DATE

**1. Simple Date Conversion**

* Converts a date string to a timestamp.(string থেকে ডেইট অবজেক্ট এ নিয়ে যাবে।)

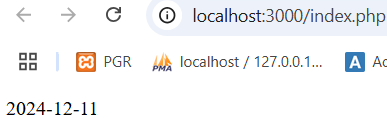
<?php

*//Converts a date string to a timestamp.*

    $date=strtotime("2024-12-11");

    echo date("Y-m-d",$date)."\n";

?>



**2. Relative Date and Time Parsing**

* strtotime() supports relative formats like "now," "tomorrow," "yesterday," and "+1 week."

<?php

*// 2. Relative Date and Time Parsing*

   $timestamp2 = strtotime("now");

   echo date("Y-m-d H:i:s", $timestamp2) . "<br>";  *// Current date and time*

   $timestamp3 = strtotime("tomorrow");

   echo date("Y-m-d", $timestamp3) . "<br>";  *// Tomorrow's date*

   $timestamp4 = strtotime("yesterday");

   echo date("Y-m-d", $timestamp4) . "<br>";  *// Yesterday's date*

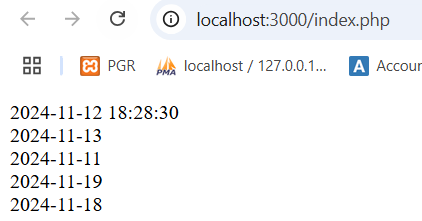
   $timestamp5 = strtotime("+1 week");

   echo date("Y-m-d", $timestamp5) . "<br>";  *// Date 1 week from now*

   $timestamp6 = strtotime("next Monday");

   echo date("Y-m-d", $timestamp6) . "<br>";  *// Next Monday's date*

?>



**3. Relative Days and Times**

* Use + or - to add or subtract days, weeks, months, years, hours, etc.

<?php

*// 3. Relative Days and Times*

    $timestamp7 = strtotime("+1 day");

    echo date("Y-m-d", $timestamp7) . "<br>";  *// Tomorrow's date*

    $timestamp8 = strtotime("-2 days");

    echo date("Y-m-d", $timestamp8) . "<br>";  *// Date 2 days ago*

    $timestamp9 = strtotime("+2 months");

    echo date("Y-m-d", $timestamp9) . "<br>";  *// Date 2 months from today*

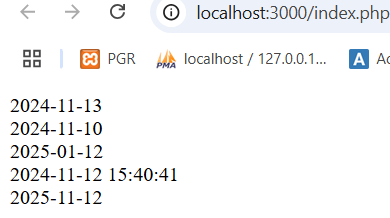
    $timestamp10 = strtotime("-3 hours");

    echo date("Y-m-d H:i:s", $timestamp10) . "<br>";  *// Date 3 hours ago*

    $timestamp11 = strtotime("+1 year");

    echo date("Y-m-d", $timestamp11) . "<br>";  *// Date 1 year from now*

?>



**4. Specific Day and Time Parsing**

* For a specific date with time.(string থেকে ডেইট এ।)

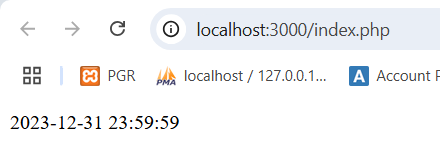
<?php

*// 4. Specific Day and Time Parsing*

    $timestamp12 = strtotime('2023-12-31 23:59:59');

    echo date('Y-m-d H:i:s', $timestamp12) . "<br>"; *// Last second of 2023*

?>



**5. Complex Date Calculation**

* Combine different date and time formats.

<?php

*// 5. Complex Date Calculation*

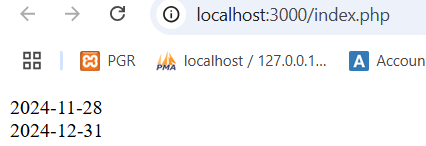
    $timestamp13 = strtotime('next Thursday +2 weeks');

    echo date('Y-m-d', $timestamp13) . "<br>"; *// Thursday two weeks from now*

    $timestamp14 = strtotime('last day of next month');

    echo date('Y-m-d', $timestamp14) . "<br>"; *// Last day of the next month*

?>



**6. Parsing Specific Times on a Given Day**

* Specify exact times on a relative day.

<?php

*// 6. Parsing Specific Times on a Given Day*

    $timestamp15 = strtotime('tomorrow noon');

    echo date('Y-m-d H:i:s', $timestamp15) . "<br>"; *// Noon tomorrow*

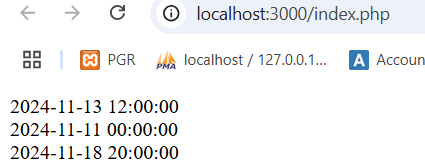
    $timestamp16 = strtotime('yesterday midnight');

    echo date('Y-m-d H:i:s', $timestamp16) . "<br>"; *// Midnight yesterday*

    $timestamp17 = strtotime('next Monday 8pm');

    echo date('Y-m-d H:i:s', $timestamp17) . "<br>"; *// 8 pm on the next Monday*

?>



**7. Handling Timezones**

* Adjust for timezone by specifying "UTC" or other timezone offsets.(বিভিন্ন টাইমজোন এ ডেইট প্রদর্শন।)

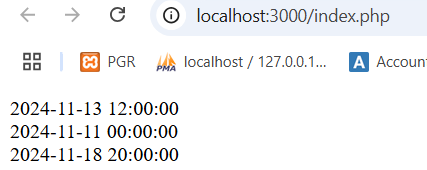
<?php

   date\_default\_timezone\_set('UTC');

   echo strtotime("now");              *// Current time in UTC*

   echo strtotime("2024-01-01 10:00 UTC"); *// Specific UTC time*

?>



**8. End or Start of Month, Week, Year**

* Handle beginning or end of month, week, or year.

<?php

*// 8. End or Start of Month, Week, Year*

    $timestamp19 = strtotime("first day of January 2024");

    echo date("Y-m-d", $timestamp19) . "<br>";  *// Start of January 2024*

    $timestamp20 = strtotime("last day of February 2024");

    echo date("Y-m-d", $timestamp20) . "<br>";  *// End of February 2024 (leap year)*

    $timestamp21 = strtotime("first day of this month");

    echo date("Y-m-d", $timestamp21) . "<br>";  *// Start of current month*

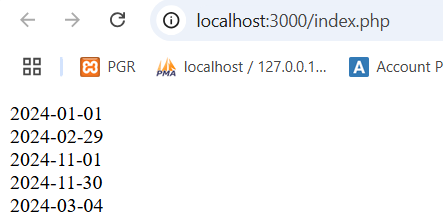
    $timestamp22 = strtotime("last day of this month");

    echo date("Y-m-d", $timestamp22) . "<br>";  *// End of current month*

    $timestamp23 = strtotime("first Monday of March 2024");

    echo date("Y-m-d", $timestamp23) . "<br>";  *// First Monday of March 2024*

?>



**9. Start and End of Day**

* Get the start or end of the current day.

<?php

*// 9. Start and End of Day*

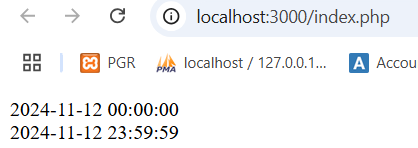
    $timestamp24 = strtotime('today midnight');

    echo date('Y-m-d H:i:s', $timestamp24) . "<br>"; *// Start of today*

    $timestamp25 = strtotime('tomorrow midnight -1 second');

    echo date('Y-m-d H:i:s', $timestamp25) . "<br>"; *// End of today*

?>



**10. Date Strings with Words**

* Parse date formats with day names and date keywords.

<?php

*// 10. Date Strings with Words*

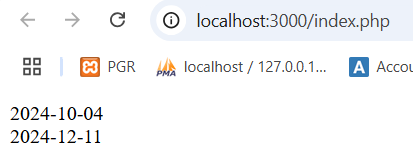
    $timestamp26 = strtotime("first Friday of October 2024");

    echo date("Y-m-d", $timestamp26) . "<br>";  *// First Friday of October 2024*

    $timestamp27 = strtotime("second Wednesday of next month");

    echo date("Y-m-d", $timestamp27) . "<br>";  *// Second Wednesday of next month*

?>



DATETIME

সবসময় আমরা DateTime UTC তে রাখব এবং পরে যে কোন নির্দিষ্ট টাইমজোন এ রিটার্ন/display করব।(UTC/GMT তে কোন অফসেট এ্যাড হয় না, মানে বাংলাদেশ GMT/UTC+6 তাই UTC তে ২ টা দেখাবে আর Asia তে ৮টা দেখাবে।)



<?php

*// 1.Create DateTime in UTC and Store in UTC for consistency*

    $date=new DateTime('now',new DateTimeZone('UTC'));

*// This is stored in UTC in the database*

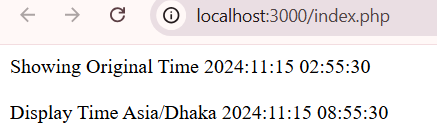
    echo 'Showing Original Time '.$date->format('Y:m:d H:i:s')."<br>";

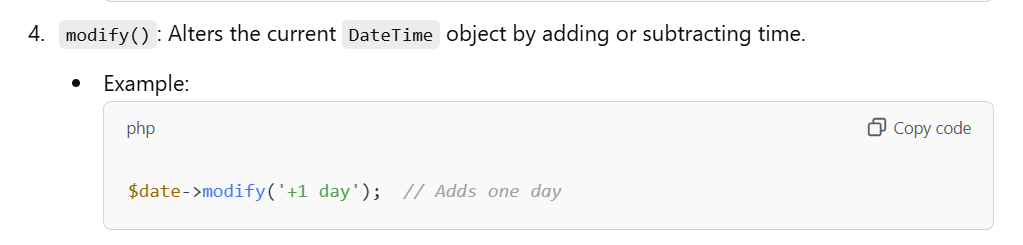
*//2. Retrieve and display in Asia/Dhaka*

    $date->setTimezone(new DateTimeZone('Asia/Dhaka'));

    echo 'Display Time Asia/Dhaka'.$date->format('Y:m:d H:i:s');

?>





<?php

*// 1.Create DateTime in UTC and Store in UTC for consistency*

    $date=new DateTime('now',new DateTimeZone('UTC'));

*//2. set timezone*

    $date->setTimezone(new DateTimeZone('Asia/Dhaka'));

    echo 'Original Date '.$date->format('Y:m:d H:i:s')."<br><br>";

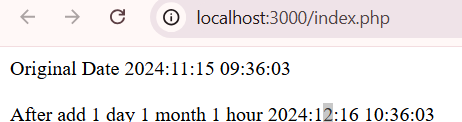
    $date->modify('+1 day');

    $date->modify('+1 month');

    $date->modify('+1 hour');

    echo 'After add 1 day 1 month 1 hour '.$date->format('Y:m:d H:i:s');

?>



DateInterval(P2Y1M10D)//2 year 1 month 10 days interval





এক্ষেত্রে ’now’ timezone set হবে UTC(no +6) এবং ইউজার থেকে নিব asia(UTC+6) যখন difference করব তখন এটা নিয়ে চিন্তা করতে হবে না ।PHP automatically handles time zone offsets.

<?php

*// 1.Create DateTime in UTC and Store in UTC for consistency*

    $date1 = new DateTime('now', new DateTimeZone('UTC'));

*//2. set timezone*

    $date2 = new DateTime('1994-04-20 08:30:00');

    $date2->setTimezone(new DateTimeZone('Asia/Dhaka'));

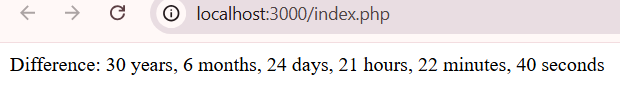
*// 4. Calculate the difference between 'now' (Asia/Dhaka) and the specific date (Asia/Dhaka)*

    $diff = $date1->diff($date2);

*// 5. Output the difference in years, months, days, hours, minutes, seconds*

    echo "Difference: " . $diff->format('%y years, %m months, %d days, %h hours, %i minutes, %s seconds') . "<br>";

?>



<?php

*// 1.Create DateTime in UTC and Store in UTC for consistency*

    $date1 = new DateTime('now', new DateTimeZone('UTC'));

*//2. set timezone*

    $date2 = new DateTime('1994-04-20 08:30:00');

    $date2->setTimezone(new DateTimeZone('Asia/Dhaka'));

    echo 'Showing Original Time '.$date2->format('Y:m:d H:i:s a')."<br><br>";

*// 4. Calculate the difference between 'now' (Asia/Dhaka) and the specific date (Asia/Dhaka)*

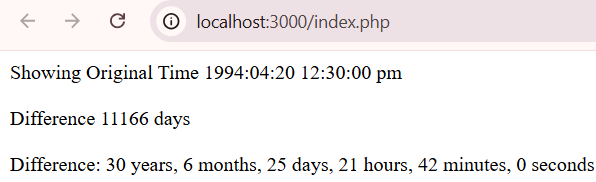
    $diff = $date2->diff($date1);

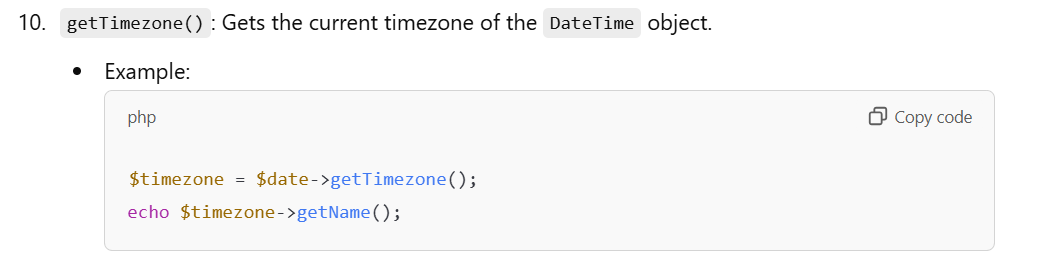
    echo "Difference ".$diff->format('%r%a days')."<br><br>";

*// 5. Output the difference in years, months, days, hours, minutes, seconds*

    echo "Difference: " . $diff->format('%y years, %m months, %d days, %h hours, %i minutes, %s seconds') . "<br>";

?>



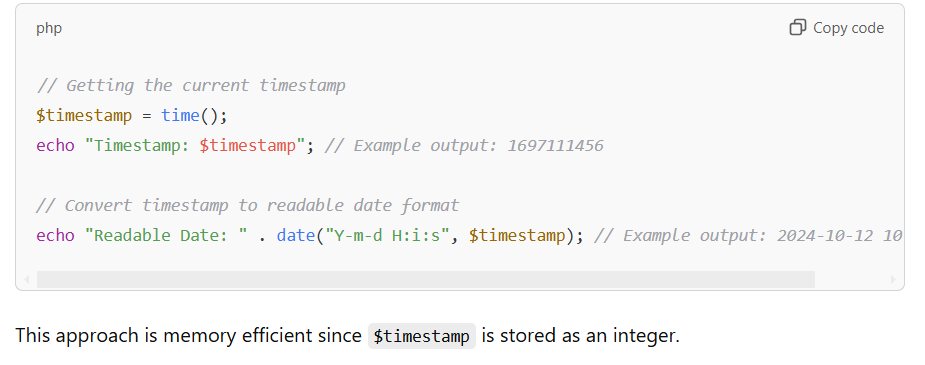


TIMESTAMP OR DATETIME?

### 1. ****Using Timestamp(এখানে 1971/1/1 এর ০ সেকেন্ড থেকে বর্তমান সময় পর‌্যন্ত calculate করবে সেকেন্ড/মিলিসেকেন্ড এ যা Numeric এ ডাটা স্টোর করবে ডাটাবেসে।)****

A timestamp in PHP is an integer representing seconds since the Unix Epoch (1970-01-01 00:00:00 UTC). You can get the current timestamp using time():

* If you use TIMESTAMP in MySQL, store the timestamp directly as an integer.



### 2. ****Using DateTime(এটা ব্যবহার করা বেটার।)****

The DateTime object offers more flexibility for formatting, manipulation, and timezone handling:

* If you use DATETIME, PHP's DateTime format can directly be stored in MySQL DATETIME columns.



**Array**

**Indexed Arrays**: Use numeric indices (either default or custom).

(এখানে ইনডেক্স 0,১,… থাকে)

Indexed array access করা যাবে for loop এবং foreach loop দিয়ে।

<?php

*//index array access by for loop and foreach loop*

    $indexed\_array=[10,20,30,"utsab","babu"];

*//for loop*

    for($i=0;$i<count($indexed\_array);$i++){

        echo "Index=$i value= $indexed\_array[$i] <br>";

    }

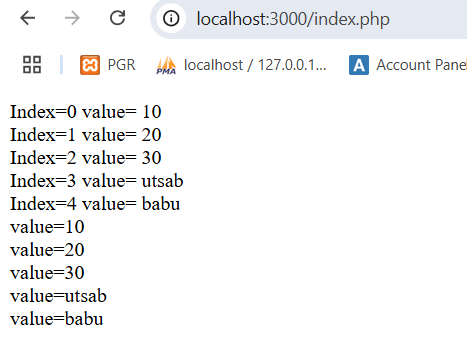
*//foreach*

    foreach($indexed\_array as $value){

        echo "value=$value<br>";

    }

?>



**Associative Array:** Use descriptive keys (either strings or integers) to associate with values.

[key value থাকবে এবং এটা foreach লুপ দিয়ে access করতে হবে।]

<?php

    $person=[

        "name"=>"utsab",

        "id"=>154061,

        "result"=>3.41

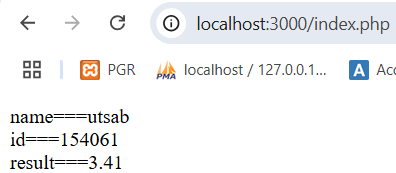
    ];

    foreach($person as $key=>$value){

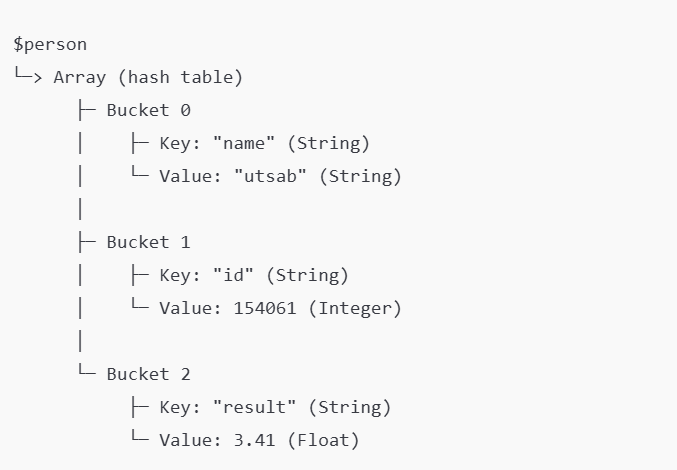
        echo "$key===$value<br>";

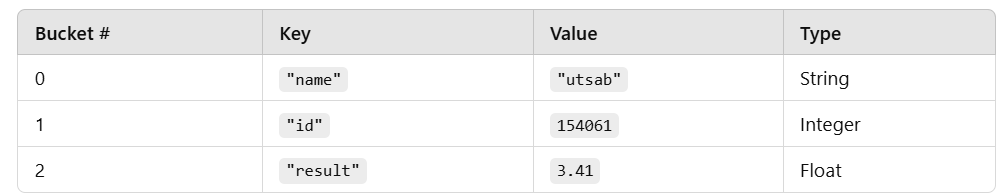
    }

?>

****

**Memory structure**

****

****

<?php

$students = [

    "student1" => [

        "name" => "Utsab",

        "subjects" => [

            "Math" => "A",

            "English" => "B+",

            "Science" => "A-"

        ]

    ],

    "student2" => [

        "name" => "Babu",

        "subjects" => [

            "Math" => "B",

            "English" => "A",

            "Science" => "B+"

        ]

    ],

    "student3" => [

        "name" => "Alice",

        "subjects" => [

            "Math" => "A-",

            "English" => "B",

            "Science" => "A"

        ]

    ]

];

*// Displaying data with foreach*

foreach ($students as $studentKey => $studentInfo) {

    echo "Student ID: $studentKey<br>";

    echo "Name: " . $studentInfo["name"] . "<br>";

    echo "Subjects and Grades:<br>";

    foreach ($studentInfo["subjects"] as $subject => $grade) {

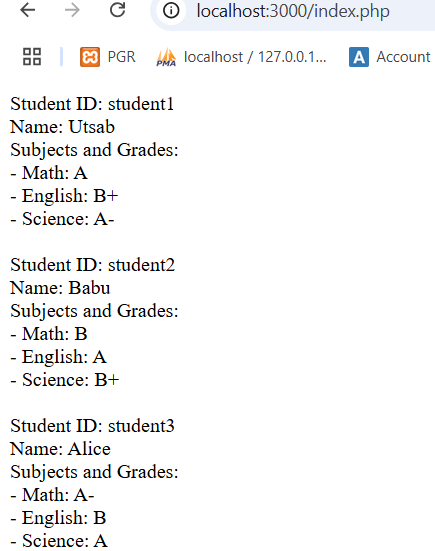
        echo "- $subject: $grade<br>";

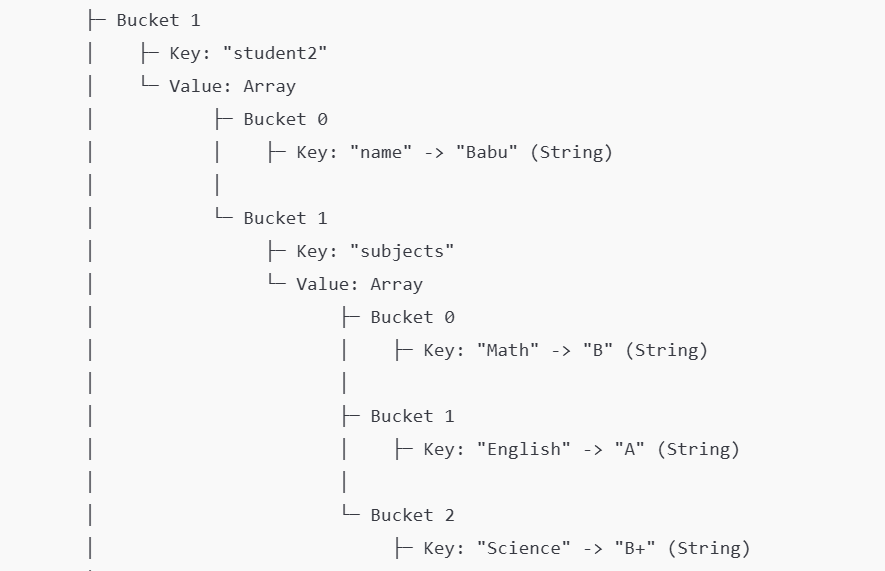
    }

    echo "<br>";

}

?>

****



3 loop

<?php

    $students=[

        "student1"=>[

            "name"=>"utsab",

            "subjects"=>[

                "math"=>"A+",

                "english"=>"B+",

                "science"=>"A"

            ]

        ],

        "student2"=>[

            "name"=>"babu",

            "subjects"=>[

                "math"=>"A",

                "english"=>"A-",

                "science"=>"Fail"

            ]

        ]

    ];

    foreach($students as $singleStudent=>$studentInfos){

        echo"Student Id $singleStudent <br>";

        foreach($studentInfos as $studentInfoKey=>$studentInfoValue){

            if($studentInfoKey=='subjects'){

                foreach($studentInfoValue as $singleSubject=>$singleSubjectResult){

                    echo"$singleSubject $singleSubjectResult<br>";

                }

            }else{

                echo" $studentInfoKey $studentInfoValue<br>";

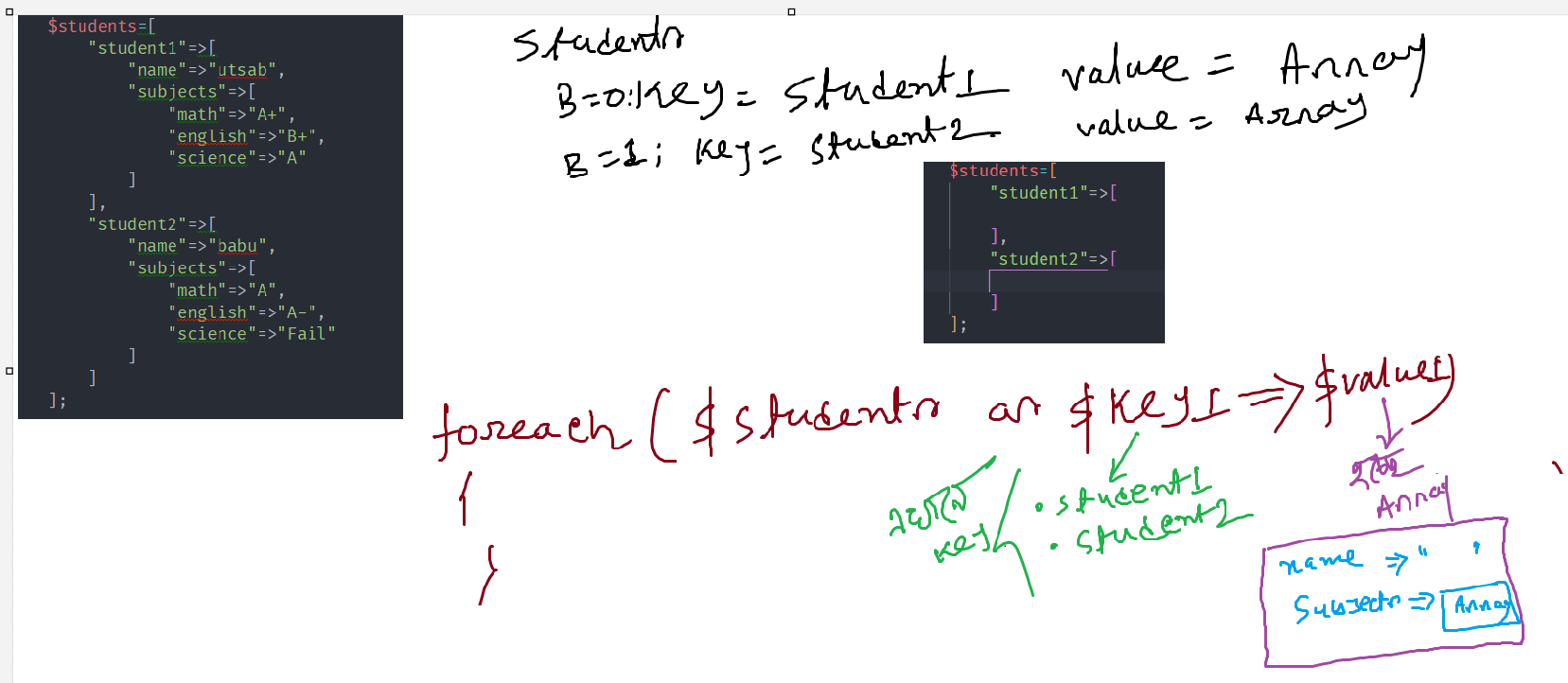
            }

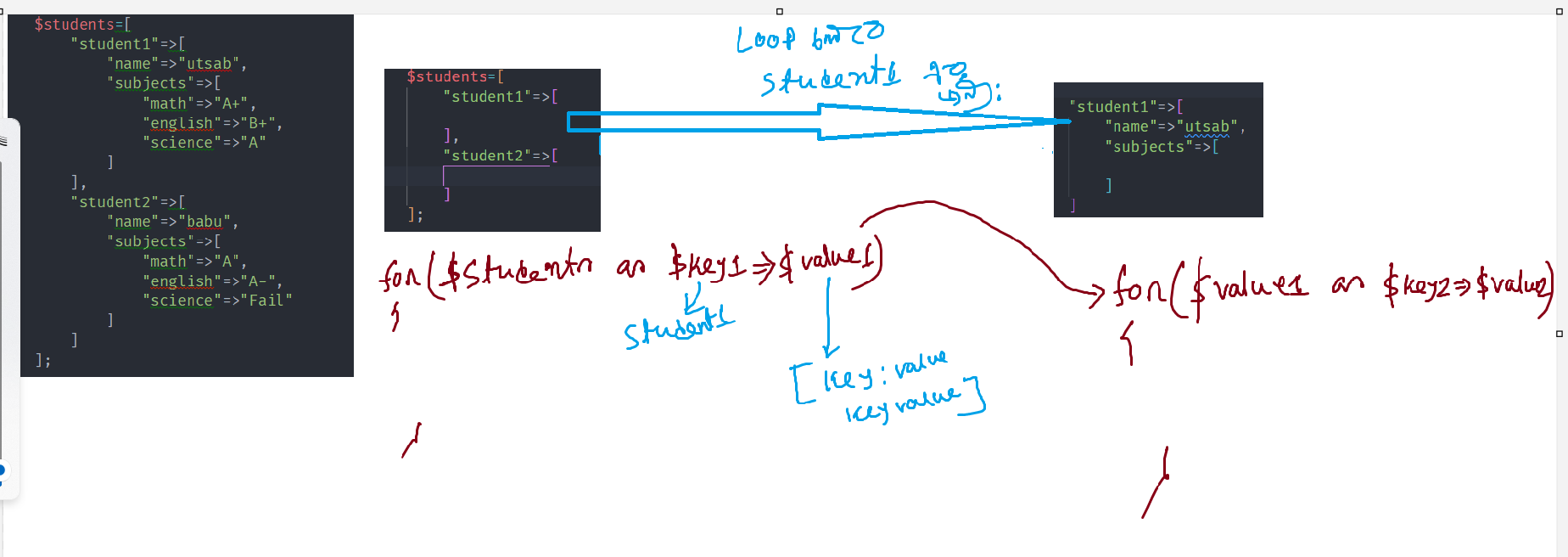
        }

        echo"<br><br>";

    }

?>





Array

**For Indexed Array To String Conversion**

 **implode()**: Concatenates array elements with a separator.

 **json\_encode()**: Converts to JSON format

 **serialize()**: Serializes the array for storage or transmission.

 **Manual concatenation**: Uses a loop for custom formatting.

 **print\_r() / var\_export()**: Outputs the array structure for debugging

<?php

*//The implode() function joins array elements into a single string, with an optional separator.*

*//it takes element from array and join using ,*

    $array = ['apple', 'banana', 'cherry'];

    $string = implode(', ', $array); *// Separator is ", "*

    echo $string."<br><br>"; *// Output: "apple, banana, cherry"*

*//The json\_encode() function converts an array into a JSON string,*

*// which can be useful for structured data.*

    $string\_1=json\_encode($array);

    echo $string\_1."<br><br>";

*//serialize() is a PHP-specific way to convert an array to a storable string format.*

*//It’s often used for storage or caching purposes.*

    $string\_2=serialize($array);

    echo $string\_2."<br><br>";

*//using foreach loop*

    $string\_3="";

    foreach($array as $single\_element){

        $string\_3=$string\_3.$single\_element;

    }

    echo $string\_3."<br><br>";

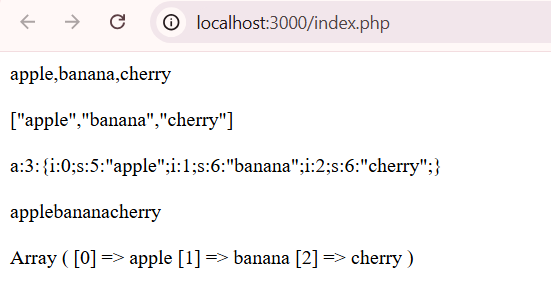
*//print\_r() and var\_export() convert the array to a readable string format.*

*//These are often used for debugging.*

    $string\_4 = print\_r($array, true);  *// `true` returns output as string*

    echo $string\_4."<br><br>";

?>



**Associative Arrays To String Conversion**(implode use করব না।)

* Use json\_encode() or serialize() for data transfer.
* Use a loop for custom key-value formatting.
* Use http\_build\_query() for URL-friendly format.

For associative arrays, you’ll need to handle both keys and values. Here are a few ways to convert them to strings:

<?php

    $assocArray = ['name' => 'Utsab', 'age' => 25, 'country' => 'Bangladesh'];

*//Using json\_encode()*

*//json\_encode() can be used to easily convert the array to a JSON-formatted string.*

    $string\_1 = json\_encode($assocArray);

    echo $string\_1 . '<br><br>';

*//Using a Loop to Concatenate Key-Value Pairs*

    $string\_2 = '';

    foreach ($assocArray as $key => $value) {

        $string\_2 = $string\_2 . "$key: $value, ";

    }

    $string\_2 = rtrim($string\_2, ', '); *// Remove the trailing/right comma and space*

    echo $string\_2 . '<br><br>';

*//If you need a URL query-like format, http\_build\_query() converts*

*// an associative array into a URL-encoded query string.*

*// Output: "name=Utsab&age=25&country=India"*

    $string\_3 = http\_build\_query($assocArray);

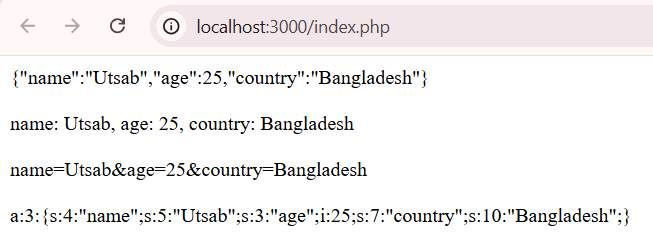
    echo $string\_3 . '<br><br>';

*//serialize() converts the array to a serialized string, which is useful for storage or data transfer.*

    $string\_4 = serialize($assocArray);

    echo $string\_4 . '<br><br>';

?>



**Indexed Array: String to Indexed Array**

**Indexed Array**:

* Use explode() for delimited strings.
* Use json\_decode() for JSON strings. .(true না দিলে অবজেক্ট দিবে। true দিলে Array পাব।)
* Use str\_split() for splitting by individual characters.

<?php

    $string = 'apple,banana,cherry';

*//If you have a simple delimited string (like a comma-separated list),*

*// use explode() to split it into an indexed array.*

    $indexed\_array\_1 = explode(',', $string);

    print\_r($indexed\_array\_1);

    echo '<br>';

    foreach ($indexed\_array\_1 as $value) {

        echo $value . ' ';

    }

    echo '<br><br>';

*//Using json\_decode() for JSON-Formatted Indexed Array*

*//If the string is JSON-encoded, use json\_decode().*

*// json\_decode($string)*

*//This converts a JSON-encoded string into a PHP array or object.*

*//By default, json\_decode() returns an object.If you want array, pass true as second parameter:*

    $string = '["apple,banana,cherry"]';

    $indexed\_array\_2 = json\_decode($string, true);

    foreach ($indexed\_array\_2 as $value) {

        echo $value . ' ';

    }

    echo '<br><br>';

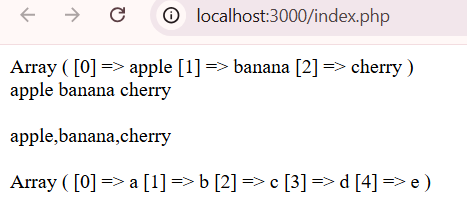
*//For single-character splitting, use str\_split().*

    $string = 'abcde';

    $indexedArray = str\_split($string); *// Splits each character*

    print\_r($indexedArray);

?>



**Associative Array: String to Indexed Array(explode দরকার নাই।)**

**Associative Array**:

* Use json\_decode() for JSON-formatted strings. .(true না দিলে অবজেক্ট দিবে। true দিলে Array পাব।)
* Use parse\_str() for query string format.
* Use unserialize() for serialized strings.
* Use explode() with a loop for custom key-value pairs.

<?php

*//Using json\_decode() for JSON-Formatted Associative Array*

*//If the string is JSON-encoded, json\_decode() works well.*

    $string = '{"name":"utsab","age":25,"country":"India"}';

    $associative\_array\_1 = json\_decode($string, true); *//return string to array*

    $associative\_array\_2 = json\_decode($string); *//return string to object*

    print\_r($associative\_array\_1);

    echo '<br><br>';

    print\_r($associative\_array\_2);

    echo"<br><br>";

    foreach($associative\_array\_1 as $key=>$value){

        echo "$key $value";

    }

    echo"<br><br>";

*//Using parse\_str() for Query String Format*

*//If the string is in a query string format (key=value&key2=value2), parse\_str() can be used.*

    $string = 'name=Utsab&age=25&country=India';

    parse\_str($string, $assocArray); *// Converts to an associative array*

    print\_r($assocArray);

    echo"<br><br>";

*//Using unserialize() for Serialized Associative Array*

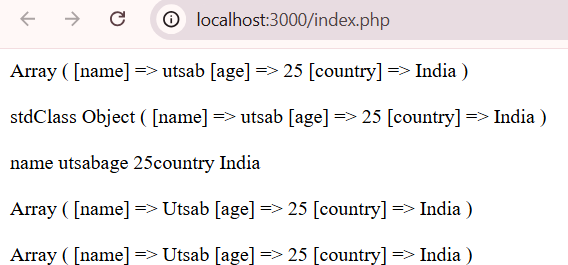
*//If the string was created with serialize(), use unserialize() to revert it back to an associative array.*

    $string = 'a:3:{s:4:"name";s:5:"Utsab";s:3:"age";i:25;s:7:"country";s:5:"India";}';

    $assocArray = unserialize($string);

    print\_r($assocArray);

?>



FILE OPERATION:

**Use file\_get\_contents() and file\_put\_contents() for Simplicity**  
 These functions are concise and efficient for most small to medium file operations.

(ছোট/মিডিয়াম ফাইল এর জন্য সহজে file\_get\_contents() দিয়ে read করব এবং file\_put\_contents() দিয়ে সহজে write করব।)

**file\_put\_contents() (string এ কনভার্ট করে নিতে হবে অবশ্যই।)**

<?php

    $input=[

        "name"=>"utsab",

        "id"=>"154061",

        "result"=>[

            "semester1"=>3.41,

            "semester2"=>3.42

        ]

    ];

*// Convert the array to a JSON string*

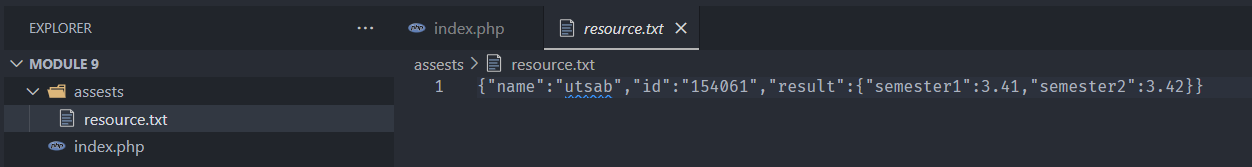
*//cannot write array/object to file directly*

    $converted\_input=json\_encode($input);

    file\_put\_contents("assests/resource.txt",$converted\_input);

?>

Output:



যদি আমরা Text File এ ঠিক মত/অবজেক্ট এর মত write করতে চায় তাহলে JSON\_PRETTY\_PRINT use করব।

<?php

    $input=[

        "name"=>"utsab",

        "id"=>"154061",

        "result"=>[

            "semester1"=>3.41,

            "semester2"=>3.42

        ]

    ];

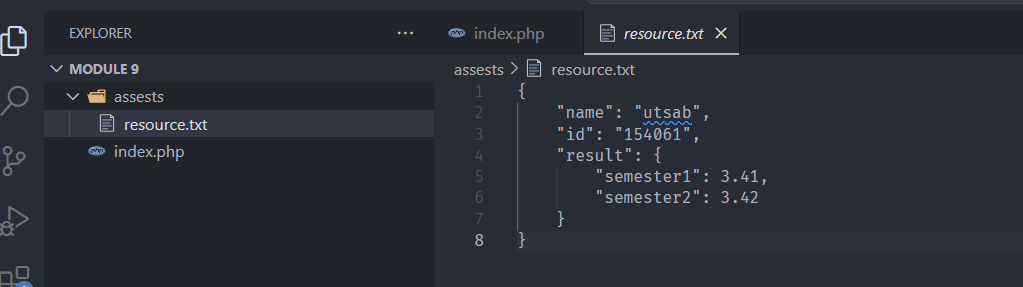
*// Convert the array to a JSON string*

*//cannot write array/object to file directly*

    $converted\_input=json\_encode($input,JSON\_PRETTY\_PRINT);

    file\_put\_contents("assests/resource.txt",$converted\_input);

?>



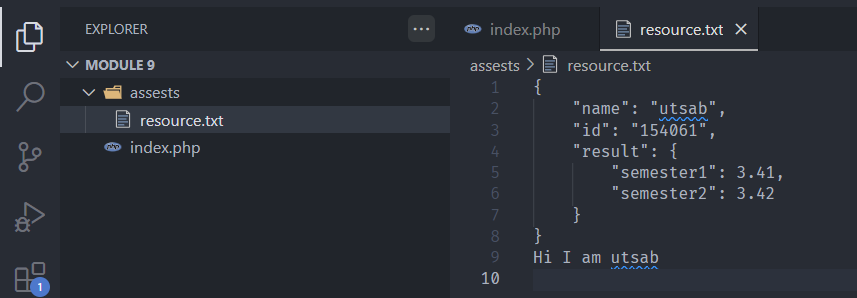
FILE\_APPEND এর মাধ্যমে নতুন লাইন এ্যাড হবে:

<?php

    $input="Hi I am utsab\n";

    file\_put\_contents("assests/resource.txt",$input,FILE\_APPEND);

?>

****

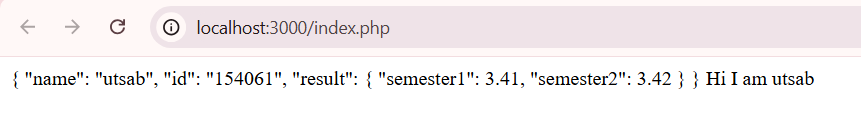
**file\_get\_contents()**

<?php

   $contents=file\_get\_contents("assests/resource.txt");

   echo $contents;

?>



**File() method for read**

<?php

*//using file*

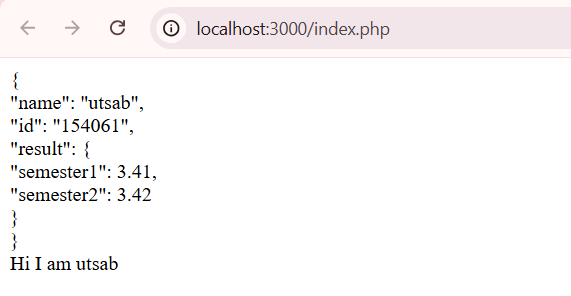
   $contents=file("assests/resource.txt",FILE\_IGNORE\_NEW\_LINES|FILE\_SKIP\_EMPTY\_LINES);

   foreach($contents as $line){

        echo $line."<br>";

   }

?>



**file\_get\_contents() দিয়ে json object read করব(json\_decode()):**

<?php

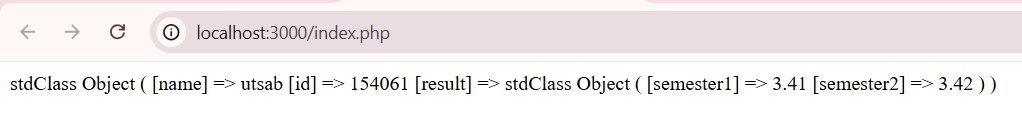
    $contents=file\_get\_contents("assests/resource.txt");

    $converted\_into\_json=json\_decode($contents);

*//var\_dump($converted\_into\_json);*

    print\_r($converted\_into\_json);

?>



**Large Files Read**

For **large files** or **incremental processing**, use fopen() with a loop (e.g., fgets()).

For large files, process line-by-line with **fgets()** to avoid memory overload.

<?php

    $file = fopen("assests/resource.txt", "r");

    if ($file) {

        while (($line = fgets($file)) !== false) {

            echo $line;

        }

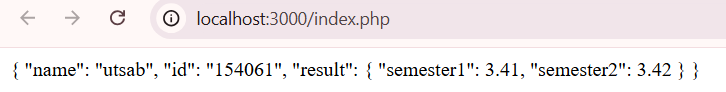
        fclose($file);

    } else {

        echo "Error opening file.";

    }

?>



File Locking  
Use file locking with flock() to prevent simultaneous read/write issues in a multi-user environment:

<?php

    $file = fopen('example.txt', 'w');

    if (flock($file, LOCK\_EX)) {

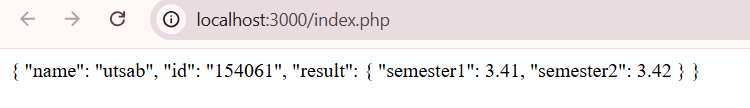
        fwrite($file, 'Successfully .');

        flock($file, LOCK\_UN);

    }

    fclose($file);

?>



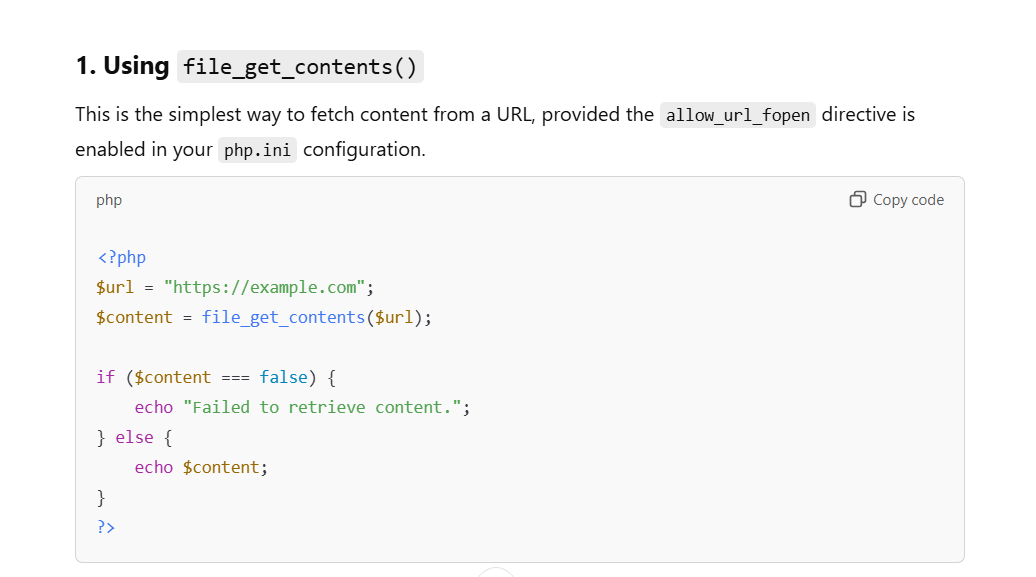
Content Read From URL

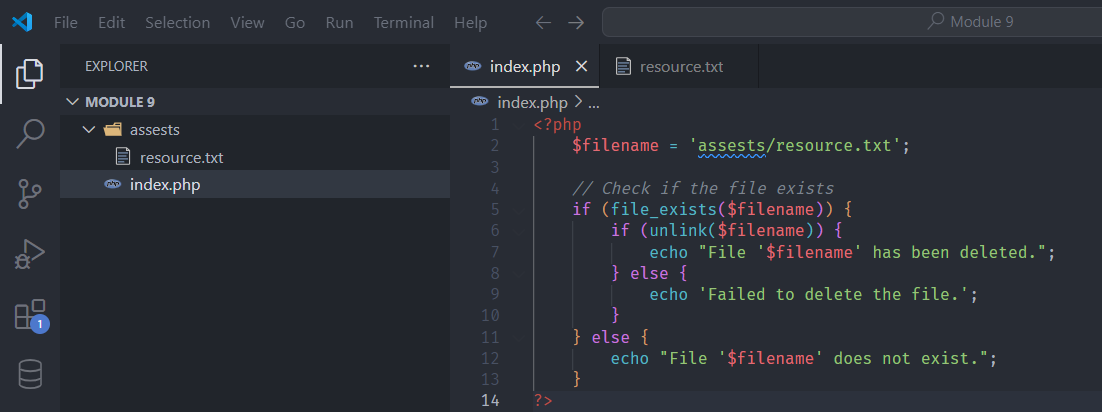
|  |  |
| --- | --- |
| **file\_get\_contents()** | For simple requests with minimal configuration. |

|  |  |
| --- | --- |
| **cURL** | When you need advanced options like headers, timeouts, or POST. |

|  |  |
| --- | --- |
| **stream\_context\_create()** | When using file\_get\_contents() but need extra control. |

|  |  |
| --- | --- |
| **Guzzle** | For complex or large-scale HTTP requests in modern applications. |



Delete File unlink()  


<?php

    $filename = 'assests/resource.txt';

*// Check if the file exists*

    if (file\_exists($filename)) {

        if (unlink($filename)) {

            echo "File '$filename' has been deleted.";

        } else {

            echo 'Failed to delete the file.';

        }

    } else {

        echo "File '$filename' does not exist.";

    }

?>

