

Simple Blog Activity

Overview: This will be our first basic application. It will be a very simple blog where a single administrator can login and enter a new blog entry consisting of a Title and Message.

Any web visitor can then view all the entries.

Admin section

- Create an /admin/ subdirectory.
- In this, create an insert.php file.
- In this file, create the necessary HTML form.

The image shows a web form titled "Simple Blog" with the subtitle "Insert New Blog Entry". Below the title is a label "Public Blog". The form contains two input fields: "Title:" and "Message:". The "Message:" field is a larger text area. At the bottom of the form is a button labeled "Submit Info".

-
- Create the necessary variables and conditional logic to capture the user information if the user has filled out the form and submitted it.
 - We will add simple validation that checks for fields NOT being empty.
- Add a variable for the time: `$timedate = date('l, F d, Y @ g:i a');`
- Create a custom function that will write the new blog entry:

```
function writeToBlog($thisTitle, $thisMessage, $thisTimedate){  
    //echo "$thisTitle, $thisMessage, $thisTimedate";  
  
    // To get blog entries in the correct order, we need to:  
    // Get the existing data from the textfile DB  
    // Get the current entry from the submitted form
```

DMIT2025 – PHP/MySQL

// Join the new entry to the existing (older) entries, then write everything back to the textfile.

```
$handle = fopen("blogfile.txt", "r"); //open the file for reading
if($handle){
    while(!feof($handle)){

        $buffer = fgets($handle, 4096);
        $existingText .= $buffer;
    } // end while
    //echo $existingText;
    fclose($handle);
} // end if

$handle = fopen("blogfile.txt", "w");
$newStuff = "\n<div class=\"newEntry\">";
$newStuff .= "\n\t<div class=\"title\">" . $thisTitle . "</div>";
$newStuff .= "\n\t<div class=\"timedate\">" . $thisTimedate . "</div>";
$newStuff .= "\n\t<div class=\"message\">" . $thisMessage . "</div>";
$newStuff .= "\n</div>";

$allStuff = $newStuff . $existingText;

fwrite($handle, $allStuff);
fclose($handle);

} // end writeToBlog
```

- Create a file called “blogfile.txt” in the admin section. This will be the data file. Change permissions (CHMOD 777) as necessary.

Output

In the root index.php page, all we need to do is include the datafile.

```
include ("admin/blogfile.txt");
```

The necessary styling can then be added.

DMIT2025 – PHP/MySQL



Securing the Admin Section

- Create a login.php page in the admin section. Do the secure login using sessions.
- Add a security check to the top of the insert.php page.
- Add an index.php to the admin section and use a redirect to the login page.



A screenshot of a web application titled "Simple Blog". The title is displayed in a large, stylized font on a dark blue background. To the right of the title, the word "Login" is written in a smaller, italicized font. Below the title, there is a white rectangular area containing a login form. The form has two input fields: "Username:" and "Password:". Below these fields is a button labeled "Submit Info". The entire form is enclosed in a light blue border.

-

Wrap Up

- Test and see how it works.
- Style the whole application.
- Note the limitations in using a flatfile for data.
 - We can't edit posts.
 - We can't search.
 - Etc.
 - All of these are good reasons most of the web has abandoned flatfiles and uses real databases.
 - We will be doing the same...soon.