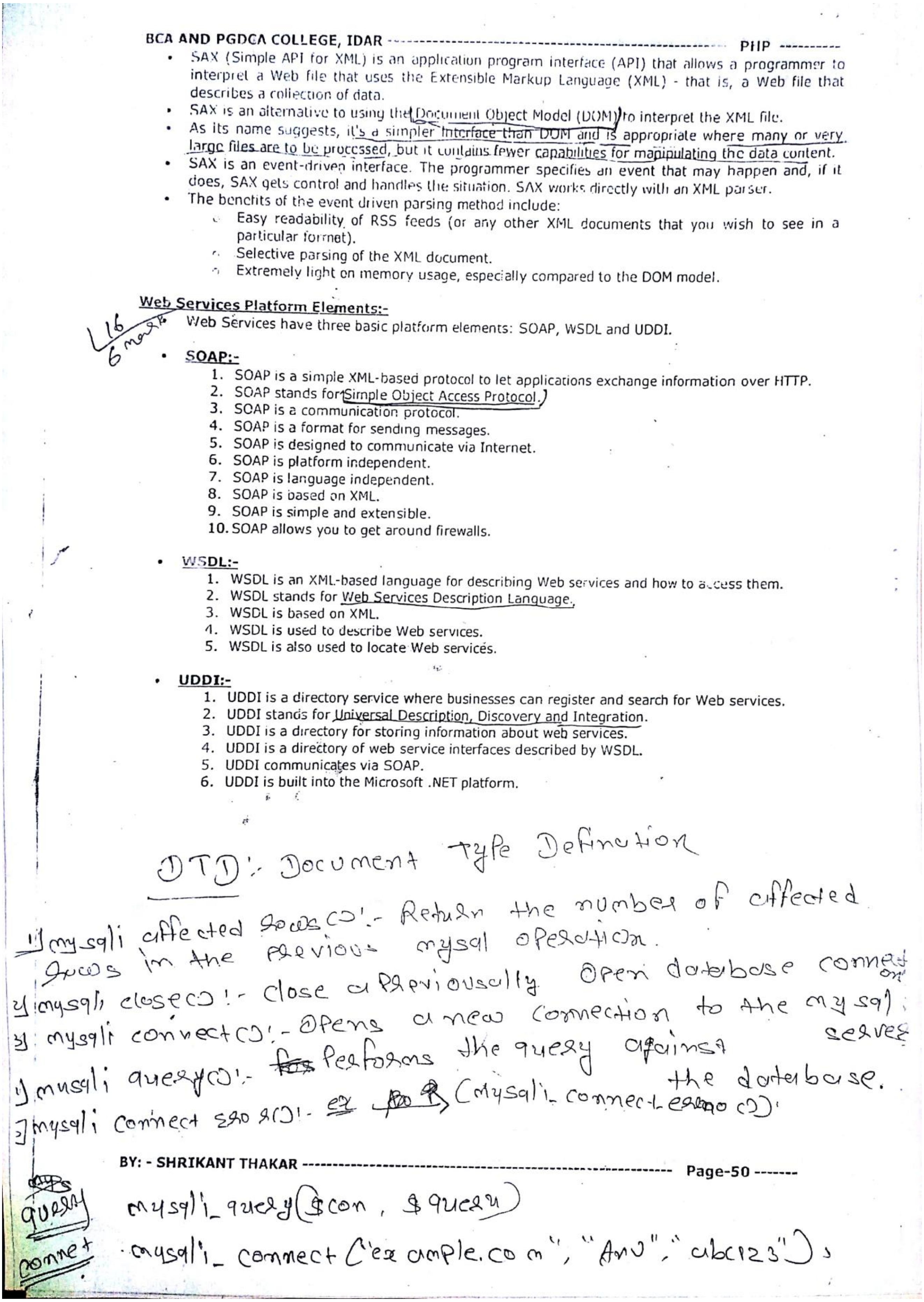
TMP

Web Service:

A web service is a set of open protocols and standards that allow data to be exchanged between different applications or systems. Web services can be used by software programs written in a variety of programming languages and running on a variety of platforms to exchange data via computer networks such as the Internet in a similar way to inter-process communication on a single computer.

Functions of Web Services

- It’s possible to access it via the internet or intranet networks.
- XML messaging protocol that is standardized.
- Operating system or programming language independent.
- Using the XML standard, it is self-describing.
- A simple location approach can be used to locate it.



- SAX (Simple API for XML) is an application program interface (API) that allows a programmer to interpret a Web file that uses the Extensible Markup Language (XML) - that is, a Web file that describes a collection of data.
- SAX is an alternative to using the Document Object Model (DOM) to interpret the XML file.
- As its name suggests, it's a simpler interface than DOM and is appropriate where many or very large files are to be processed, but it contains fewer capabilities for manipulating the data content.
- SAX is an event-driven interface. The programmer specifies an event that may happen and, if it does, SAX gets control and handles the situation. SAX works directly with an XML parser.
- The benefits of the event driven parsing method include:
  - Easy readability of RSS feeds (or any other XML documents that you wish to see in a particular format).
  - Selective parsing of the XML document.
  - Extremely light on memory usage, especially compared to the DOM model.

Web Services Platform Elements:-

L16 6 mark Web Services have three basic platform elements: SOAP, WSDL and UDDI.

- **SOAP:-**
  1. SOAP is a simple XML-based protocol to let applications exchange information over HTTP.
  2. SOAP stands for Simple Object Access Protocol.
  3. SOAP is a communication protocol.
  4. SOAP is a format for sending messages.
  5. SOAP is designed to communicate via Internet.
  6. SOAP is platform independent.
  7. SOAP is language independent.
  8. SOAP is based on XML.
  9. SOAP is simple and extensible.
  10. SOAP allows you to get around firewalls.
- **WSDL:-**
  1. WSDL is an XML-based language for describing Web services and how to access them.
  2. WSDL stands for Web Services Description Language.
  3. WSDL is based on XML.
  4. WSDL is used to describe Web services.
  5. WSDL is also used to locate Web services.
- **UDDI:-**
  1. UDDI is a directory service where businesses can register and search for Web services.
  2. UDDI stands for Universal Description, Discovery and Integration.
  3. UDDI is a directory for storing information about web services.
  4. UDDI is a directory of web service interfaces described by WSDL.
  5. UDDI communicates via SOAP.
  6. UDDI is built into the Microsoft .NET platform.

DTD :- Document Type Definetion

1) mysqli affected rows() :- Return the number of affected rows in the previous mysql operation.

2) mysqli close() :- close a previouscilly Open database connection

3) mysqli connect() :- Opens a new connection to the mysql server

4) mysqli query() :- ~~for~~ Performs the query against the database.

5) mysqli connect error() :- ex ~~[illegible]~~ → (mysqli_connect_errno());

~~[illegible]~~ query mysqli_query($con , $query)

connect mysqli_ connect ("example.com", "Anu", "abc123") ;