**CC Mini Project Report**

**Title:**

Implementation of storage as a service specifically for images using web hosting.

**Abstract:**

The project implements Storage as a Service by making use of HTML and PHP and some basic CSS for interface. This project is specific for images. So a PHP script has been added to ensure that any other type of file is rejected by the system while uploading. Also, a size limit of roughly 5Mb has been added. At any point of time, the user can access any single file uploaded. If he needs all the files together, he can get them too. The files would be downloaded together in a Zip file.

**Introduction:**

The project is a file storage system for images wherin images can be uploaded and downloaded at convenience. File type and file size can be limited.

**Implementation details:**

Languages used were HTML,CSS and PHP.

HTML form was created and buttons were added for choosing,uploading and downloading files.

Basic CSS was used for interface.

PHP scripting was used implementing each modules.

Upload.php for uploading, display.php for displaying details about individual files and making it possible to download them and download.php to download all the files at once in a zip file.

**Code:**

**main.html**

<!DOCTYPE html>

<html>

<body>

<form action="upload.php" method="post" enctype="multipart/form-data">

Select image to upload:

<input type="file" name="fileToUpload" id="fileToUpload">

<input type="submit" value="Upload Image" name="submit">

</form>

<form action="download.php" method="post">

<input type="submit" value="Download">

</form>

<form action="display.php" method="post">

<input type="submit" value="Display file names">

</form>

</body>

</html>

**upload.php**

<?php

$target\_dir = "uploads/";

$target\_file = $target\_dir . basename($\_FILES["fileToUpload"]["name"]);

$uploadOk = 1;

$imageFileType = pathinfo($target\_file,PATHINFO\_EXTENSION);

// Check if image file is a actual image or fake image

if(isset($\_POST["submit"])) {

$check = getimagesize($\_FILES["fileToUpload"]["tmp\_name"]);

if($check !== false) {

echo "File is an image - " . $check["mime"] . ".";

$uploadOk = 1;

} else {

echo "File is not an image.";

$uploadOk = 0;

}

}

// Check if file already exists

if (file\_exists($target\_file)) {

echo "Sorry, file already exists.";

$uploadOk = 0;

}

// Check file size

if ($\_FILES["fileToUpload"]["size"] > 5000000) {

echo "Sorry, your file is too large.";

$uploadOk = 0;

}

// Check if $uploadOk is set to 0 by an error

if ($uploadOk == 0) {

echo "Sorry, your file was not uploaded.";

// if everything is ok, try to upload file

} else {

if (move\_uploaded\_file($\_FILES["fileToUpload"]["tmp\_name"], $target\_file)) {

echo "The file ". basename( $\_FILES["fileToUpload"]["name"]). " has been uploaded.";

} else {

echo "Sorry, there was an error uploading your file.";

}

}

?>

**display.php**

<?php

if ($handle = opendir('uploads/')) {

while (false !== ($file = readdir($handle))) {

if ($file != "." && $file != "..") {

echo "$file";

echo "<br>";

}

}

closedir($handle);

}

?>

**download.php**

<?php

$dir = 'uploads/';

$zip\_file = 'file.zip';

// Get real path for our folder

$rootPath = realpath($dir);

// Initialize archive object

$zip = new ZipArchive();

$zip->open($zip\_file, ZipArchive::CREATE | ZipArchive::OVERWRITE);

// Create recursive directory iterator

/\*\* @var SplFileInfo[] $files \*/

$files = new RecursiveIteratorIterator(

new RecursiveDirectoryIterator($rootPath),

RecursiveIteratorIterator::LEAVES\_ONLY

);

foreach ($files as $name => $file)

{

// Skip directories (they would be added automatically)

if (!$file->isDir())

{

// Get real and relative path for current file

$filePath = $file->getRealPath();

$relativePath = substr($filePath, strlen($rootPath) + 1);

// Add current file to archive

$zip->addFile($filePath, $relativePath);

}

}

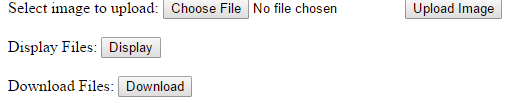
// Zip archive will be created only after closing object

$zip->close();

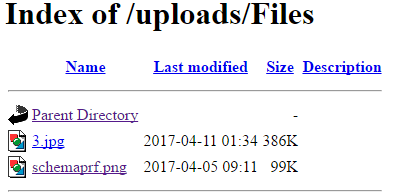
**Result:**

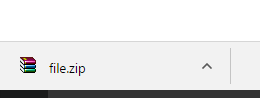
Images could be uploaded from anywhere. Details about individual uploaded files regarding their date of upload and their size is displayed successfully. Individual files can be downloaded at any time. Also, all the images as a whole can be downloaded in the form of a zip file at any time.

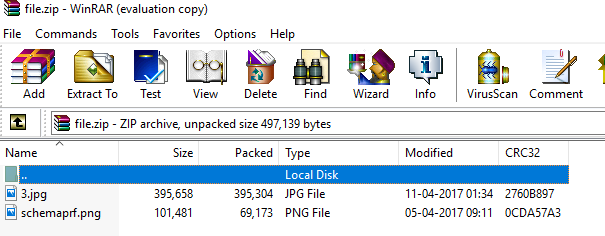
**Snapshots:**



2.PNG







**Conclusion:**

A cloud storage system for images was successfully implemented and hosted on the web under the domain <http://vhfilesharing.comeze.com/> .  
**References:**

[www.w3schools.com](http://www.w3school.com) For syntax

[www.stackoverflow.com](http://www.stackoverflow.com) For error debugging.