```
<?php
/*
 * ### CKFinder : Configuration File - Basic Instructions
 *
 * In a generic usage case, the following tasks must be done to configure
 * CKFinder:
 *     1. Check the $baseUrl and $baseDir variables;
 *     2. If available, paste your license key in the "LicenseKey" setting;
 *     3. Create the CheckAuthentication() function that enables CKFinder for authenticated users;
 *
 * Other settings may be left with their default values, or used to control
 * advanced features of CKFinder.
 */

/**
 * This function must check the user session to be sure that he/she is
 * authorized to upload and access files in the File Browser.
 *
 * @return boolean
 */
function CheckAuthentication()
{
        // WARNING : DO NOT simply return "true". By doing so, you are allowing
        // "anyone" to upload and list the files in your server. You must implement
        // some kind of session validation here. Even something very simple as...

        // return isset($_SESSION['IsAuthorized']) && $_SESSION['IsAuthorized'];

        // ... where $_SESSION['IsAuthorized'] is set to "true" as soon as the
        // user logs in your system.
        // To be able to use session variables don't forget to add session_start().

        return true;
}

// LicenseKey : Paste your license key here. If left blank, CKFinder will be
// fully functional, in demo mode.
$config['licenseName'] = '';
$config['licenseKey'] = '';

/*
 * Uncomment lines below to enable PHP error reporting and displaying PHP errors.
 * Do not do this on a production server. Might be helpful when debugging why CKFinder does not work as expected.
*/
// error_reporting(E_ALL);
// ini_set('display_errors', 1);

/*
To make it easy to configure CKFinder, the $baseUrl and $baseDir can be used.
Those are helper variables used later in this config file.

$baseUrl : the base path used to build the final URL for the resources handled
in CKFinder. If empty, the default value (/userfiles/) is used.

Examples:
        $baseUrl = 'http://example.com/ckfinder/files/';
        $baseUrl = '/userfiles/';

ATTENTION: The trailing slash is required.
*/
if($_SERVER['HTTP_HOST']=='192.168.10.2')
{
        $baseUrl = 'http://192.168.10.2/iiser/userfiles/';
}
else
{
        $baseUrl = 'http://www.iiserpune.ac.in/newsite/userfiles/';
}

/*
$baseDir : the path to the local directory (in the server) which points to the
above $baseUrl URL. This is the path used by CKFinder to handle the files in
the server. Full write permissions must be granted to this directory.

Examples:
        // You may point it to a directory directly:
        $baseDir = '/home/login/public_html/ckfinder/files/';
        $baseDir = 'C:/SiteDir/CKFinder/userfiles/';

        // Or you may let CKFinder discover the path, based on $baseUrl:
        $baseDir = resolveUrl($baseUrl);

ATTENTION: The trailing slash is required.
*/
//$baseDir = resolveUrl($baseUrl);
if($_SERVER['HTTP_HOST']=='192.168.10.2')
{
        $baseDir = '/var/www/html/iiser/userfiles/';

}
else
{
  $baseDir = $_SERVER['DOCUMENT_ROOT'].'/newsite/userfiles/' ;
}

//echo $baseDir;

/*
 * ### Advanced Settings
 */

/*
Thumbnails : thumbnails settings. All thumbnails will end up in the same
directory, no matter the resource type.
*/
$config['Thumbnails'] = Array(
                'url' => $baseUrl . '_thumbs',
                'directory' => $baseDir . '_thumbs',
                'enabled' => true,
                'directAccess' => false,
                'maxWidth' => 100,
                'maxHeight' => 100,
                'bmpSupported' => false,
                'quality' => 80);

/*
Set the maximum size of uploaded images. If an uploaded image is larger, it
gets scaled down proportionally. Set to 0 to disable this feature.
*/
$config['Images'] = Array(
                'maxWidth' => 1600,
                'maxHeight' => 1200,
                'quality' => 80);

/*
RoleSessionVar : the session variable name that CKFinder must use to retrieve
the "role" of the current user. The "role", can be used in the "AccessControl"
settings (bellow in this page).

To be able to use this feature, you must initialize the session data by
uncommenting the following "session_start()" call.
*/
$config['RoleSessionVar'] = 'CKFinder_UserRole';
//session_start();

/*
AccessControl : used to restrict access or features to specific folders.

Many "AccessControl" entries can be added. All attributes are optional.
Subfolders inherit their default settings from their parents' definitions.

        - The "role" attribute accepts the special '*' value, which means
          "everybody".
        - The "resourceType" attribute accepts the special value '*', which
          means "all resource types".
*/

$config['AccessControl'][] = Array(
                'role' => '*',
                'resourceType' => '*',
                'folder' => '/',

                'folderView' => true,
                'folderCreate' => true,
                'folderRename' => true,
                'folderDelete' => true,

                'fileView' => true,
                'fileUpload' => true,
                'fileRename' => true,
                'fileDelete' => true);

/*
For example, if you want to restrict the upload, rename or delete of files in
the "Logos" folder of the resource type "Images", you may uncomment the
following definition, leaving the above one:

$config['AccessControl'][] = Array(
                'role' => '*',
                'resourceType' => 'Images',
                'folder' => '/Logos',

                'fileUpload' => false,
                'fileRename' => false,
                'fileDelete' => false);
*/

/*
ResourceType : defines the "resource types" handled in CKFinder. A resource
type is nothing more than a way to group files under different paths, each one
having different configuration settings.

Each resource type name must be unique.

When loading CKFinder, the "type" querystring parameter can be used to display
a specific type only. If "type" is omitted in the URL, the
"DefaultResourceTypes" settings is used (may contain the resource type names
separated by a comma). If left empty, all types are loaded.

maxSize is defined in bytes, but shorthand notation may be also used.
Available options are: G, M, K (case insensitive).
1M equals 1048576 bytes (one Megabyte), 1K equals 1024 bytes (one Kilobyte), 1G equals one Gigabyte.
Example: 'maxSize' => "8M",
*/
$config['DefaultResourceTypes'] = '';

$config['ResourceType'][] = Array(
                'name' => 'files',                                      // Single quotes not allowed
                'url' => $baseUrl . 'files',
                'directory' => $baseDir . 'files',
                'maxSize' => 0,
                'allowedExtensions' =>
'7z,aiff,asf,avi,bmp,csv,doc,fla,flv,gif,gz,gzip,jpeg,jpg,mid,mov,mp3,mp4,mpc,mpeg,mpg,ods,odt,pdf,png,ppt,pxd,qt,ram,rar,rm,rmi,rmvb,rtf,sdc,sitd,swf,
sxc,sxw,tar,tgz,tif,tiff,txt,vsd,wav,wma,wmv,xls,zip',
                'deniedExtensions' => '');

$config['ResourceType'][] = Array(
                'name' => 'Images',
                'url' => $baseUrl . 'images',
                'directory' => $baseDir . 'images',
                'maxSize' => 0,
                'allowedExtensions' => 'bmp,gif,jpeg,jpg,png',
                'deniedExtensions' => '');

$config['ResourceType'][] = Array(
                'name' => 'flash',
                'url' => $baseUrl . 'flash',
                'directory' => $baseDir . 'flash',
                'maxSize' => 0,
                'allowedExtensions' => 'swf,flv',
                'deniedExtensions' => '');

/*
 Due to security issues with Apache modules, it is recommended to leave the
 following setting enabled.

 How does it work? Suppose the following:

        - If "php" is on the denied extensions list, a file named foo.php cannot be
          uploaded.
        - If "rar" (or any other) extension is allowed, one can upload a file named
          foo.rar.
        - The file foo.php.rar has "rar" extension so, in theory, it can be also
          uploaded.

In some conditions Apache can treat the foo.php.rar file just like any PHP
script and execute it.

If CheckDoubleExtension is enabled, each part of the file name after a dot is
checked, not only the last part. In this way, uploading foo.php.rar would be
denied, because "php" is on the denied extensions list.
*/
$config['CheckDoubleExtension'] = true;

/*
If you have iconv enabled (visit http://php.net/iconv for more information),
you can use this directive to specify the encoding of file names in your
system. Acceptable values can be found at:
        http://www.gnu.org/software/libiconv/

Examples:
        $config['FilesystemEncoding'] = 'CP1250';
        $config['FilesystemEncoding'] = 'ISO-8859-2';
*/
$config['FilesystemEncoding'] = 'UTF-8';

/*
Perform additional checks for image files
if set to true, validate image size
*/
$config['SecureImageUploads'] = true;

/*
Indicates that the file size (maxSize) for images must be checked only
after scaling them. Otherwise, it is checked right after uploading.
*/
$config['CheckSizeAfterScaling'] = true;

/*
For security, HTML is allowed in the first Kb of data for files having the
following extensions only.
*/
$config['HtmlExtensions'] = array('html', 'htm', 'xml', 'js');

/*
Folders to not display in CKFinder, no matter their location.
No paths are accepted, only the folder name.
The * and ? wildcards are accepted.
*/
$config['HideFolders'] = Array(".svn", "CVS");

/*
Files to not display in CKFinder, no matter their location.
No paths are accepted, only the file name, including extension.
The * and ? wildcards are accepted.
*/
$config['HideFiles'] = Array(".*");

/*
After file is uploaded, sometimes it is required to change its permissions
so that it was possible to access it at the later time.
If possible, it is recommended to set more restrictive permissions, like 0755.
Set to 0 to disable this feature.
Note: not needed on Windows-based servers.
*/
$config['ChmodFiles'] = 0777 ;

/*
See comments above.
Used when creating folders that does not exist.
*/
$config['ChmodFolders'] = 0755 ;

/*
Force ASCII names for files and folders.
If enabled, characters with diactric marks, like à, ä, ö, ć, č, đ, ś
will be automatically converted to ASCII letters.
*/
$config['forceAscii'] = false;
```