

## What is Pagination in PHP?

- Pagination is the process of displaying the data on multiple pages rather than showing them on a single page.
- You usually do pagination when there is a database with numerous records.
- Dividing records increases the readability of the data.
- It can retrieve this data as per the user's requests.
- The most common example of pagination is the result page of a Google search. A search query entered into the search bar can fetch thousands of results. And it divides these results into multiple pages. You can click on any page number given at the bottom to visit that page.

## Advantages of Pagination

- Pagination in PHP is best suited for projects that require a sizeable amount of data because it adds professionalism to the work. Apart from that, it increases the readability of the project by segregating the code into pages and makes it look clean.
- It makes the web-pages work more efficiently. Pagination in PHP makes web-pages load in much lesser time, thus saving a lot of time and data.

Example: Suppose you want to access a website of images and there are two websites in front of you: the first one has 2000 images on a single web page while the other one has 20 web pages with 100 images on each page. Here, the website with a single page takes a lot more time to load as compared to the other website, because the server has to make a lot of HTTP requests and the page could become unresponsive. Whereas, pagination allows you to display a limited amount of data on a single page that in return limits the HTTP requests. Thus putting a lesser load on the server.

Since pagination allows the data to be distributed on several pages, it creates a great opportunity to generate huge revenue by increasing the advertisements on each page of the website.

## Disadvantages of Pagination

- Since pagination distributes the data and creates several pages, it may result in a low-rank page. This is because if a page is not directly linked to the home page and requires several clicks, the search engine ranks it on a lower level.
- One of the main reasons why pagination is avoided in many cases is that it is considered an enormous overhead in PHP. It adds unnecessary indirect costs of

styling, markup, data segregation, and logic to a project. Projects with small databases mostly ignore pagination.

- It can also make the search for specific data difficult.

Example: Suppose an exam has been conducted at your university and the results were uploaded to the university website. Now, you want to check your result specifically. It is already understood that a university has thousands of students segregated according to the branch and class. In this case, if the university has done pagination of the data then it must distribute the data on several pages and you have to check every page for your name unless the site has a “search by” algorithm. The addition of such an algorithm seems hard to implement most times and cannot be achieved so easily. Therefore, it is best to avoid pagination in such cases.

## Implementing Pagination with PHP and MySQL

The steps to follow to implement pagination with PHP and MySQL are:

- Create a large-scale database.
- Create a table and insert the list of records in it.
- Create a link for pagination to segregate/distribute the data over several different pages and add it to the bottom of the table created in step 2.
- Retrieve the data from the table and show it on the different pages.

Example:

```
<?php
```

```
include('database-conn.php');
```

```
if($con)
```

```
{
```

```
    $limit = 3;
```

```
    $q1 = "select * from user";
```

```
    $result = mysqli_query($con, $q1);
```

```
    $total_rows = mysqli_num_rows($result);
```

```
    // get the required number of pages
```

```
    $total_pages = ceil ($total_rows / $limit);
```

```
    // update the active page number
```

```
    if (!isset ($_GET['page']) )
```

```
{
```

```

    $page_number = 1;
} else {
    $page_number = $_GET['page'];
}
// get the initial page number
$initial_page = ($page_number-1) * $limit;

// get data of selected rows per page
$q = " select * from user LIMIT $initial_page, $limit";
$result = mysqli_query($con,$q);
?>

<table border="1" cellspacing="0" cellpadding="0">
<caption style='font-weight:bold'>User Information</caption>
<tr>
    <th width="10%">Sr. No</th>
    <th width="20%">Username</th>
    <th width="20%">Password</th>
    <th width="20%">Role</th>
    <th width="20%"></th>
</tr>

<?php
    $sr=1;
    while($r=mysqli_fetch_assoc($result))
    {
        echo "<tr>";
        echo "<td align='center'>".$sr."</td>";
        $u=$r['username'];
        echo "<td align='center'>".$r['username']."</td>";
        echo "<td align='center'>".$r['password']."</td>";
        echo "<td align='center'>".$r['role']."</td>";
        echo "<td align='center'><a href='database-edit.php?uname=$u'>Edit</a> | ";
        echo "<a href='database-delete.php?uname=$u'>Delete</a></td>";
        echo "</tr>";
        $sr++;
    }
    echo "</table>";

// show page number with link
for($page_number = 1; $page_number<= $total_pages; $page_number++)

```

```
{
    echo '<a href = "samepage.php?page='.$page_number.'">' . $page_number . '
</a>';
}
}
else{
    echo mysqli_error();
}
?>
```