

Wordpress

Technical Documentation & Guidelines

Table of Contents

[Introduction](#)

[Installing WordPress](#)

[Installation Steps](#)

[WordPress - Back-office/Admin area.](#)

[Posts section in admin Panel](#)

[Appearance Section in Admin Panel](#)

[Appearance > Themes](#)

[Appearance > Menus](#)

[Plugins Section in Admin Panel](#)

[Settings Section in Admin Panel](#)

[Media](#)

[Wordpress files](#)

[WordPress Main Folder](#)

[WP-Admin Folder](#)

[wp-content](#)

[wp-content/themes](#)

[wp-includes](#)

[Adding Logo Image](#)

[Adding a Favicon](#)

[Wordpress Menu](#)

[Including stylesheet / javascript files](#)

[How to show breadcrumbs](#)

[Trim Headings and Description](#)

[Custom Image Thumbnails](#)

[Show the Custom image sizes in admin](#)

[Specialized Page Template](#)

[Custom Page Templates](#)

[Conditional Tags](#)

[Custom Post Types](#)

[Admin Editor](#)

[WordPress Shortcode](#)

[Attributes in shortcodes](#)

[Wordpress CSS Coding Standard](#)

[Structure](#)

[Property Ordering](#)

[Selectors](#)

[Comments](#)

[Properties value](#)

[Media Queries](#)

[Useful Plugins](#)

[Reference Links](#)

Project

Project hosting: /

Tech Doc edited by: TL

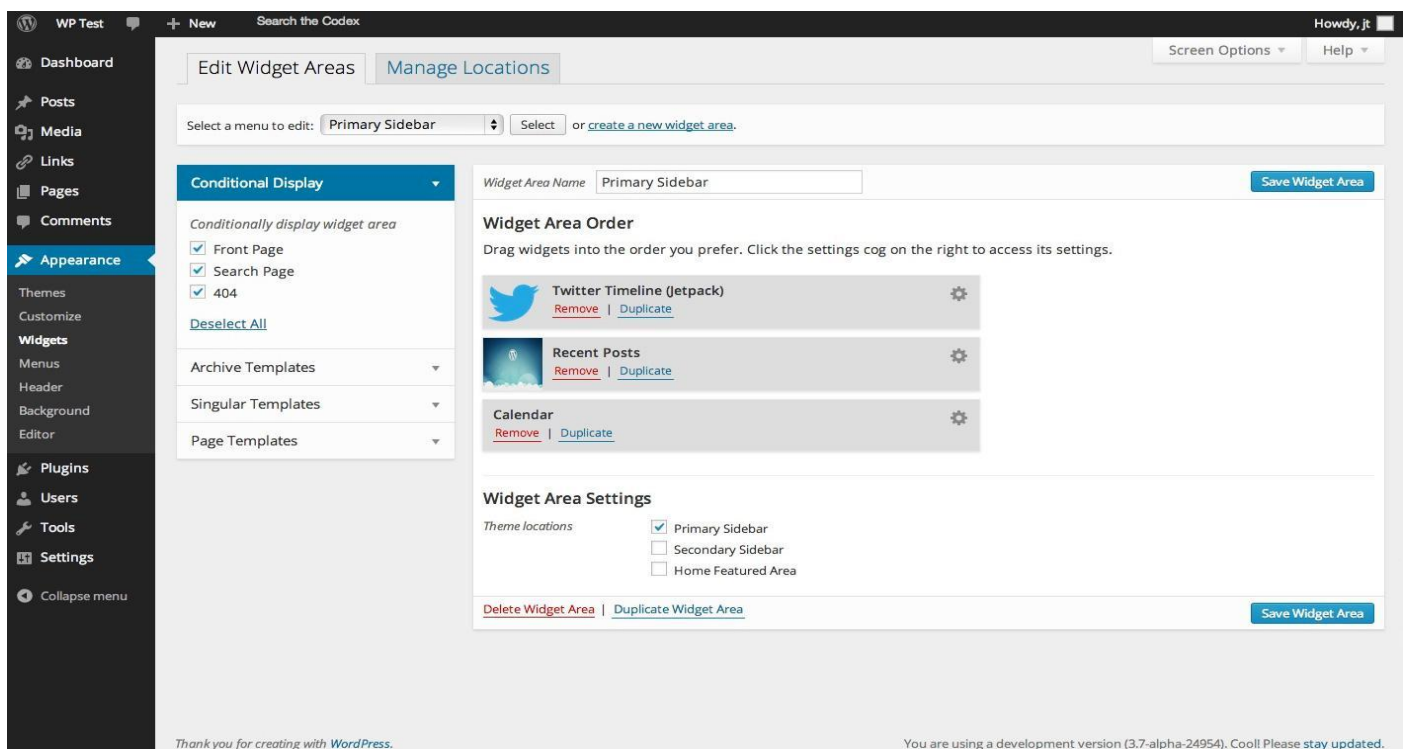
Developer: Narendran

Introduction



WordPress is a free and open-source blogging tool and a content management system (CMS) based on PHP and MySQL. Features include a plugin architecture and a template system.

One of the best things about WordPress is that it makes it easy to publish content and Ultimate in SEO Friendliness. WordPress has a simple user interface and provides more themes and plugins than any other CMS. It is a widely used platform and has a large community of developers, designers, and users.



Default Wordpress admin panel

Interesting Statistics

- 19,000 free plugins available for download.
- 74.6 Million Sites Depend on WordPress. (74,652,825*)
- 48 of Technorati's Top 100 Blogs Are Managed With WordPress (Mashable, Techcrunch, etc.,)
- 6 New WordPress.com Posts Every Second
- 40 Translations of WordPress
- WordPress.com Employs Only 229 People

- 98 Versions of WordPress to Date (Latest: 4.1)
- Akismet Is the Most Popular Plugin (Comment spam prevention for wordpress blog).

Installing WordPress

Wordpress is well-known for its famous 5-Minute Install. Under most circumstances, installing WordPress is a very simple process and takes less than five minutes to complete. Many web hosts now offer tools (e.g. Fantastico) to automatically install WordPress for you. However, if you wish to install WordPress yourself, the following guide will help.

The following installation guide will help you to understand the process.

Things to Know Before Installing WordPress

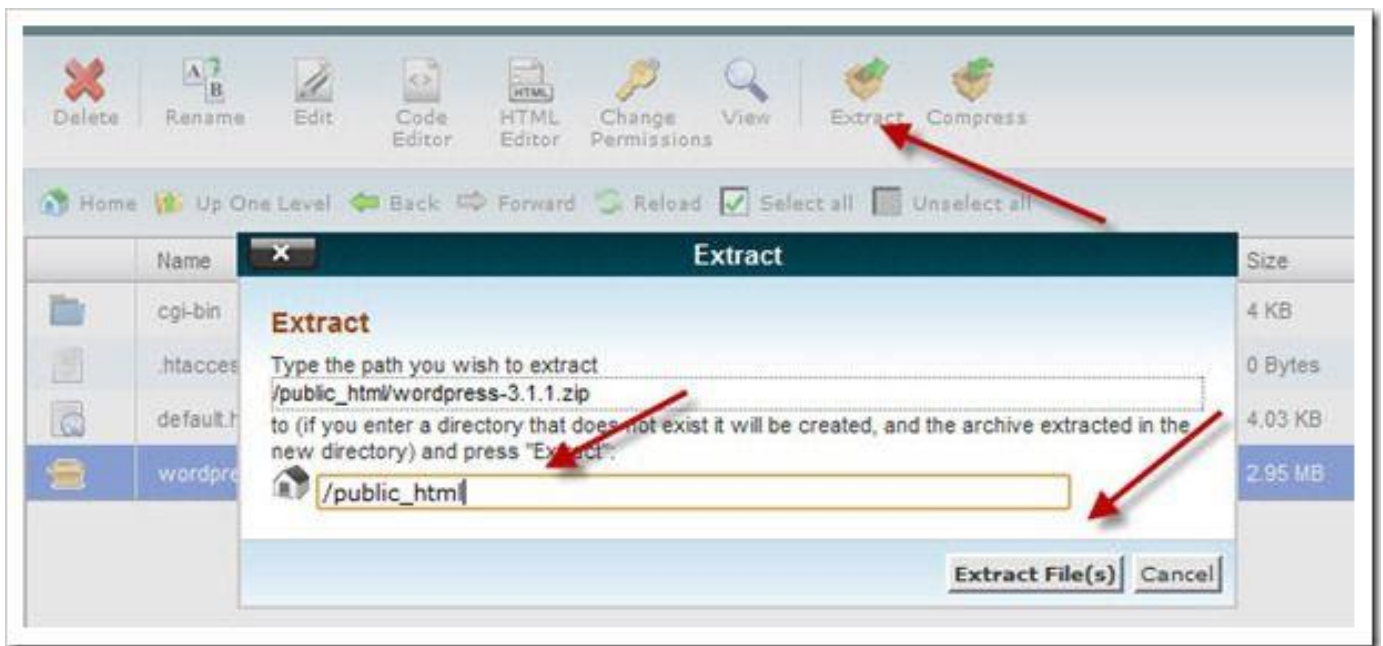
Before you begin the install, there are a few things you need to have and do.

- Access to your web server (via shell or FTP)
- Your Web Host with PHP version 5.2.4 or greater (recommended: PHP 5.4 or greater)
- MySQL version 5.0 or greater (recommended: MySQL 5.5 or greater)
- WordPress latest build
- A secure password for your Secret Key
- HTML Editor

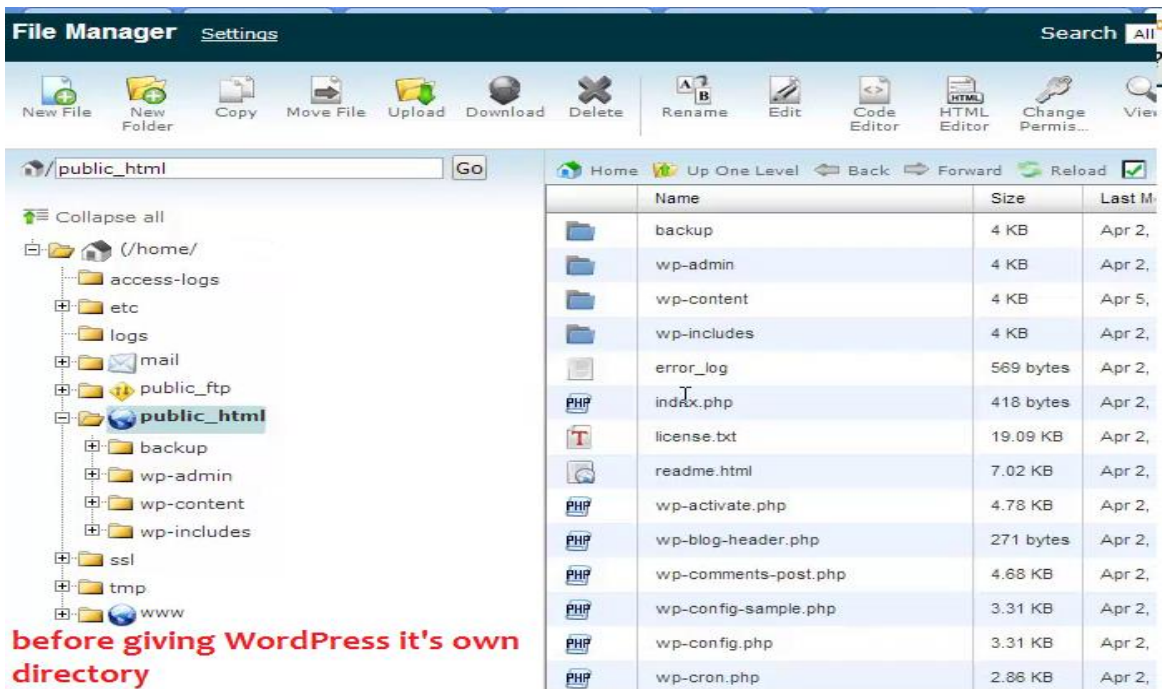
Installation Steps

Once you are ok with the above requirements, follow the below steps.

1: Download and unzip the WordPress package if you haven't already.

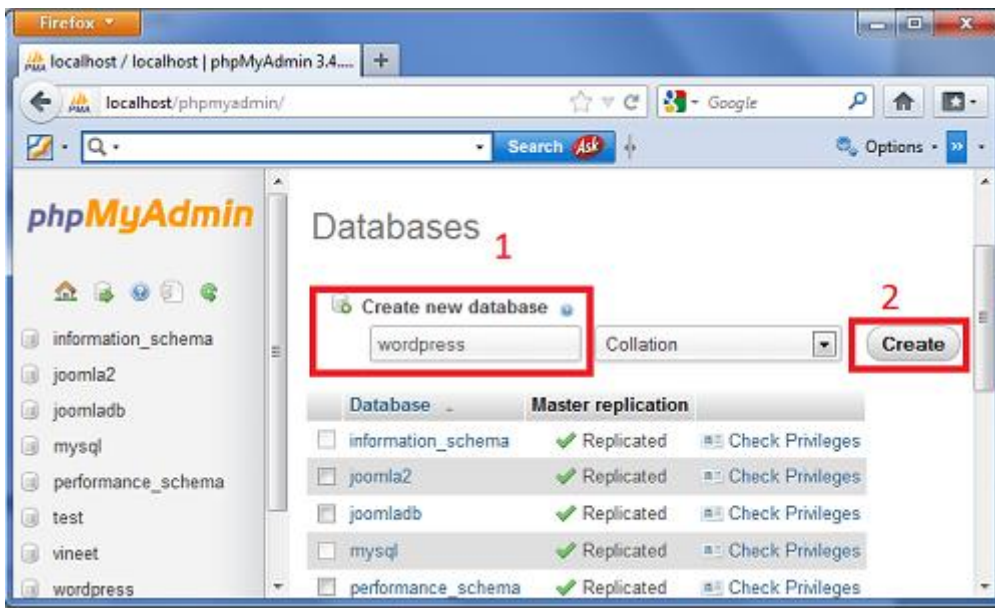


If you want to integrate WordPress into the root of your domain (e.g. <http://example.com/>), move or upload all contents of the unzipped WordPress directory (excluding the WordPress directory itself) into the root directory of your web server.



If you want to have your WordPress installation in its own subdirectory on your website (e.g. <http://example.com/blog/>), create the blog directory on your server and upload the contents of the unzipped WordPress package to the directory.

2: Create a database for WordPress on your web server.



Note: *Collation must be utf8_unicode_ci*

3: Find and rename wp-config-sample.php to wp-config.php, then add your database information in the same.

Enter your database information under the section labeled

// ** MySQL settings - You can get this info from your web host ** //

DB_NAME - The name of the database you created for WordPress in Step 2 .

DB_USER - The username you created for WordPress in Step 2.

DB_PASSWORD - The password you chose for the WordPress username in Step 2.

DB_HOST - The hostname.

```
// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define('DB_NAME', 'rouy_db');

/** MySQL database username */
define('DB_USER', 'rouy_name');

/** MySQL database password */
define('DB_PASSWORD', 'rouy@4321');

/** MySQL hostname */
define('DB_HOST', 'localhost');
```

Enter your secret key values under the section labeled -

* Authentication Unique Keys.

You can generate these using the <https://api.wordpress.org/secret-key/1.1/salt/> WordPress.org secret-key service

If you don't have the HTML editor, don't rename wp-config-sample.php to wp-config.php. You can give this directly by loading <http://yourwebaddrees/wp-admin/setup-config.php> in your web browser.



The image shows the WordPress installation database configuration screen. At the top is the WordPress logo. Below it is a note: "Below you should enter your database connection details. If you're not sure about these, contact your host." There are five input fields with labels and help text:

Field Label	Input Value	Help Text
Database Name	wordpress	The name of the database you want to run WP in.
User Name	username	Your MySQL username
Password	password	...and your MySQL password.
Database Host	localhost	You should be able to get this info from your web host, if localhost does not work.
Table Prefix	wp_	If you want to run multiple WordPress installations in a single database, change this.

At the bottom left is a "Submit" button.

4: Run the WordPress installation script by accessing the URL in a web browser. This should be the URL where you uploaded the WordPress files.

- If you installed WordPress in the root directory, you should visit: <http://example.com/>
- If you installed WordPress in its own subdirectory called blog, for example, you should visit: <http://example.com/blog/>



Welcome

Welcome to the famous five minute WordPress installation process! You may want to browse the [ReadMe documentation](#) at your leisure. Otherwise, just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

Information needed

Please provide the following information. Don't worry, you can always change these settings later.

Site Title

Username

Usenames can have only alphanumeric characters, spaces, underscores, hyphens, periods and the @ symbol.

Password, twice

A password will be automatically generated for you if you leave this blank.

Strength indicator

Hint: The password should be at least seven characters long. To make it stronger, use upper and lower case letters, numbers and symbols like ! " ? \$ % ^ &).

Your E-mail

Double-check your email address before continuing.

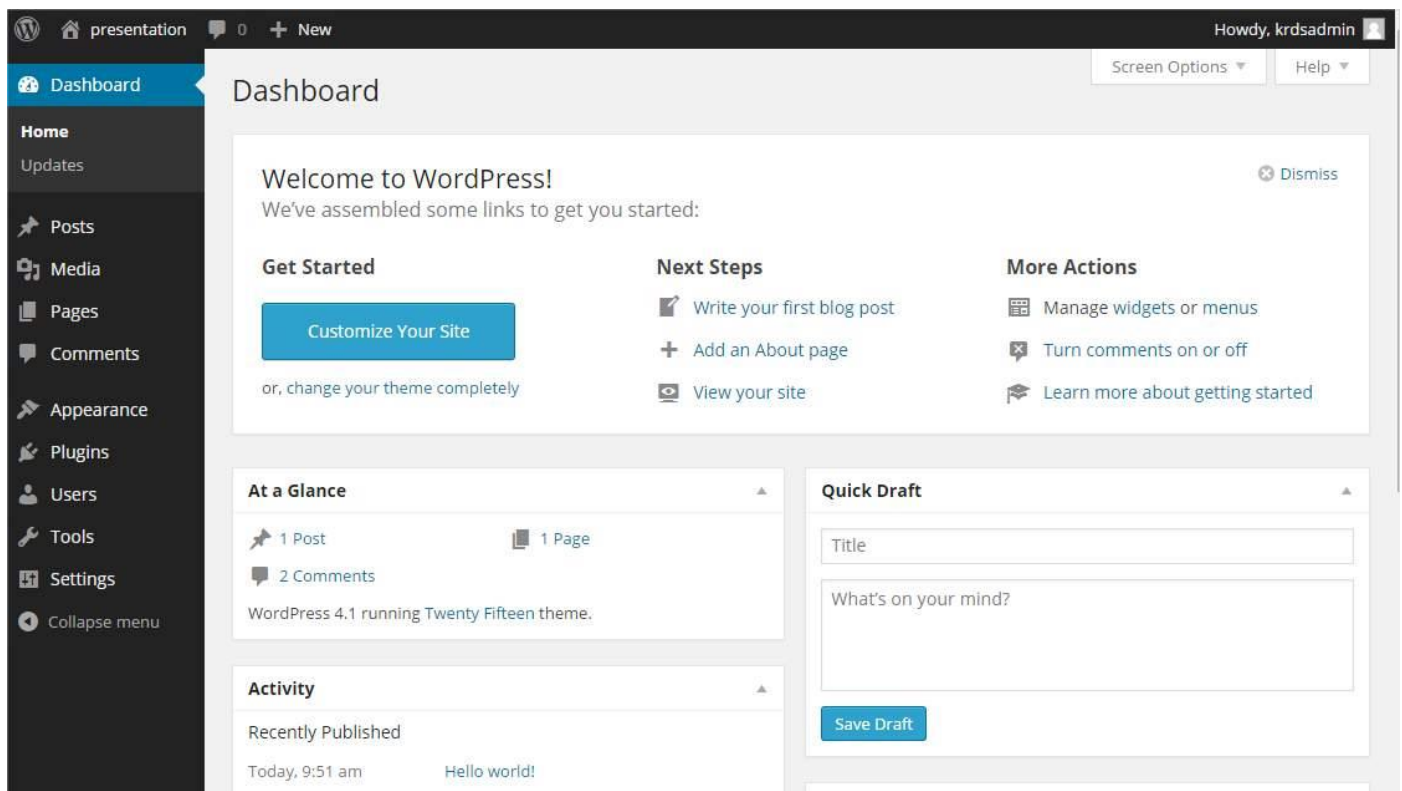
Privacy

☒ Allow my site to appear in search engines like Google and Technorati.

Install WordPress

That's it! WordPress should now be installed.

WordPress - Back-office/Admin area.



Screenshot:

Default

wordpress

admin

page

Wordpress Admin panel is simple yet powerful for both posting your content and online file editing. Its UI is pretty simple with one header, side-menu and content box.

Admin Panel > Header

- **First link with wordpress logo** - You can browse useful links about wordpress like about wordpress, support and feedbacks to forum.
- **Home Icon** - This link will redirect you to the home page of your website.
- **Comment Icon** - This link gives you notifications and moderation alerts for posts.
- **Plus Icon** - You can add posts, pages and media files from this menu
- **Right corener Menu** - You can edit your profile settings.

Admin Panel > Dashboard

The Dashboard tells you about recent activity both at your site and in the WordPress community at large and provide access to updating WordPress, plugins, and themes.

- The Dashboard Screen provides you a number of links to start writing Posts or Pages.
- Statistics and links on the number of posts, pages, Categories and Tags.
- List of the recent comments. Configurable boxes of Incoming Links, and RSS feeds from the

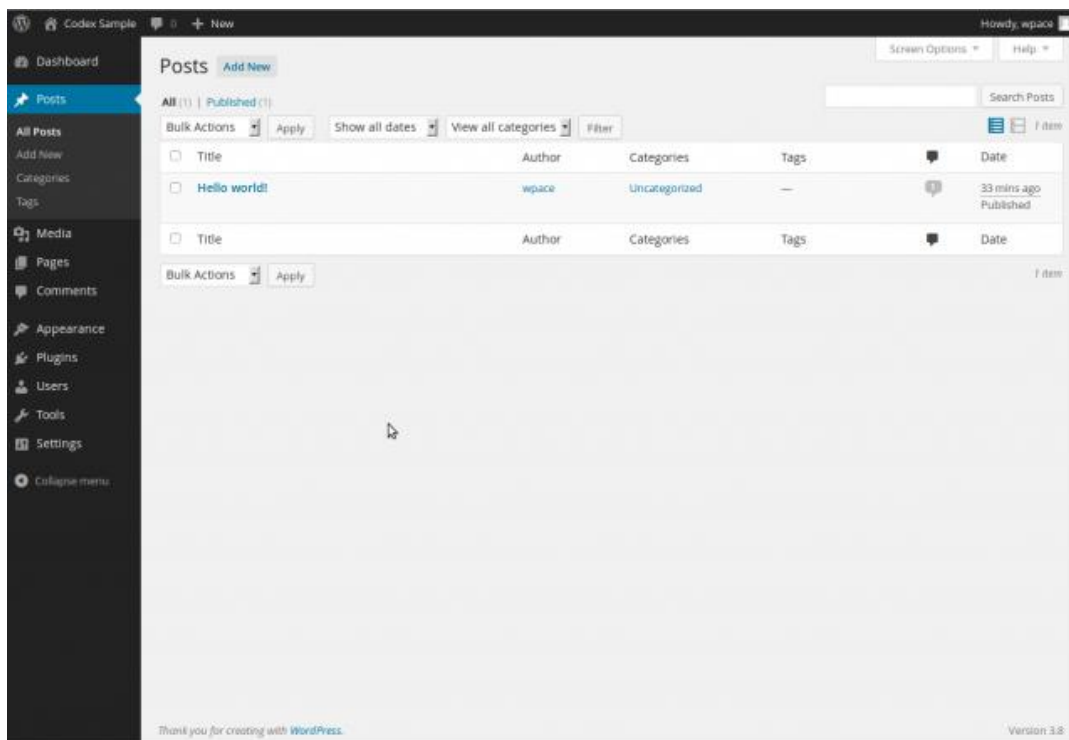
WordPress Blog, the Plugins blog, and Planet WordPress are also displayed.

Posts section in admin Panel

This Screen populate the website with actual information. **Client will spend most of their administration time here.** Posts are the principal element (or content) of a blog. The Posts are the writings, compositions, discussions, discourses, musings, and, yes, the rantings, of a blog owner and contributors. Posts, in most cases, are the reason a blog exists; without Posts, there is no blog!

Posts > All Posts

Via the All Posts Screen you can select the Post or Posts you wish to edit, delete, or view. Multiple Posts can be selected for deletion and for editing.



The table of Posts contains the following columns:

[] – This checkbox, when clicked (checked), 'selects' that particular Post to be processed by a Bulk Action.

Title - This is the Post's Title displayed as a link. Click the Title link to allow this Post to be edited. Next to the Title, if a Post is of a Draft, Private, Pending, Sticky, or Password Protected nature, bold text will display showing that.

"ID" - This is not a column in the Table, but by hovering the mouse over the Post Title, the Post ID is revealed as part of the URL displayed in the browser status bar (in Firefox the status bar is displayed at the bottom of the screen). A Post's ID number is the unique number WordPress' database uses to identify individual Posts.

Author - Displayed in the form of a link, this is the [author](#) who wrote the Post. Clicking the author link causes all the Posts authored by that user to be displayed in the Table of Posts (thus allowing a Bulk Action to be applied to all Posts for a given author).

Categories - Displays all the Categories, in the form of links, to which this Post belongs. Each Category link, when clicked will cause the list of Posts assigned that Category to be displayed in the Table of Posts (note the Category Filter box at the top of the Table will depict the Category being displayed).

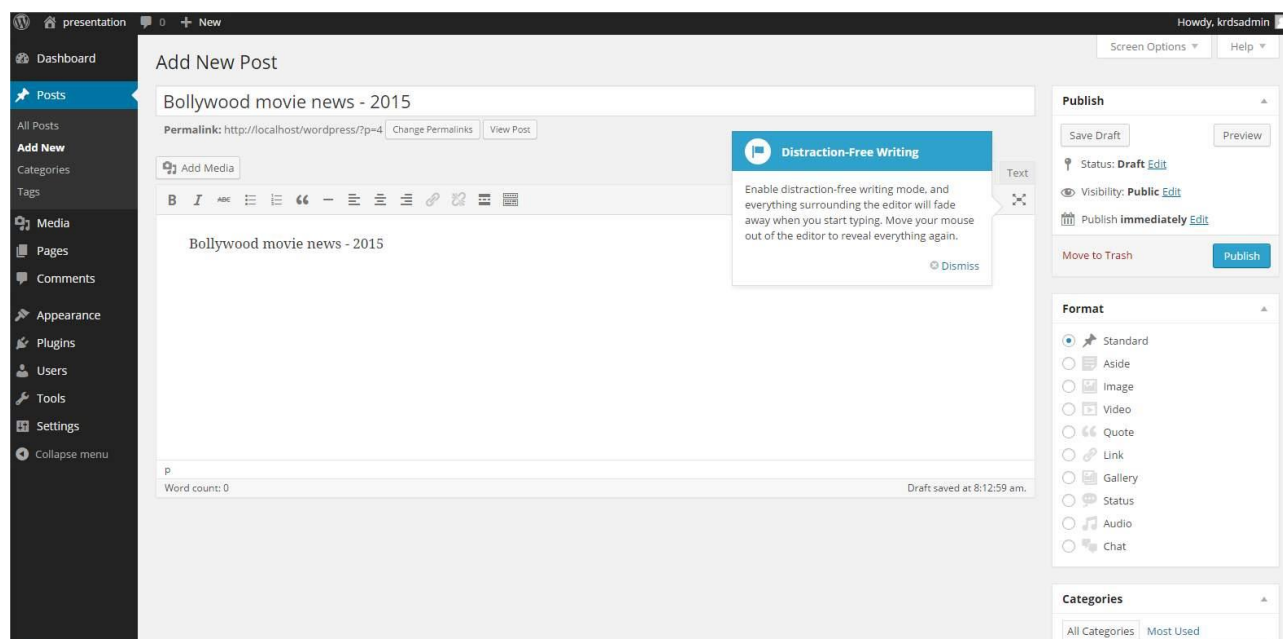
Tags - Displays all the Tags, in the form of links, associated with a Post. Each Tag link, when clicked will cause the list of Posts assigned that Tag to be displayed in the Table of Posts.

Comment bubble - A comment bubble is the column heading, and each Post row has comment bubble with the number of comments for that Post. If a Posts has any comments, then the number comments is displayed in a blue bubble. Clicking on a blue comment bubble causes the Comments Screen to be displayed to allow moderation of those comments.

Date - The Date column for each Post shows the Date 'Published' for Published Posts or the Date 'Last Modified' for other Posts. If the Post is a "future Post," the number of days until that Posts is scheduled for publishing is displayed.

Posts > Add New

The Posts Add New Screen allows you to create new Posts. We have distraction free writing mode to show your post content only. other blocks like header and sidebar will be hidden.



Descriptions of Post Fields

Title/Headline Box

The title of your post. You can use any phrases, words or characters. Avoid using the same title twice as that will cause problems. You can use commas, apostrophes, quotes, hypens/dashes and other typical symbols in the post like "My Site - Here's Lookin' at You, Kid". WordPress will then clean it up to generate a user-friendly and URL-valid name of the post (also called the "post slug") to compose the permalink for the post.

Body Copy Box

The blank box where you enter your writing, links, links to images, and any information you want to display on your site. You can use either the Visual or the Text view to compose your posts. For more on the Text view, see the section below, [Visual Versus Text View](#).

Preview button

Allows you to view the post before officially publishing it.

Publish box

Contains buttons that control the state of your post. The main states are Published, Pending Review and Draft.

Publish box - Visibility

This determines how your post appears to the world. Public posts will be visible by all website visitors once published. Password Protected posts are published to all, but visitors must know the password to view the post content. Private posts are visible only to you (and to other editors or admins within your site)

Publish box - Save

Allows you to save your post as a draft / pending review rather than immediately publishing it. To return to your drafts later, visit Posts - Edit in the menu bar, then select your post from the list.

Publish

Publishes your post on the site. By default, at the time the post is first auto-saved, that will be the date and time of the post within the database.

Format Panel

Post formats replaces the need to use categories to accomplish the similar styling features. Until now, developers hacked WordPress categories to accomplish this effect which ends up creating a lot of clutter and confusion for new bloggers.

Available Post Formats - Aside, chat, gallery, link, image, quote, status and video.

Categories

Every Post in WordPress is filed under one or more Categories. Categories allow the classification of your Posts into groups and subgroups, thereby aiding viewers in the navigation and use of your site.

Featured Image

We can add banner image for the post here.

Revisions

A list of all revisions made to the current post or page. Clicking on a revision will open a dedicated revision change where you can compare the current version of the post or page with any previous versions. There is also an option to restore any previous versions.

Note: You can set basic options for writing, such as the size of the post box, how smiley tags are converted, and other details by going to Administration Panels > Settings > Writing.

Posts > Categories

The Posts Categories Screen allows you to add, edit, and delete Categories, as well as organize your categories hierarchically. Multiple Categories can be selected for deletion. A search option allows you to find the Categories you want to edit or delete. Also remember Categories can be added in the Posts Add New Screen.

Each Category may be assigned to a Category Parent so that you may set up a hierarchy within the category structure.

Name

To reiterate, the Category Name must be unique.

Slug

The Category slug must be unique. The Category Slug is used in the URL. For example, setting a Category Name of "Recipes" and a Category Slug of "food" would show all "Recipes" posts with a URL like example.com/blog/food/.

Parent

It will help to assign a sub-Category. You just have to select the sub-Category's Parent only.

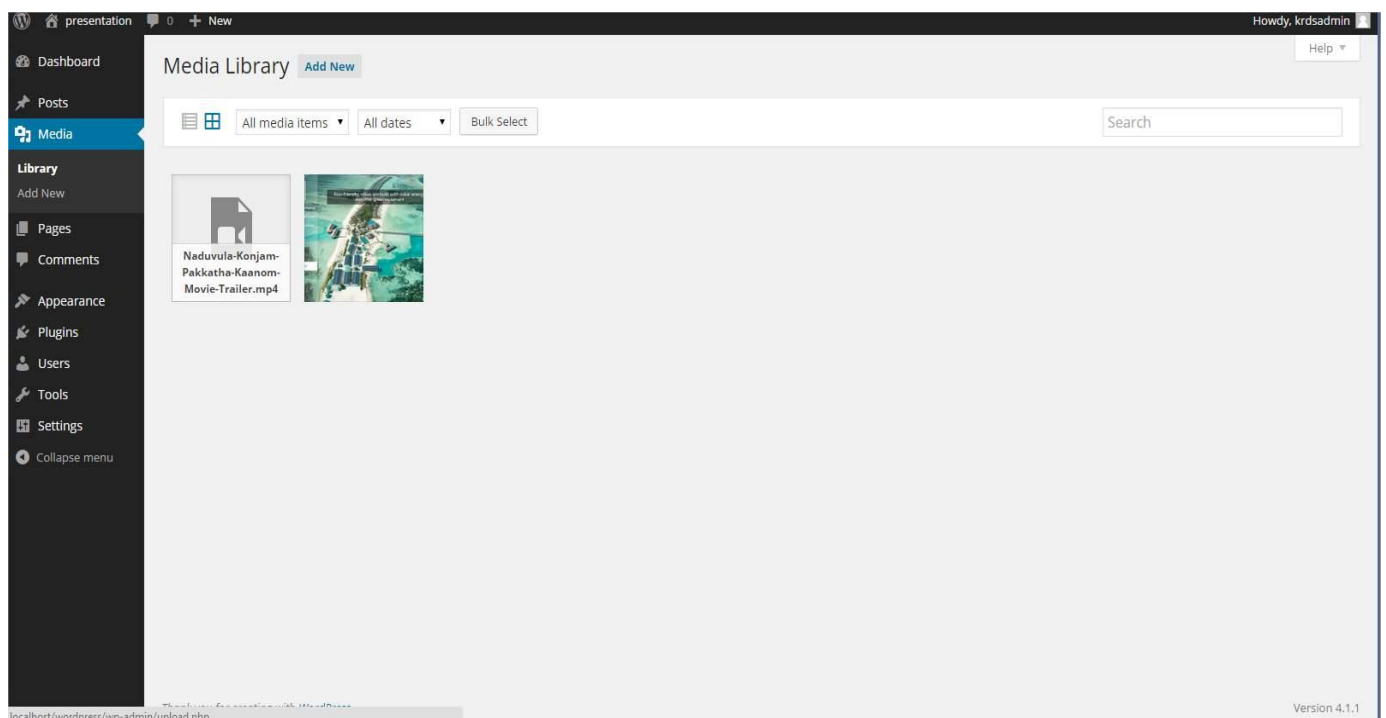
Post > Tags

Tags are the keywords you might assign to each post. Not to be confused with Categories, Tags have no hierarchy, meaning there's no relationship from one Tag to another. But like Categories, Tags provide another means to aid your readers in accessing information on your blog.

The Posts Tags Screen allows you to add, change, or delete Tags. Multiple Tags can be selected for deletion. A search option allows you to find the Tags you want to edit or delete. Also remember Tags can be added in the Posts Add New Screen.

Media Section in Admin Panel

Media is the images, video, recordings, and files, you upload and use in your blog. Media is typically uploaded and inserted into the content when writing a Post or Page. Note that the Uploading Files section in the Settings Media Screen describes the location and structure of the upload directory.



Media > Library

The Media Library Screen allows you edit, delete or view Media previously uploaded to your blog. Multiple Media objects can be selected for deletion. Search and filtering ability is also provided to allow you to find the desired Media.

Media > Add New Media

The Media Add New Screen allows you to upload new media to later use with posts and pages. A Flash Uploader is provided and the ability to use a Browser Uploader is supplied if the Flash Uploader does not work.

Media > Edit Media

The Edit Media page allows you to edit the information (title, caption, alt text, etc.) of media files uploaded to your site. This page does not appear as a link in the main Dashboard navigation, but is reached by clicking on the Edit link in the Media Library Screen that appears when you hover over each item, or when uploading media for the first time on the [Media Add New Screen](#) once your upload has completed.

Pages Section in Admin Panel

A Page is another tool to add content to a WordPress site and is often used to present "static" information about the site; Pages are typically "timeless" in nature. A good example of a Page is the information contained in "About" or "Contact" Pages. A Page should not be confused with the time-oriented objects called Posts, nor should a WordPress Page be confused with the word "page" referring to any web page or HTML document on the Web. Because, Pages live outside of the normal blog chronology, and as such, are not displayed with the rest of your Posts, but are displayed individually.

Pages > All Pages

The All Pages Screen provides the necessary tools to edit, delete or view existing Pages. On this Screen you can select the Page to edit, delete or view. As with Posts, a powerful bulk edit tool allows certain fields to be edited for a whole group of Pages. A handy in-line edit tool, called Quick Edit, allows you to update many fields for an individual Page. Various search and filtering options allow you to find the Pages you want to edit or delete.

Pages > Add New Page

The Add New Page Screen allows you to create new Pages.

Comments Section in Admin Panel

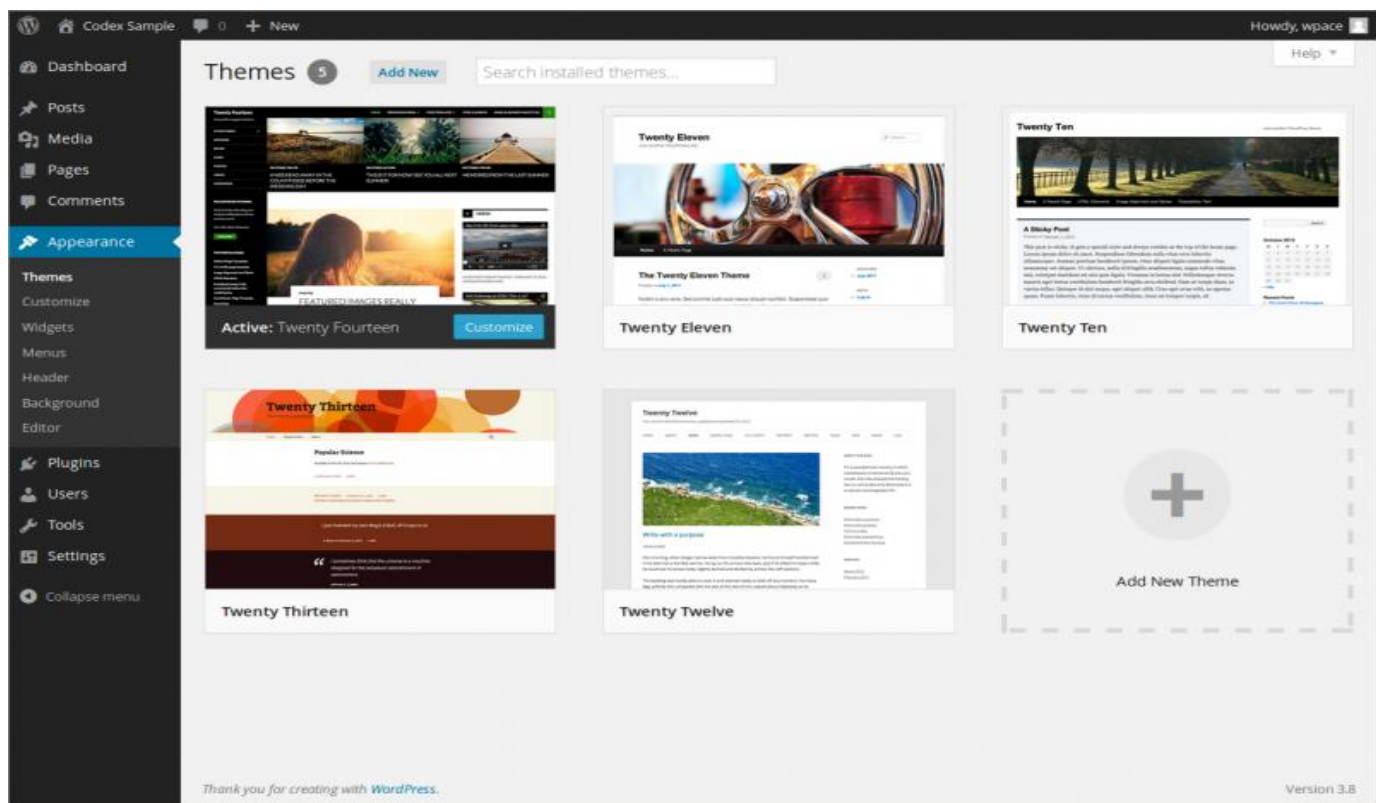
Comments are a feature of blogs which allow readers to respond to Posts. Typically readers simply provide their own thoughts regarding the content of the post, but users may also provide links to other resources, generate discussion, or simply compliment the author for a well-written post.

In the Comments Screen you can edit and delete as well as mark comments as spam. Comments that are awaiting moderation can be marked as approved or previously approved comments can be unapproved. Multiple comments can be selected and approved, marked as spam, unapproved, or deleted. A section at the top of the Comments Screen displays the number of comments awaiting moderation and the number of approved comments. A search box allows you to find specific comments.

Appearance Section in Admin Panel

Appearance > Themes

From the Appearance Themes Screen under the Manage Theme tab you can choose which Theme will be presented to users visiting your site. You can also view screenshots of each Theme you have uploaded to your site. In addition, under the Install Themes tab you can find and install new Themes.



We can use add our own layout by manually adding new folder in themes folder. The root path for themes folder,

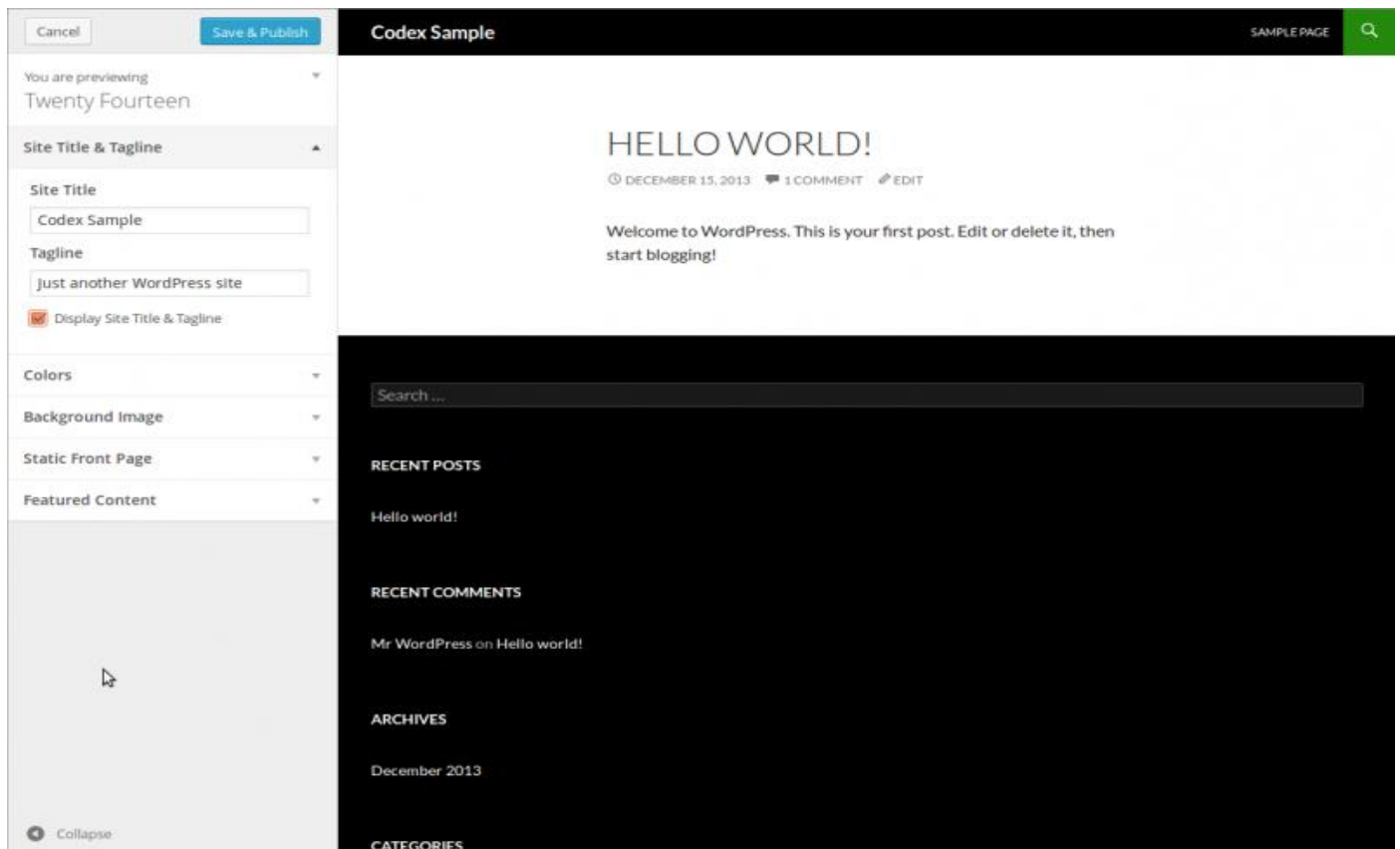
mainFolder -> **wp-content** -> **themes**

You can see the default themes in that folder. you have to copy and paste the any one folder and just edit the available files only. If you likes to add any new pages, you have to add page in admin panel. here you can only edit the common templates.

Appearance > Customize

The Customize Screen displays the settings that can be customized for a specific theme. This option will only be present if the Theme author has configured the theme to allow this capability. For instance, the WordPress Twenty Fifteen theme provides options that allows the user to set the colors and a background image.

Site	Title	&	Tag	Line	:
Site	Title	-	Choose a title and tagline for your site	Text box for site title	
Tag	Line	-	Text box for	tag	line
Display	- Checkbox to enable or hide the display of title and tagline				



Colors:

The color of the title and background as displayed by the theme.

Background Color - Select a color in the same manner as for the title. This background may only be visible on wide displays.

Background

Image:

Select an optional background image to use in place of the solid color specified in Colors above.

No Image - The default state:

Upload New - Either drag/drop a background image file on the provided box or follow the "select a file" link to launch your O/S file selection dialog.

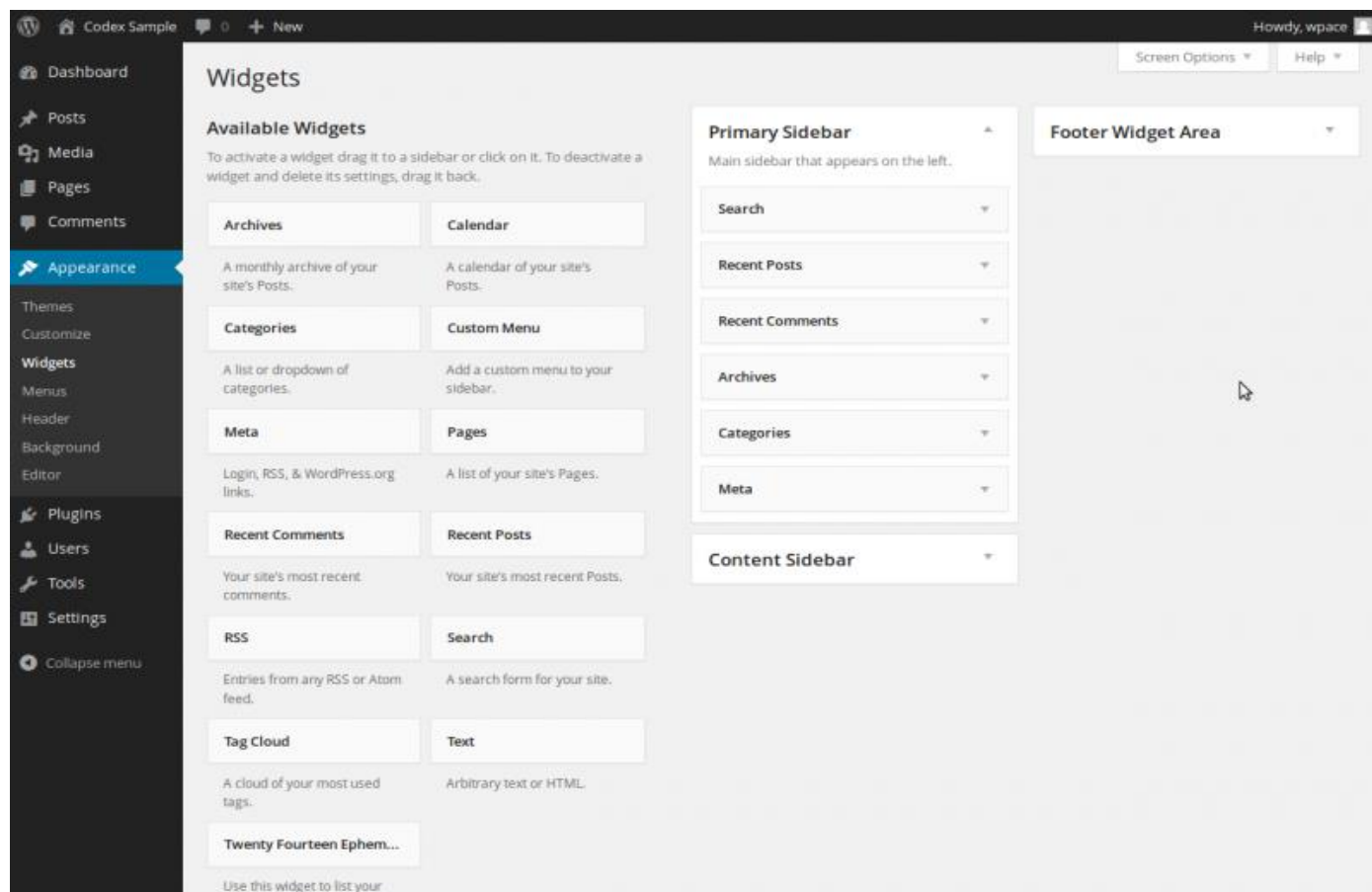
Static Front Page:

Your site's home page can either contain your latest posts or display a static page or post.

Appearance > Widgets

Widgets are gadgets or gizmos that allow you to add various pieces of information to your Theme's sidebar content. From the Appearance Widgets Screen you can add, delete, and configure, the

Widgets use in one or more of your Theme's sidebar.



Configuring Your Widgets

- Add new widgets from the Available Widgets section by dragging them to the Sidebar you want to customize. (There might be more than one sidebar option, depending on your theme.)
- Return to the Widgets Panel to continue adding Widgets to the Sidebar as needed.
- To arrange the Widgets within the Sidebar, drag and drop the widgets in the order you want.
- To customize the Widget features, click the down arrow (or edit link) in the upper right corner to expand the Widget's interface.
- To save the Widget's customization, click Save. (If simply moving a widget to the sidebar without making customizations, you do not need to click the Save button.)
- Preview your site again, and all of your changes should be visible.

Remove a Widget

- Decide which Sidebar from which you want to remove your widget if there is more than one option.

- For the Widget that is to be removed, click the down arrow (or edit link) in the upper right corner to expand the Widget's interface.
- Click the Delete link in the dialog box. The Widget will now be removed permanently.

Note: If you would like to remove a widget temporarily and keep your settings, drag the widget from the Sidebar into the Inactive Widgets section. When you are ready to return the widget, drag it from your Inactive Widgets section back to the Sidebar.

Moving a Widget

- The Widget must be located in a Sidebar or other active content area before it can be moved.
- Click on the Widget Title and hold the mouse button down, then drag the Widget bar up or down to the location desired. At that point release the mouse button. That process is called drag-and-drop.

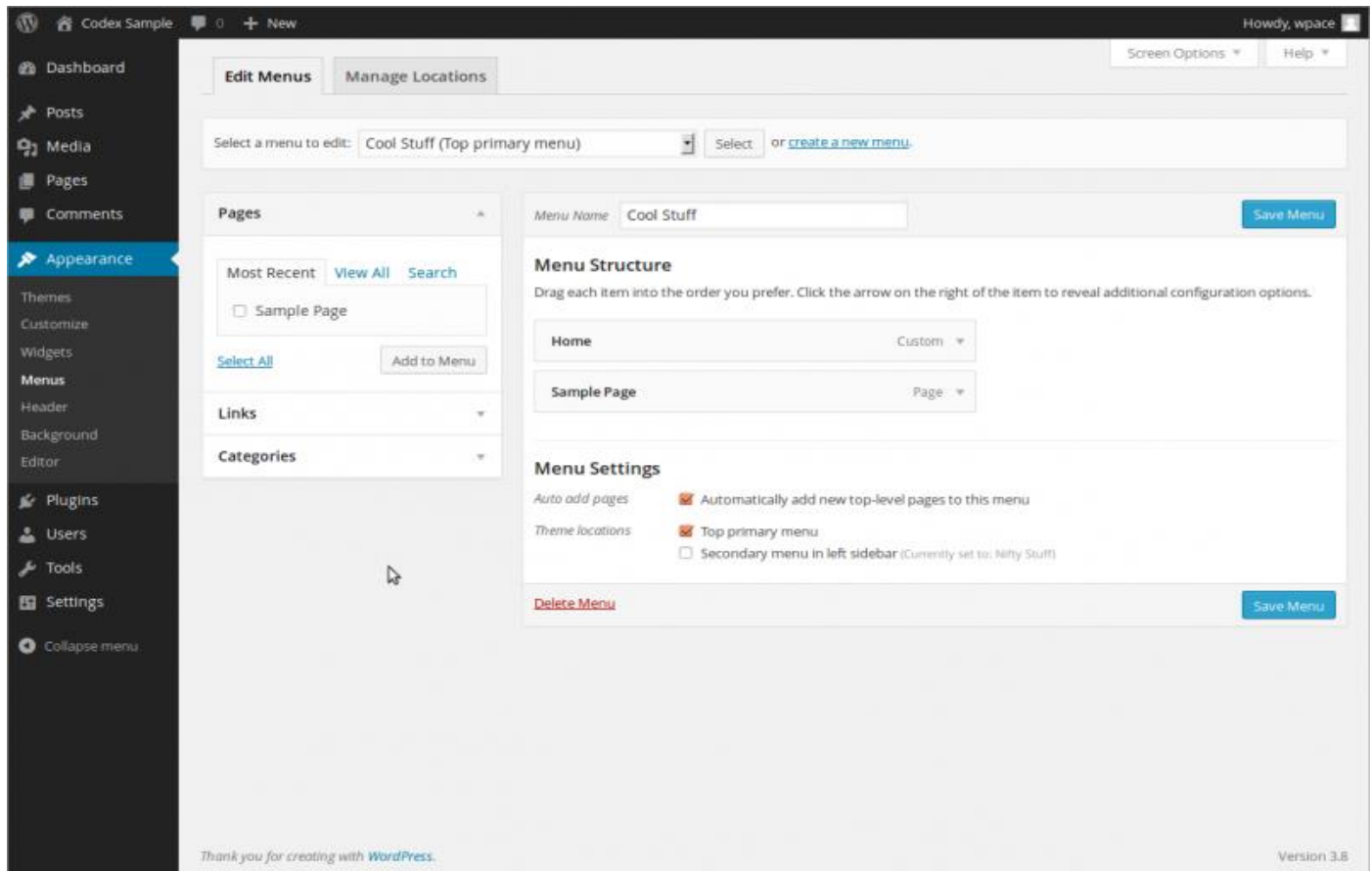
No Sidebars Defined

If the current active Theme is not widget compatible, meaning the theme is not coded for widgets, you will see the message - "You are seeing this message because the theme you are currently using isn't widget-aware, meaning that it has no sidebars that you are able to change."

Appearance > Menus

From the Appearance Menus Screen you can create and edit navigation menus for visitors use. The Menus feature allows you to create a navigation menu of pages, categories, custom links, tags, etc. that is presented to your visitors. A custom menu WordPress Widget allows display of a custom menu in the sidebar or other widgetized areas in the Theme. This option will only be present if the Theme author

has configured the theme to allow this capability and the site administrator uses them.



- To create a new custom menu, click the "create a new menu" link, give the menu a name, and click Save Menu button.
- If more than one menu is supported, you can select where this menu should go using Manage Locations.
- After a menu is created, then detail items can be added to the menu. Choose items like custom links, pages, and categories, from the left column to add to the menu.
- Custom links allow links defined by you that don't fit into the other types of menu items. You need to give an URL and a Menu name.
- You can rearranging the menu by drag and drop.
- You can remove or edit the link by clicking dropdown icon in the right corner of the menu item.
- You can add submenu using [under <menu-item>](#) link in the drop-down panel of the menu item.

Appearance > Header

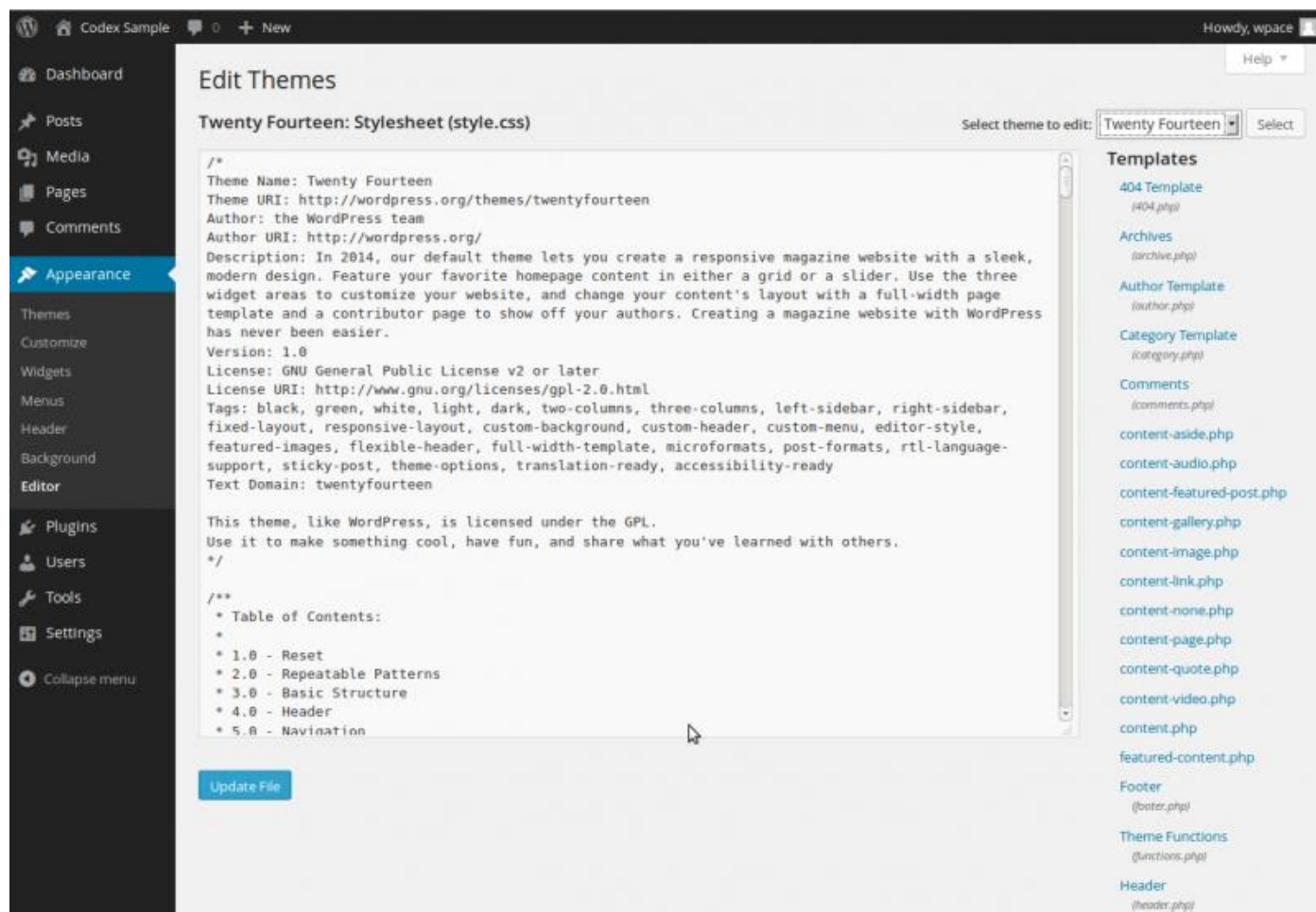
The Header feature allows you to manage what image is displayed in a Theme's header. This option will only be present if the Theme author has configured to header to allow this capability.

Appearance > Background

The Background feature allows you to manage the look and feel of background for your theme. This option will only be present if the Theme author has configured the theme to allow this capability.

Appearance > Editor

Use the Theme Editor to edit the various files that comprise your Themes. The Appearance Editor Screen allows you to designate which theme you want to edit then displays the files in that theme. Each file (Template and CSS) in the theme can be edited in the large text box.



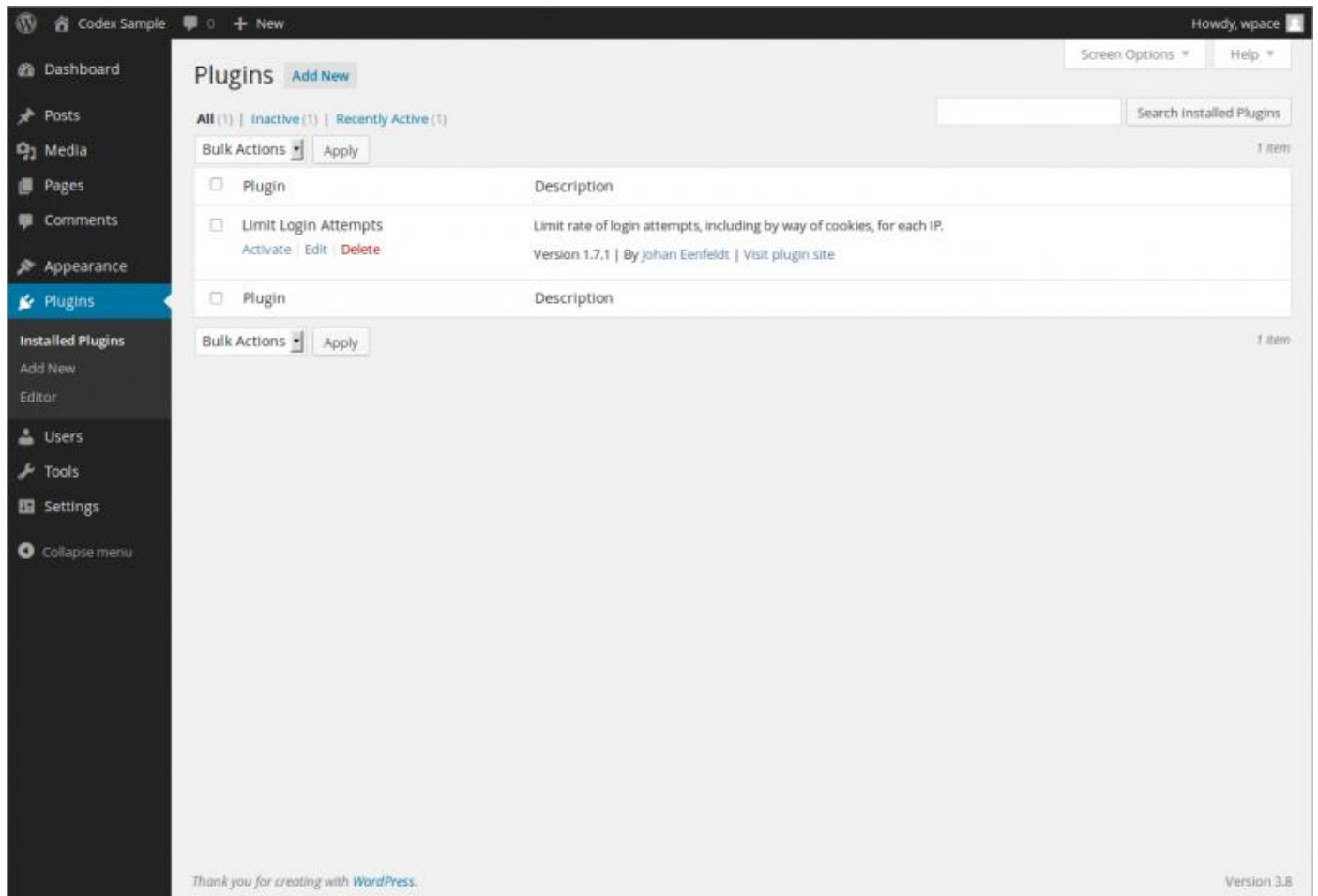
Update

Remember to click this button to save the changes you have made to the file. After clicking this button you should see a splash message at the top of the screen saying File Edited Successfully. If you don't see that message, then your changes are not saved! Note that if a file is not writeable the Update File button will not be available.

File

Plugins Section in Admin Panel

Plugins allow you to add new features to your WordPress blog that don't come standard with the default installation. There are a rich variety of Available Plugins for WordPress, and with the following Screens, plugin installation and management is a snap.



- **Plugins -> Installed plugins** Screen allows you to view the plugins you've downloaded and choose which plugins you want activated on your site.
- **Plugins -> Add New** Screen allows you to add new plugins.
- Using the **Plugins -> Editor** Screen, you can modify the source code of all your plugins.
- You have to click the [activate plugin](#) link to activate the plugin. otherwise its simply added to your build only.

Files related to installed plugins are available in **mainFolder -> wp-content -> plugins**.

Users Section in Admin Panel

Every blog probably has at least two users: admin, the account initially set up by WordPress, and the user account, as the author/owner of the blog, use to write posts. perhaps you want several authors for your blog. If you want a person to be able to post to your blog, that person must have access to a user account; typically, every person will have her or his *own* user account.

- Via the Users option in the main navigation menu you can set up all of the user accounts you need, as well as change user information, or delete users.
- You can also specify your, and others', personal information, such as name, e-mail, etc. from these User Administration Screens.
- All Users - You can manage the accounts of all your site's users at the All Users Screen.
- Add New User - You can create new users with the Users Add New Screen.
- Your Profile - The Users Your Profile Screen allows to change any information related to your user account.

Tools Section in Admin Panel

WordPress Tools provide you the ability to speed up WordPress for your local machine, import content from other sources, export your content, or to upgrade your WordPress software to a new release.

Tools -> Import

WordPress supports the importing data from a number external sources. In many cases, posts, comments, pages, categories, tags, and users, can be imported.

The Tools Import Screen list the software packages that WordPress can import and details what types of data from each of those platforms qualifies for import. Also see Importing Content for a more extensive list of import possibilities.

Tools -> Export

WordPress Export will create an XML file for you to save to your computer. The format, which is called a WordPress eXtended RSS or WXR file, will contain your posts, comments, custom fields, categories, and tags.

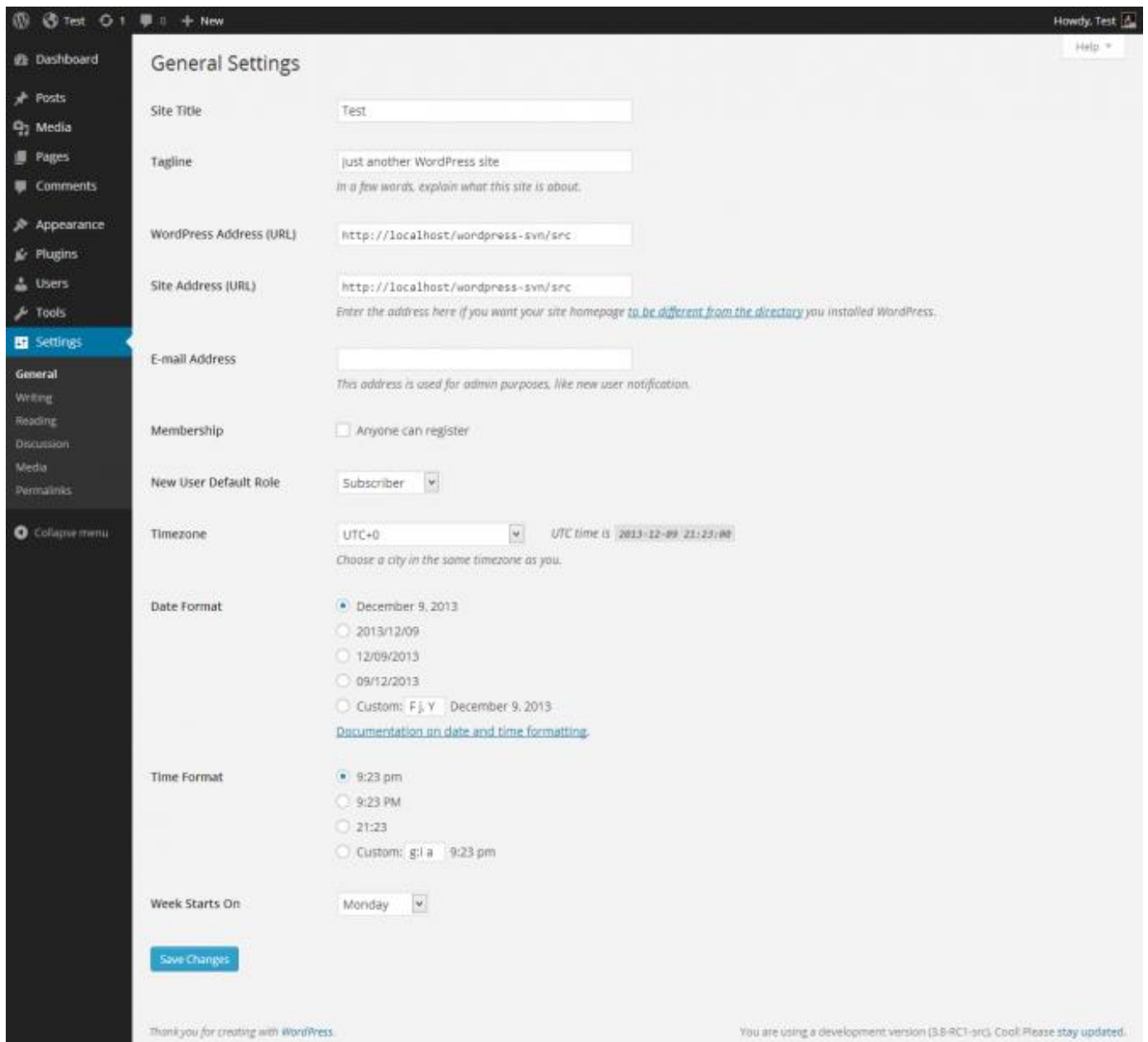
The Tools Export Screen guides you through the easy process of exporting your blog. Take note that the Exporting is a useful method to backup your WordPress data.

Settings Section in Admin Panel

All the settings you've encountered in the other Administration Screens have dealt with very specific parts of your site, or have been of limited scope (only applying to one Category, for example). In the Settings Administration Screen are all of the settings that define your blog as a whole: settings which determine how your site behaves, how you interact with your site, and how the rest of the world interacts with your site.

General

The Settings General Screen is the default Screen in the Settings Administration Screen and controls some of the most basic configuration settings for your site: your site's title and location, who may register an account at your blog, and how dates and times are calculated and displayed.



Writing

Using the Settings Writing Screen, you can control the interface with which you write new posts. These settings control the size of the 'post box' in the Add New Post Screen, the default Category, the default Link Category, the default Post Format, the default image sizes, and the optional Post via e-mail feature.

Reading

You can decide if you want posts, or a "static"Page, displayed as your blog's front (main) page. You can also adjust how many posts are displayed on that main page. In addition, you can adjust syndication feed features to determine how the information from your site is sent to a reader's web browser or other applications.

Discussion

The Settings Discussion Screen allows you to control settings concerning incoming and outgoing comments, pingbacks and trackbacks. You can also control from this Screen the circumstances under which your blog sends you e-mail notifying you about the goings on at your site, and you can decide if your blog should show Avatars and their ratings.

Media

The Settings Media Screen allows you to determine how images, documents, and other media files will be organized when uploaded, and to specify the maximum dimensions in pixels to use when inserting an image into the body of a post.

Media Settings

Image sizes

The sizes listed below determine the maximum dimensions in pixels to use when adding an image to the Media Library.

Thumbnail size	Width	<input type="text" value="150"/>	Height	<input type="text" value="150"/>
<input checked="" type="checkbox"/> Crop thumbnail to exact dimensions (normally thumbnails are proportional)				
Medium size	Max Width	<input type="text" value="300"/>	Max Height	<input type="text" value="300"/>
Large size	Max Width	<input type="text" value="1024"/>	Max Height	<input type="text" value="1024"/>

Uploading Files

☒ Organize my uploads into month- and year-based folders

Permalinks

The Settings Permalinks Screen controls how that custom URL structure is defined. By default, WordPress uses web URLs which have question marks and lots of numbers in them; however, WordPress offers you the ability to create a custom URL structure for your permalinks and archives. This can improve the aesthetics, usability, and forward-compatibility of your links.

Wordpress files

WordPress Main Folder

index.php

Core WordPress index. This is the file that produces the blog output.

license.txt

The WordPress GPL license file.

my-hacks.php

Contains supplemental routines that are processed before output is produced. *This is not included in the default install, but is referenced by the administration page if available.*

readme.html

Pre-install information about WordPress.

wp-atom.php

Produces [Atom](#) syndication output.

wp-blog-header.php

Decides what to display based on the parameters that are passed to the blog. Included from any page that wants to display WordPress content.

wp-comments-post.php

Receives posted comments and adds them to the database.

wp-commentsrss2.php

Produces RSS2 syndication output of post comments.

wp-config-sample.php

A sample configuration file for connecting WordPress to your MySQL database.

wp-config.php

The actual configuration file used to connect WordPress to your MySQL database. This is not included in the default install, but must exist for successful operation of WordPress. You may need to edit this file to modify certain configuration settings.

wp-feed.php

Decides the type of feed to produce based on a request and re-routes to the correct feed-producing file.

wp-links-opml.php

Produces OPML output of Links that were added to the blog via the WordPress admin menu.

wp-login.php

The WordPress login page for registered users.

wp-mail.php

Used for obtaining blog posts that were submitted via email. The URL of this file is usually added to a cron job so that it is regularly retrieved, so that new email posts are accepted.

wp-pass.php

Accepts the password needed to view password-protected posts, then redirects back to the protected post.

wp-rdf.php

Produces RDF syndication output.

wp-register.php

Allows new users to register usernames in WordPress via an online form.

wp-rss.php

Produces RSS syndication output.

wp-rss2.php

Produces RSS2 syndication output.

wp-settings.php

Performs various pre-execution routines, including checking for correct installation, including auxiliary functions, applying user plugins, initializing execution timers, etc.

wp-trackback.php

Handles incoming trackback requests.

wp.php

A very simple template for displaying blog posts. Does not include a lot of the fancy but fragile features of index.php.

xmlrpc.php

Handles incoming xmlrpc commands. Among other things, this allows posting without using the built-in web-based administrative interface.

WP-Admin Folder

wp-admin/admin.php

The core of the admin files. Connects to the database, integrates the dynamic menu data, displays non-core console (dashboard) pages, etc.

wp-admin/admin-footer.php

Contains the bottom of the admin console. Included from each of the console pages.

wp-admin/admin-functions.php

Contains various functions used by the admin console.

wp-admin/admin-header.php

Contains the top half of the admin console. Includes menu-header.php for menu logic.

wp-admin/bookmarklet.php

The pop-up page that is displayed when using the bookmarklet. Uses the default edit-form.php page for post input.

wp-admin/categories.php

Admin console page for category management.

wp-admin/edit.php

Admin console for post management.

wp-admin/edit-comments.php

Admin console page for comment management.

wp-admin/edit-form-advanced.php

Admin console page for the advanced post editing form. Included from post.php.

wp-admin/edit-form.php

Admin console page for the simple post editing form. Included from post.php.

wp-admin/edit-form-comment.php

Edits a specific post comment.

wp-admin/edit-page-form.php

Admin console page for editing pages. Included from post.php and page-new.php.

wp-admin/edit-pages.php

Admin console for page management.

wp-admin/index.php

The administrative default page. Redirects to an appropriate page based on user access.

wp-admin/install-helper.php

A set of functions for database maintenance, including the popular-in-plugins `maybe_create_table()` and `maybe_add_column()`.

wp-admin/install.php

Installs WordPress.

wp-admin/link-add.php

Admin console page to add links.

wp-admin/link-categories.php

Admin console page for link category management.

wp-admin/link-import.php

Admin console page used to import links.

wp-admin/link-manager.php

Admin console page for link management.

wp-admin/link-parse-opml.php

Contains functions used to parse an OPML file when importing links.

wp-admin/menu-header.php

The file used to display the menu in the admin interface.

wp-admin/menu.php

The default admin menu core functions.

wp-admin/moderation.php

Contains functions related to comments moderation.

wp-admin/options.php

Admin console file that can be used to change *every* settings in one update.

wp-admin/options-discussion.php

Admin console page to manage options regarding comments and trackbacks.

wp-admin/options-general.php

Admin console page to manage options regarding basic configuration settings.

wp-admin/options-misc.php

Admin console page to manage options regarding file uploads, link tracking and support for custom "hacks".

wp-admin/options-permalink.php

Admin console page to manage options regarding permalinks.

wp-admin/options-reading.php

Admin console page to manage options regarding how the information from your site is sent to a reader's web browser or other application.

wp-admin/options-writing.php

Admin console page to manage options regarding the interface with which you write new posts.

wp-admin/page-new.php

Admin console page to create a new page.

wp-admin/plugin-editor.php

Admin console page to edit a plugin file.

wp-admin/plugins.php

Admin console page to manage plugins.

wp-admin/post.php

Admin console page to create a new post.

wp-admin/profile.php

Admin console page to manage your profile.

wp-admin/setup-config.php

Admin console page to create the *wp-config.php* file for the first time.

wp-admin/sidebar.php

wp-admin/templates.php

Admin console page to edit any server-writeable file.

wp-admin/theme-editor.php

Admin console page to edit any file within a specific theme.

wp-admin/themes.php

Admin console page to manage themes.

wp-admin/upgrade-functions.php

Contains functions related to version upgrading.

wp-admin/upgrade-schema.php

Contains default MySQL tables structure and default options, used when upgrading.

wp-admin/upgrade.php

Admin console page to upgrade from a lower version to a higher one.

wp-admin/user-edit.php

Admin console page to edit a user.

wp-admin/users.php

Admin console page to manage users.

wp-admin/wp-admin.css

The default stylesheet for the administrative console.

wp-admin/images

This directory holds images associated and used by WordPress in the Administration Panels.

wp-admin/import

All files here are used for importing posts from the relevant department. Mostly as same as page title.(Ex. blogger.php used for importing posts from Blogger.)

wp-content

This directory is typically not upgraded. The /wp-content/ directory is for user-supplied content. WordPress upgrades are supposed to specifically avoid doing anything to any of the contents of this directory, unless the user wants the latest version of the default WordPress Themes.

WordPress Themes and WordPress Plugins are stored in this directory.

wp-content/plugins

All WordPress plugins are placed in this directory. The default plugin shipped with WordPress is an example plugin mostly for plugin developers, the Hello Dolly plugin, which displays random lines of the song "Hello Dolly." Current versions also include the Akismet comment spam fighting WordPress Plugin as an option.

wp-content/themes

All WordPress Theme data is placed within this directory in its own folder such as example.com/wp-content/themes/themedirectory/.

All the pages here are important.

wp-content/themes/themedir

All files associated with a WordPress Theme are stored in their own directory under the wp-content/themes/themedir directory.

themedir/header.php

Used to manage the header of every page.

themedir/index.php

Used to manage how the posts are displayed in your homepage.

themedir/footer.php

Used to manage the footer of the pages.

themedir/page.php

This is the template that displays all pages by default.

themedir/single.php

This is the template that displays the detailed view of posts..

themedir/sidebar.php

Used to manage the sidebar.

themedir/comments.php

Used to manage how the comments are displayed.

themedir/search.php

Used to display a search form to search for entries.

themedir/function.php

Used to manage the additional functions to the theme.

themedir/content-none.php

Used to manage the content-none page.

themedir/404.php

Used to manage the 404 Error page.

themedir/style.css

The main CSS file of WordPress.

themedir/images

WordPress Themes store images in a subdirectory under their Theme folder. For example, images used by the default theme are in *wp-content/themes/default/images/*.

themedir/js

This folder manages all the scripts related to that particular theme.

wp-includes

wp-includes contains everything needed to run WordPress via the frontend (and then some). It is the territory of WordPress Core, and as the adage goes, never modify core under any circumstances.

While wp-content may define plugins and themes, the API itself and the vast majority of the WordPress core sits in that folder. These are depended on by almost the entirety of WordPress. All your standard functions and their dependencies will be found in this folder.

wp-admin may define additional items but it is not as vital to the running of the frontend.

wp-includes/classes.php

Contains basic classes including the core post-fetching mechanism, WP_Query, and the rewrite manger, WP_Rewrite.

wp-includes/class-pop3.php

Contains a support class for accessing POP mailboxes. Used by wp-mail.php

wp-includes/class-snoopy.php

[Snoopy](#) is a PHP class that simulates a web browser. It automates the task of retrieving web page content and posting forms, for example.

wp-includes/functions-compat.php

Contains support functions that are present in newer PHP versions that are used only when running on older PHP versions.

wp-includes/functions-formatting.php

Contains support functions for cleaning XHTML and formatting text correctly in certain character sets.

wp-includes/functions-post.php

Contains support functions for managing posts in the database, querying user capabilities, fetching and writing comments, etc.

wp-includes/functions.php

Contains numerous vital support functions. Largest file in WordPress -- almost double the next largest file.

wp-includes/gettext.php

Part of the PHP-gettext GPL translation library.

wp-includes/kses.php

Contains functions that are used to render and filter HTML in posts or comments.

wp-includes/links.php

Contains functions that are used to manage and use the *Links* feature of WordPress.

wp-includes/template-functions-author.php

Contains theme functions related to the author of a post or a comment.

wp-includes/template-functions-category.php

Contains theme functions related to categories.

wp-includes/template-functions-general.php

Contains general theme functions.

wp-includes/template-functions-links.php

Contains theme functions related to links

wp-includes/template-functions-post.php

Contains theme functions related to posts.

wp-includes/template-functions.php

Contains all the "template-" files above.

wp-includes/vars.php

Used to set miscellaneous variables.

wp-includes/version.php

Used to set the currently used WordPress version.

wp-includes/wp-db.php

Contains functions used to connect to the MySQL database.

Adding Logo Image

By default, WordPress theme might include a blog title link in the header, but to customize and brand the site we want to replace it with a logo.

We can add this in **header.php** (*Root: mainFolder > wp-content > yourThemedir*). Open the file in text editor, you can find the below line,

```
<h1>
<a href="<?php echo get_option('home'); ?>/"><?php bloginfo('name'); ?></a>
</h1>
```

Replace the text with your logo image(you have to upload the image in theme directory) as below,

```
<h1>
<a href="<?php echo get_option('home'); ?>/">
  " />
</a>
</h1>
```

Adding a Favicon

1. Upload the favicon.ico file into your current theme's main folder.
(Most common root: **mainFolder > wp-content > yourThemedir**)
2. Edit the your theme's header.php. You can edit it in two ways.
 - a. Via Admin: **Admin Panel > Appearance > Theme Editor > header.php**
 - b. Via Text Editor: **mainFolder > wp-content > yourThemedir > header.php**
3. Add the below line exactly next to title tag.
<link rel="shortcut icon" href="<?php echo get_stylesheet_directory_uri(); ?>/favicon.ico" />

```

1
2
3 <!DOCTYPE html>
4 <html lang="en-US">
5 <head>
6   <meta charset="UTF-8">
7   <meta name="viewport" content="width=device-width, initial-scale=1">
8   <title>presentation | Just another WordPress site</title>
9   <link rel="shortcut icon" href="http://localhost/wordpress/wp-content/themes/pure-simple/favicon.ico" />
10  <link rel="profile" href="http://gmpg.org/xfn/11">
11  <link rel="pingback" href="http://localhost/wordpress/xmlrpc.php">
12  <link rel="alternate" type="application/rss+xml" title="presentation &raquo; Feed" href="http://localhost/wordpress/?feed=rss2">
13  <link rel="alternate" type="application/rss+xml" title="presentation &raquo; Comments Feed" href="http://localhost/wordpress/?feed=comments-rss2">
14  <link rel="stylesheet" id="open-sans-css" href="//fonts.googleapis.com/css?family=Open+Sans%3A300italic%2C400italic%2C600italic%2C700italic%2C800italic%2C900italic%2C&ver=4.1.1">
15  <link rel="stylesheet" id="dashicons-css" href="http://localhost/wordpress/wp-includes/css/dashicons.min.css?ver=4.1.1">
16  <link rel="stylesheet" id="admin-bar-css" href="http://localhost/wordpress/wp-includes/css/admin-bar.min.css?ver=4.1.1">
17  <link rel="stylesheet" id="puresimple-responsive-css" href="http://localhost/wordpress/wp-content/themes/pure-simple/css/responsive.css?ver=4.1.1">
18  <link rel="stylesheet" id="puresimple-fontawesome-css" href="http://localhost/wordpress/wp-content/themes/pure-simple/css/fontawesome.css?ver=4.1.1">
19  <link rel="stylesheet" id="puresimple-opensans-css" href="http://localhost/wordpress/wp-content/themes/pure-simple/css/opensans.css?ver=4.1.1">
20  <link rel="stylesheet" id="puresimple-style-css" href="http://localhost/wordpress/wp-content/themes/pure-simple/style.css?ver=4.1.1">
21  <script type="text/javascript" src="http://localhost/wordpress/wp-includes/js/jquery/jquery.js?ver=1.11.1"></script>
22  <script type="text/javascript" src="http://localhost/wordpress/wp-includes/js/jquery/jquery-migrate.min.js?ver=1.2.1"></script>
23  <link rel="EditURI" type="application/rsd+xml" title="RSD" href="http://localhost/wordpress/xmlrpc.php?rsd" />
24  <link rel="wlwmanifest" type="application/wlwmanifest+xml" href="http://localhost/wordpress/wp-includes/wlwmanifest.xml" />
25  <meta name="generator" content="WordPress 4.1.1" />

```

Wordpress Menu

We shouldn't edit the menu items, as it is in the core build. (Root: **wp-includes** > **nav-menu.php**)

Menu layout coded in **ul > li > a** format. Any classes related to the link are mentioned in the **** tag, not in the **a** tag. Important classes are,

- menu-item - Default class
- current-menu-item - Active class
- menu-item-has-children - List with Sub Menu
- menu-item-object-category - Category List Menu
- menu-item-object-page - Page Menu
- menu-item-{id} - Unique class for each menu.

By default the layout looks like below.

```

▼ <nav id="primary-navigation" class="site-navigation primary-navigation" role="navigation">
  ▶ <div class="toggle-container visible-xs visible-sm hidden-md hidden-lg" style="background-color: #000000;">...</div>
  <a class="screen-reader-text skip-link" href="#content">Skip to content</a>
  ▼ <div class="menu-mainmenu-container">
    ▼ <ul id="menu-mainmenu" class="nav-menu">
      ▶ <li id="menu-item-15" class="menu-item menu-item-type-custom menu-item-object-custom menu-item-home menu-item-15">...</li>
      ▶ <li id="menu-item-13" class="menu-item menu-item-type-post_type menu-item-object-page menu-item-13">...</li>
      ▶ <li id="menu-item-14" class="menu-item menu-item-type-post_type menu-item-object-page current-post-parent menu-item-14">...</li>
      ▶ <li id="menu-item-16" class="menu-item menu-item-type-taxonomy menu-item-object-category menu-item-16">...</li>
      ▼ <li id="menu-item-17" class="menu-item menu-item-type-taxonomy menu-item-object-category current-menu-parent menu-item-has-children menu-item-17">
        <a href="http://localhost/wordpress/category/movies/">Movies</a>
        ▶ <ul class="sub-menu">...</ul>
      </li>
    </ul>
  </div>
</nav>

```

But we can modify the menu style by adding custom menu plugin and edit it like as we wanted.

- First we need to add and activate the plugin in admin.
- Then, we can edit the menu style in the plugin files. (Root: **MainFolder** > **wp-content** > **Plugins** > **YourPlugin**)

Including stylesheet / javascript files

1. Upload your css/js file in your theme directory,
 - a. CSS folder: **mainFolder > wp-content > yourThemedir > css**
 - b. JS folder: **mainFolder > wp-content > yourThemedir > js**
2. To include js in the particular page, open the page in the admin editor or in the text editor and add the below line.
<script src='<?php echo get_template_directory_uri()?>/js/filename.js?ver=1'></script>
3. To include css in the particular page, open the page in the admin editor or in the text editor and add the below line.
<link href="<?php echo get_template_directory_uri()?>/css/filename.css?ver=1" rel="stylesheet" />



```
<?php
2  /**
3   * The template for displaying Restaurants
4   */
5  get_header();
6  ?>
7  <?php if(wpmu_is_phone()) { ?>
8      <script type='text/javascript' src='<?php echo get_template_directory_uri()?>/js/slick.min.js?ver=1'></script>
9      <link rel="stylesheet" href="<?php echo get_template_directory_uri()?>/css/slick.css?ver=1" />
10
11  <script>
12      jQuery(document).ready(function($)
13      {
14          jQuery('.foodStreetList').slick({
15              dots: false,
16              infinite: true,
17              speed: 300,
18              slidesToShow: 2,
19              slidesToScroll: 1,
20              focusOnSelect: true,
21              centerMode: true,
22              arrows: true,
23              variableWidth: true
24          });
25      });
26  </script>
27  <?php } ?>
```

How to show breadcrumbs

Currently most of the themes won't come with breadcrumbs. We need to add plugins to generate breadcrumbs with wordpress. As we seen **WordPress SEO by Yoast** is the important plugin for wordpress, its come with most options including breadcrumb.

After activating the plugin, Goto **SEO > Advanced Page** in admin panel. You can activate breadcrumb in **Breadcrumbs tab**.

Advanced - WordPress SEO by Yoast

Settings saved.

Breadcrumbs | Permalinks | RSS

☒ Enable Breadcrumbs

Separator between breadcrumbs: »

Anchor text for the Homepage: Home

Prefix for the breadcrumb path:

Prefix for Archive breadcrumbs: Archives for

Prefix for Search Page breadcrumbs: You searched for

Breadcrumb for 404 Page: Error 404: Page not found

☒ Bold the last page in the breadcrumb

Taxonomy to show in breadcrumbs for:

Posts: None

Then, you can insert the below breadcrumb code in your header template or wherever you want.

```
<?php if ( function_exists('yoast_breadcrumb') ) {
yoast_breadcrumb('<p id="breadcrumbs">','</p>');
} ?>
```

If you likes to add your own code in breadcrumb, you can follow the below instructions.

Add the below code in your theme functions.php

```
<?php
function the_breadcrumb () {

    // Settings
    $separator = '&gt;';
    $id = 'breadcrumbs';
    $class = 'breadcrumbs';
    $home_title = 'Homepage';

    // Get the query & post information
    global $post, $wp_query;
    $category = get_the_category();

    // Build the breadcrumbs
    echo '<ul id="' . $id . '" class="' . $class . '">';

    // Do not display on the homepage
    if ( !is_front_page() ) {
        // Home page
```

```

    echo '<li class="item-home"><a class="bread-link bread-home" href="' .
get_home_url() . '" title="' . $home_title . '">' . $home_title .
'</a></li>';
    echo '<li class="separator separator-home"> ' . $separator . ' </li>';

    if ( is_single() ) {
        // Single post (Only display the first category)
        echo '<li class="item-cat item-cat-' . $category[0]->term_id . '
item-cat-' . $category[0]->category_nicename . '"><a class="bread-cat bread-
cat-' . $category[0]->term_id . ' bread-cat-' . $category[0]-
>category_nicename . '" href="' . get_category_link($category[0]->term_id ) .
'" title="' . $category[0]->cat_name . '">' . $category[0]->cat_name .
'</a></li>';
        echo '<li class="separator separator-' . $category[0]->term_id .
'"> ' . $separator . ' </li>';
        echo '<li class="item-current item-' . $post->ID . '"><strong
class="bread-current bread-' . $post->ID . '" title="' . get_the_title() .
'">' . get_the_title() . '</strong></li>';
    } else if ( is_category() ) {
        // Category page
        echo '<li class="item-current item-cat-' . $category[0]->term_id .
' item-cat-' . $category[0]->category_nicename . '"><strong class="bread-
current bread-cat-' . $category[0]->term_id . ' bread-cat-' . $category[0]-
>category_nicename . '">' . $category[0]->cat_name . '</strong></li>';
    } else if ( is_page() ) {
        // Standard page
        if( $post->post_parent ){
            // If child page, get parents
            $anc = get_post_ancestors( $post->ID );
            // Get parents in the right order
            $anc = array_reverse($anc);
            // Parent page loop
            foreach ( $anc as $ancestor ) {
                $parents .= '<li class="item-parent item-parent-' .
$ancestor . '"><a class="bread-parent bread-parent-' . $ancestor . '" href="'
. get_permalink($ancestor) . '" title="' . get_the_title($ancestor) . '">' .
get_the_title($ancestor) . '</a></li>';
                $parents .= '<li class="separator separator-' . $ancestor
. '"> ' . $separator . ' </li>';
            }
            // Display parent pages
            echo $parents;
            // Current page
            echo '<li class="item-current item-' . $post->ID . '"><strong
title="' . get_the_title() . '"> ' . get_the_title() . '</strong></li>';
        } else {
            // Just display current page if not parents
            echo '<li class="item-current item-' . $post->ID . '"><strong
class="bread-current bread-' . $post->ID . '"> ' . get_the_title() .
'</strong></li>';
        }
    }
}

```

```

} else if ( is_tag() ) {
    // Get tag information
    $term_id = get_query_var('tag_id');
    $taxonomy = 'post_tag';
    $args = 'include=' . $term_id;
    $terms = get_terms( $taxonomy, $args );

    // Display the tag name
    echo '<li class="item-current item-tag-' . $terms[0]->term_id . '
item-tag-' . $terms[0]->slug . '"><strong class="bread-current bread-tag-' .
$terms[0]->term_id . ' bread-tag-' . $terms[0]->slug . '">' . $terms[0]->name
. '</strong></li>';
} elseif ( is_day() ) {

    // Year link
    echo '<li class="item-year item-year-' . get_the_time('Y') .
'"><a class="bread-year bread-year-' . get_the_time('Y') . '" href="' .
get_year_link( get_the_time('Y') ) . '" title="' . get_the_time('Y') . '">' .
get_the_time('Y') . ' Archives</a></li>';
    echo '<li class="separator separator-' . get_the_time('Y') . '">
' . $separator . ' </li>';

    // Month link
    echo '<li class="item-month item-month-' . get_the_time('m') .
'"><a class="bread-month bread-month-' . get_the_time('m') . '" href="' .
get_month_link( get_the_time('Y'), get_the_time('m') ) . '" title="' .
get_the_time('M') . '">' . get_the_time('M') . ' Archives</a></li>';
    echo '<li class="separator separator-' . get_the_time('m') . '">
' . $separator . ' </li>';

    // Day display
    echo '<li class="item-current item-' . get_the_time('j') .
'"><strong class="bread-current bread-' . get_the_time('j') . '">' .
get_the_time('jS') . ' ' . get_the_time('M') . ' Archives</strong></li>';
} else if ( is_month() ) {

    // Year link
    echo '<li class="item-year item-year-' . get_the_time('Y') .
'"><a class="bread-year bread-year-' . get_the_time('Y') . '" href="' .
get_year_link( get_the_time('Y') ) . '" title="' . get_the_time('Y') . '">' .
get_the_time('Y') . ' Archives</a></li>';
    echo '<li class="separator separator-' . get_the_time('Y') . '">
' . $separator . ' </li>';

    // Month display
    echo '<li class="item-month item-month-' . get_the_time('m') .
'"><strong class="bread-month bread-month-' . get_the_time('m') . '" title="' .
get_the_time('M') . '">' . get_the_time('M') . ' Archives</strong></li>';
} else if ( is_year() ) {

    // Display year archive

```



```

        echo '<li class="item-current item-current-' . get_the_time('Y')
. '"><strong class="bread-current bread-current-' . get_the_time('Y') . '"
title="' . get_the_time('Y') . '">' . get_the_time('Y') . '
Archives</strong></li>';
    } else if ( is_author() ) {

        // Get the author information
        global $author;
        $userdata = get_userdata( $author );

        // Display author name
        echo '<li class="item-current item-current-' . $userdata->
user_nicename . '"><strong class="bread-current bread-current-' . $userdata->
user_nicename . '" title="' . $userdata->display_name . '">' . 'Author: ' .
$userdata->display_name . '</strong></li>';
    } else if ( get_query_var('paged') ) {

        // Paginated archives
        echo '<li class="item-current item-current-' . get_query_var('paged') .
'"><strong class="bread-current bread-current-' . get_query_var('paged') . '"
title="Page ' . get_query_var('paged') . '">' . __('Page') . ' ' .
get_query_var('paged') . '</strong></li>';
    } else if ( is_search() ) {

        // Search results page
        echo '<li class="item-current item-current-' . get_search_query() .
'"><strong class="bread-current bread-current-' . get_search_query() . '"
title="Search results for: ' . get_search_query() . '">Search results for: '
. get_search_query() . '</strong></li>';

    } elseif ( is_404() ) {
        // 404 page
        echo '<li>' . 'Error 404' . '</li>';
    }
}
echo '</ul>';
}
?>

```

Then to call the breadcrumbs onto your page (usually in the header file), simply use the following code:

```
<?php the_breadcrumb(); ?>
```

Trim Headings and Description

wp_trim_words() function trims text to a certain number of words and returns the trimmed text. This function located in **wp-includes/formatting.php**. But, you don't need to edit that function. You have to call the function where you want to trim your content like below,

```
<?php $trimmed = wp_trim_words( $text, $num_words = 55, $more = null ); ?>
```

\$text => Text to trim // Default: None

\$num_words => Number of words // Default: 55

\$more => What to append if \$text needs to be trimmed. // Default: '…'

Below is the example of a normal title of the post

```
<?php
$content = get_the_content();
$trimmed = wp_trim_words( $content, 40, '<a href="'. get_permalink() .'">
...Read More</a>' );
echo '<h1>'.$trimmed.'</h1>';
?>
```

Custom Image Thumbnails

add_image_size() function helps you to add your own image dimensions in wordpress. You have to set it in your theme > functions.php

```
<?php add_image_size( $name, $width, $height, $crop ); ?>
```

\$name* => The new image size name // Default: None

\$width => The post thumbnail width in pixels. Set to 9999 to resize based on \$height // Default: 0

\$height => The post thumbnail height in pixels. Set to 9999 to resize based on \$width // Default: 0

\$crop => Whether to crop images to specified height and width or resize. // Default: false

Available values for **\$crop**

- **false** - Soft proportional crop mode.
- **true** - Hard crop mode.
- **array** - Specify positioning of the crop area (x_crop_position, y_crop_position).

Proportional Crop Mode: This method resizes the image proportionally without distorting it. So you might not get the dimensions that you wanted. Usually it matches the width dimension and the heights are different based on each image's proportion.

Hard Crop Mode: crop the image to the size that we have defined (in this case 120 x 120px). This is the method we use a lot in our theme designs to make sure everything is proportionate and our design is not breaking

Now you can understand the below example.


```
<?php
add_image_size( 'category-thumb', 150, 150 ); //Normal crop
add_image_size( 'latest-article', 75, 75, true ); //Hard crop from left top
add_image_size( 'home-banner', 800, 280, array('center', 'top')); //crop from
mentioned position
add_image_size( 'large', 960, '' ); // Width only important
?>
```

After you added this, your uploaded images will be saved as separate images with all those formats in **wp-content > uploads > current_year > current_month**

You can show your custom image with your new dimension in web page using below code. Open the theme file where you want to display the image and paste the following code:

```
<?php the_post_thumbnail( 'category-thumb' ); ?>
```

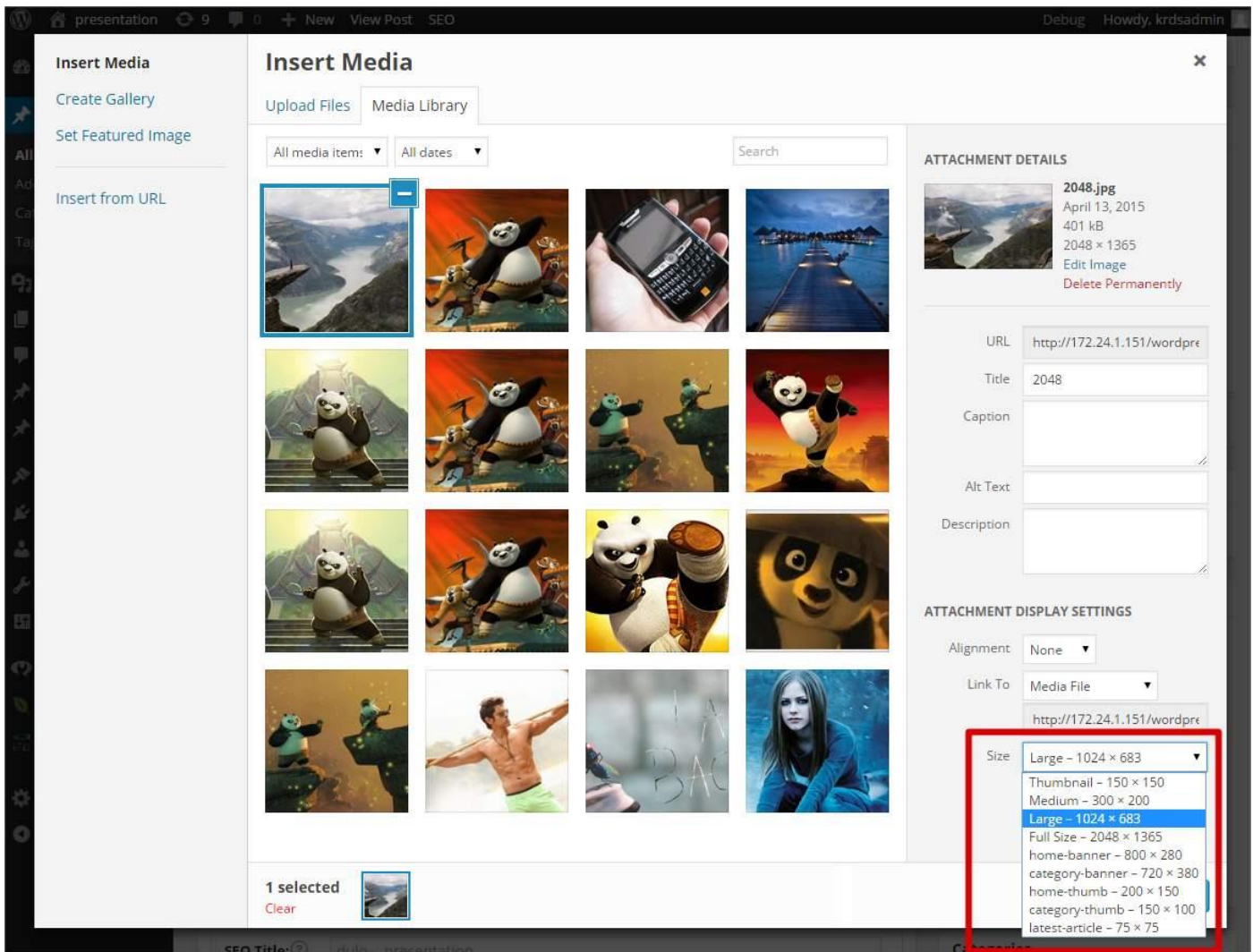
Note: This bit of code must be pasted inside the post loop.

Show the Custom image sizes in admin

To add these new sizes in the drop-down menu list of options in the Media Library when you go to embed an image, You have to call the below code.

```
<?php
function show_image_sizes($sizes) {
    $sizes['home-banner'] = __( 'home-banner', 'image' );
    $sizes['category-banner'] = __( 'category-banner', 'image' );
    $sizes['home-thumb'] = __( 'home-thumb', 'image' );
    $sizes['category-thumb'] = __( 'category-thumb', 'image' );
    $sizes['latest-article'] = __( 'latest-article', 'image' );
    return $sizes;
}
add_filter('image_size_names_choose', 'show_image_sizes');
?>
```

Now all your image sizes will be displayed in image size drop-down of image edit panel.



Specialized Page Template

For more extensive changes, intended for just one specific Page, you can create a specialized template file, named with that Page's slug or ID:

page-{slug}.php

page-{ID}.php

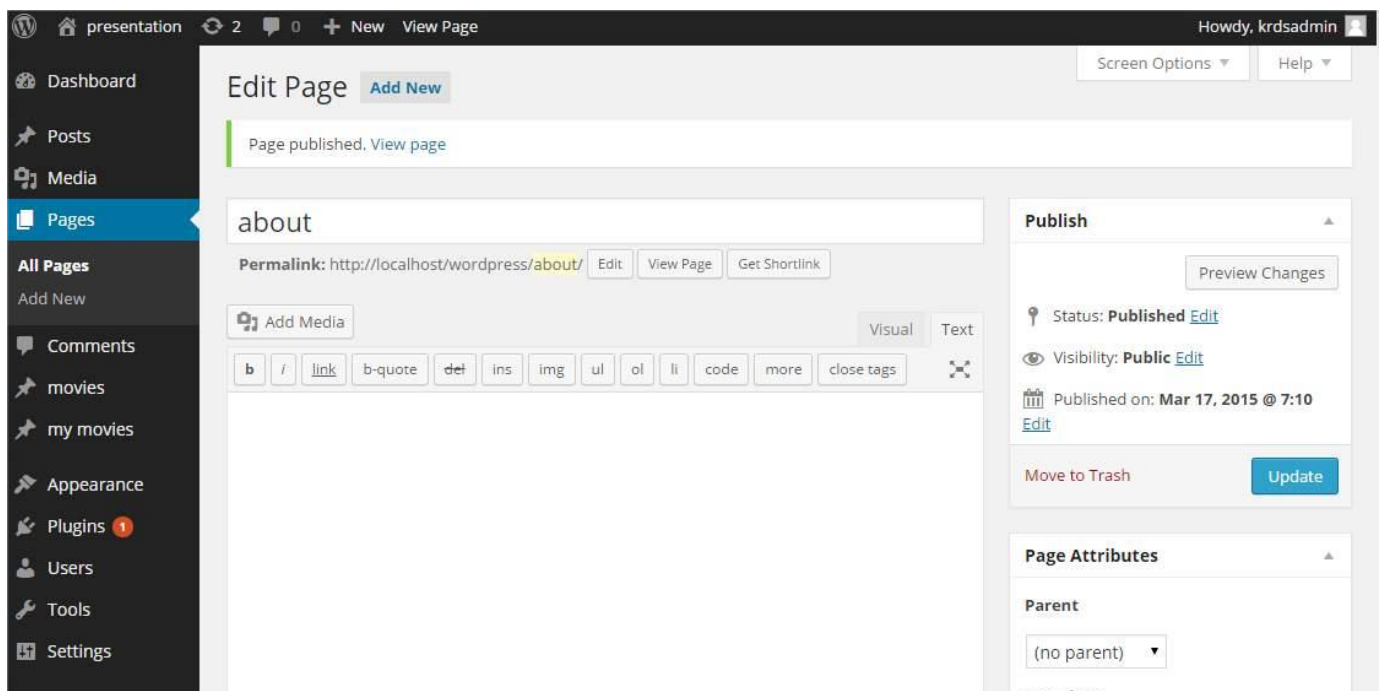
For example: Your About page has a slug of 'about' and an ID of 6. If your active Theme's folder has a file named page-about.php or page-6.php, then WordPress will automatically find and use that file to render the About page.

To be used, specialized page templates must be in your active Theme's folder:

/wp-content/themes/my-theme/

Note: A specialized page template file can not be in a subfolder.

After page added to your theme folder, you have to **publish** a blank page with the same title as slug in the admin panel as below.



You can use the below code to include header, footer and dynamic contents.

```
<?php                                     get_header();                                     ?>

<?php   if   (have_posts())   :   while   (have_posts())   :   the_post();   ?>
<?php                                     the_title();                                     ?>
<?php                                     the_content();                                   ?>
<?php endwhile; endif; ?>

<?php get_footer(); ?>
```

Custom Page Templates

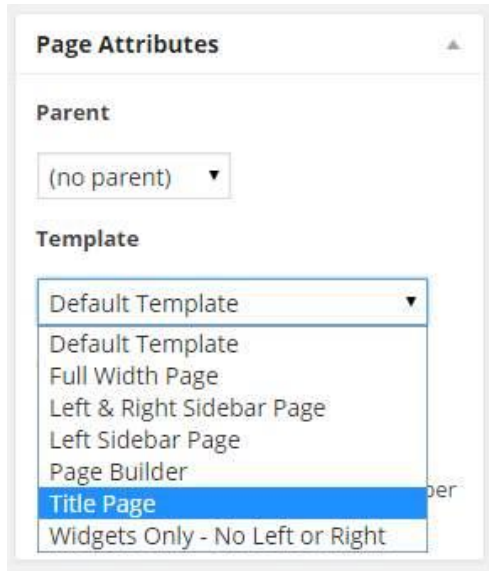
Its a template that can be used by any Page. To create a custom page template, Go to your theme directory (**wp-content > themes > themeDir**). Add a new php page as below,

```
<?php /*
Template Name: Title Page
*/
?>
```

<!-- Your HTML Content -->

Note: Title Page named for example. You can add any name there.

Save the file and to go **Pages -> Add New** in the WordPress admin panel. You should now be able to find a new entry in the **templates drop down** menu in **Page Attributes** Panel.



To begin with we'll add some basic stuff like a header and a footer. Add the following lines below the template declaration.

```
<?php                                     get_header();                               ?>
<?php get_footer(); ?>
```

Add the below lines between header and footer to show the content added in admin panel.

```
<?php      if      (have_posts())      :      while      (have_posts())      :      the_post();      ?>
<?php                                     the_title();                               ?>
<?php                                     the_content();                             ?>
<?php endwhile; endif; ?>
```

A quick, safe method for making a new Page template is to use with a copy of your page.php

Conditional Tags

You can often make smaller Page-specific changes with Conditional Tags in your Theme's page.php file. For instance, this code loads one header file (header-home.php) for your front page, another for your About page (header-about.php), and the default header.php for all other Pages:

```
if ( is_front_page() ) {
```

```

    get_header( 'home' );
} elseif ( is_page( 'About' ) ) {
    get_header( 'about' );
} else {
    get_header();
}

```

Custom Post Types

Custom post types are new post types you can create. A custom post type can be added to WordPress via the `register_post_type()` function. This function allows you to define a new post type by its labels, supported features, availability and other specifics. Here's a basic example of adding a custom post type:

```

add_action( 'init', 'create_post_type' );

function create_post_type() {
    register_post_type( 'acme_product',
        array(
            'labels' => array(
                'name' => __( 'Products' ),
                'singular_name' => __( 'Product' )
            ),
            'public' => true,
            'has_archive' => true,
            'rewrite' => array( 'slug' => 'products' ),
        )
    );
}

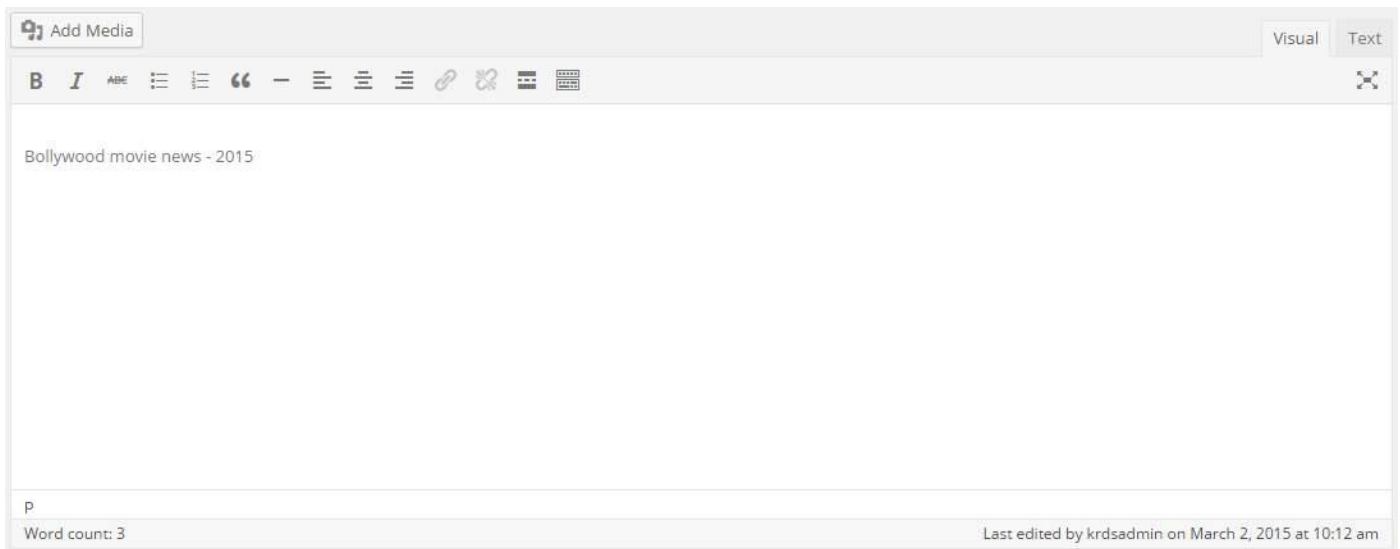
```

This creates a post type named Product identified as `acme_product`. The **`register_post_type()`** function receives two major arguments. The first one is labels which define the name of the post type in both plural and singular forms. The second one is public which is a predefined flag to show the post type on the administration screens and to make it show up in the site content itself, if it's queried for. There are many more arguments you can pass to the `register_post_type()` function. These parameters are optional, and you can use them to configure your post type on a detailed level.

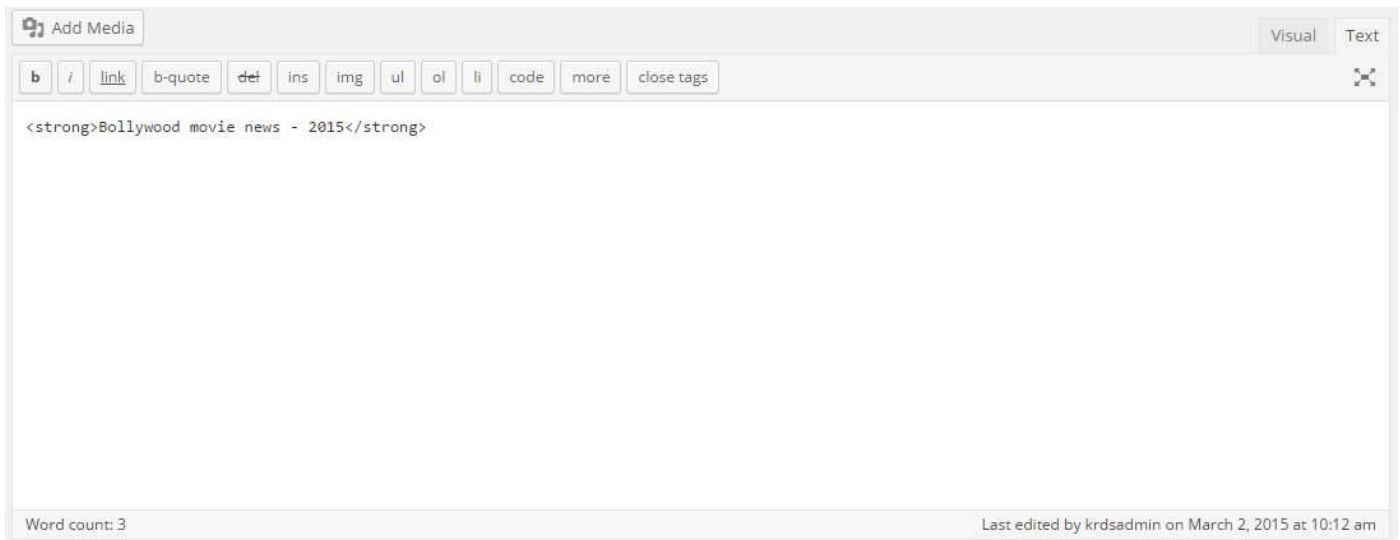
Admin Editor

When writing your post, you have the option of using the Visual or Text mode of the editor. The visual mode lets you see your post as is, while the Text mode shows you the code and replaces the WYSIWYG editor buttons with quicktags.

Visual Editor



Text Editor



The quicktags are explained as follows.

- **b** - `` HTML tag for strong emphasis of text (i.e. bold).
- **i** - `` HTML tag for emphasis of text (i.e. *italicize*).
- **b-quote** - `<blockquote></blockquote>` HTML tag to distinguish quoted or cited text.
- **del** - `` HTML tag to label text considered deleted from a post. Most browsers display as striked through text. (Assigns datetime attribute with offset from GMT (UTC))
- **link** - `` HTML tag to create a hyperlink.

- ins - `<ins></ins>` HTML tag to label text considered inserted into a post. Most browsers display as underlined text. (Assigns datetime attribute with offset from GMT (UTC))
- ul - `` HTML tag will insert an unordered list, or wrap the selected text in same. An unordered list will typically be a bulleted list of items.
- ol - `` HTML tag will insert a numbered list, or wrap the selected text in same. Each item in an ordered list are typically numbered.
- li - `` HTML tag will insert or make the selected text a list item. Used in conjunction with the ul or ol tag.
- code - `<code></code>` HTML tag for preformatted styling of text. Generally sets text in a monospaced font, such as Courier.
- more - `<!--more-->` WordPress tag that breaks a post into "teaser" and content sections. Type a few paragraphs, insert this tag, then compose the rest of your post. On your blog's home page you'll see only those first paragraphs with a hyperlink ((more...)), which when followed displays the rest of the post's content.
- page - `<!--nextpage-->` WordPress tag similar to the more tag, except it can be used any number of times in a post, and each insert will "break" and paginate the post at that location. Hyperlinks to the paginated sections of the post are then generated in combination with the `wp_link_pages()` or `link_pages()` template tag.
- lookup - Opens a JavaScript dialogue box that prompts for a word to search for through the online dictionary at answers.com. You can use this to check spelling on individual words.
- Close Tags - Closes any open HTML tags left open--but pay attention to the closing tags. WordPress is not a mind reader (!), so make sure the tags enclose what you want, and in the proper way.

Workflow Note - With Quicktag buttons that insert HTML tags, you can for example click *i* to insert the opening `` tag, type the text to be enclosed, and click */i* or Close Tags to insert the closing tag. However, you can eliminate the need for this 'close' step by changing your workflow a bit: type your text, select the portion to be emphasized (that is, italicized), then click *i* and your highlighted text will be wrapped in the opening and closing tags.

The Quicktag buttons also have the accesskey JavaScript attribute set, so you may be able to use a keyboard equivalent (e.g., Alt-b for bold) to "press" the button, depending on your browser.

WordPress Shortcode

A Wordpress shortcode is a special tag that you can enter into a post that gets replaced with different content when actually viewing the post on the website.

When you load a blog page with the [gallery] shortcode, WordPress replaces the [gallery] shortcode with all of the code that actually displays a gallery of your images.

To add a wordpress shortcode open **function.php** in your theme directory (**wp-content > themes > themeDir**)

add a function like below,

```
function caption_shortcode($atts, $content = null) {  
    return '<span class="caption">'. $content. '</span>';  
}  
add_shortcode('caption', 'caption_shortcode');
```

then call it in your code as below,

```
[caption]My Caption[/caption]
```

Output will be:

```
<span class="caption">My Caption</span>
```

Attributes in shortcodes

This will allows you to give your dynamic values for n number of times.

```
function get_input_text( $atts ) {  
    $a = shortcode_atts( array(  
        'class' =>  
        'id' =>  
    ), $atts );  
  
    return "<input type='text' id='.esc_attr($a['id']).' class='.esc_attr($a['class']).' value='name' /> ";  
}  
add_shortcode( 'input-text', 'get_input_text');
```

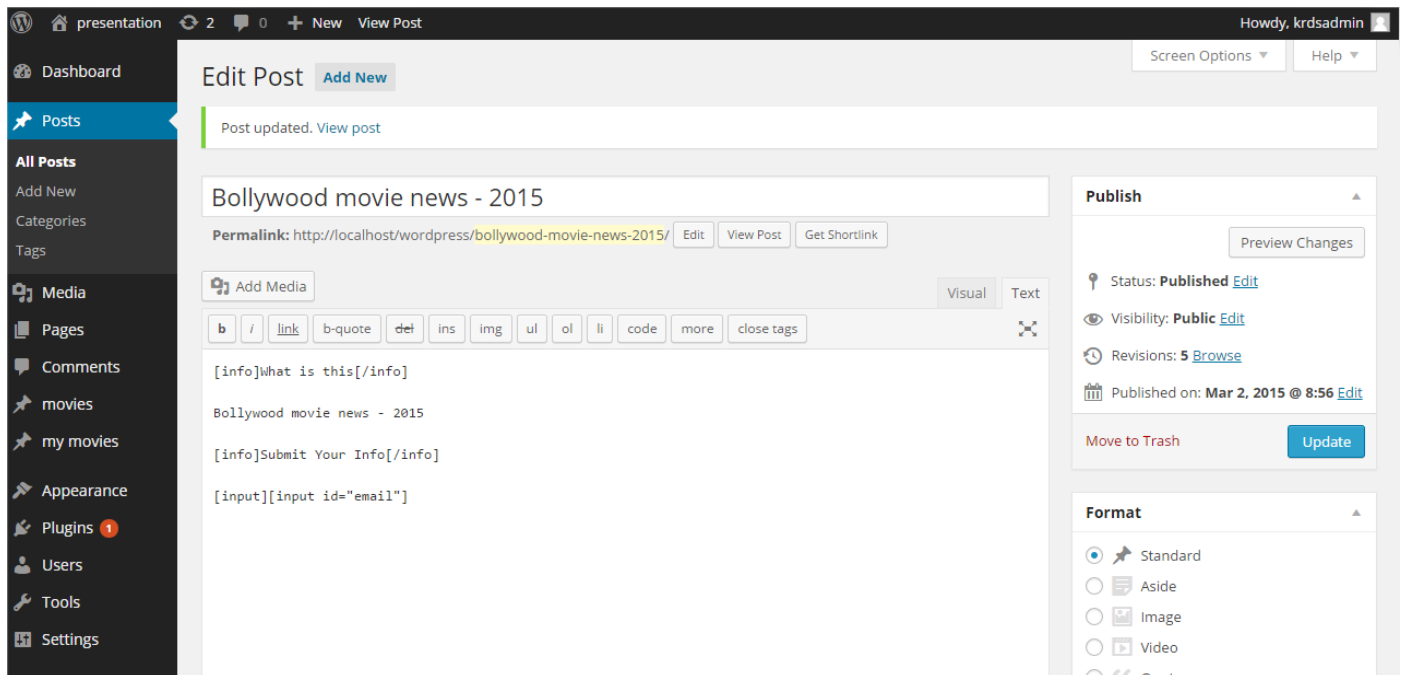
then call it in your code as below,

```
[input-text id="username" class="textBox"]
```

Output:

```
<input type='text' id='username' class='textBox' value='name' />
```


Once you created shortcodes, you can use the shortcodes in admin editor as below,



Wordpress CSS Coding Standard

Wordpress CSS Coding Standard almost meets our KRDS Coding standard. Just few things are different. Those are,

1. Hyphen instead of Camelcase for the class names.
2. We can use ID name too as selector.
3. Logical property ordering than alphabetical.
4. More importance to comments.

Lets see all the CSS Coding Standard given by Wordpress.

Structure

- Use tabs, not spaces, to indent each property.
- Add two blank lines between sections and one blank line between blocks in a section.
- Each selector should be on its own line, ending in either a comma or an opening curly brace.
- Property-value pairs should be on their own line, with one tab of indentation and an ending semicolon.
- The closing brace should be flush left, using the same level of indentation as the opening selector.

Correct:

```
1 #selector-1,
2 #selector-2,
3 #selector-3 {
4     background: #fff;
5     color: #000;
6 }
```

Incorrect:

```
1 #selector-1, #selector-2, #selector-3 {
2     background: #fff;
3     color: #000;
4 }
5
6 #selector-1 { background: #fff; color: #000; }
```

Property Ordering

The baseline for ordering is:

- Display
- Positioning
- Box model
- Colors and Typography
- Other

Example:

```
1 #overlay {  
2     position: absolute;  
3     z-index: 1;  
4     padding: 10px;  
5     background: #fff;  
6     color: #777;  
7 }
```

Selectors

- Similar to the WordPress Coding Standards for file names, use lowercase and separate words with hyphens when naming selectors.
- Avoid camelcase and underscores.
- Use human readable selectors that describe what element(s) they style.
- Attribute selectors should use double quotes around values
- Refrain from using overqualified selectors, **div.container** can simply be stated as **.container**

Correct:

```
1 #comment-form {  
2     margin: 1em 0;  
3 }  
4  
5 input[type="text"] {  
6     line-height: 1.1;  
7 }
```

Incorrect:

```
1 #commentForm { /* Avoid camelcase. */  
2     margin: 0;  
3 }  
4  
5 #comment_form { /* Avoid underscores. */  
6     margin: 0;  
7 }  
8  
9 div#comment_form { /* Avoid over-qualification. */  
10    margin: 0;  
11 }  
12  
13 #c1-xr { /* What is a c1-xr?! Use a better name. */  
14    margin: 0;  
15 }  
16  
17 input[type=txt] { /* Should be [type="text"] */  
18    line-height: 110% /* Also doubly incorrect */  
19 }
```

Comments

Comment, and comment liberally. Long comments should manually break the line length at 80 characters.

For sections and subsections:

```
1  /**
2   * #.# Section title
3   *
4   * Description of section, whether or not it has media queries, etc.
5   */
6
7  .selector {
8      float: left;
9  }
```

For inline:

```
1  /* This is a comment about this selector */
2  .another-selector {
3      position: absolute;
4      top: 0 !important; /* I should explain why this is so !important */
5  }
```

Properties value

- Space before the value, after the colon
- Do not pad parentheses with spaces
- Always end in a semicolon
- Use double quotes rather than single quotes, and only when needed, such as when a font name has a space.
- 0 values should not have units unless necessary, such as with transition-duration.
- Line height should also be unit-less, unless necessary to be defined as a specific pixel value. This is more than just a style convention, but is worth mentioning here. More information: <http://meyerweb.com/eric/thoughts/2006/02/08/unitless-line-heights/>
- Use a leading zero for decimal values, including in rgba().
- Multiple comma-separated values for one property should be separated by either a space or a newline.
- Properties should be followed by a colon and a space.
- All properties and values should be lowercase, except for font names and vendor-specific properties.
- Use hex code for colors, or rgba() if opacity is needed. Avoid RGB format and uppercase, and shorten values when possible: #fff instead of #FFFFFF.
- Use shorthand (except when overriding styles) for background, border, font, list-style, margin, and padding values as much as possible.

Correct:

```
1 .class { /* Correct usage of quotes */
2     background-image: url(images/bg.png);
3     font-family: "Helvetica Neue", sans-serif;
4 }
5
6 .class { /* Correct usage of zero values */
7     font-family: Georgia, serif;
8     text-shadow: 0 -1px 0 rgba(0, 0, 0, 0.5),
9                 0 1px 0 #fff;
10 }
```

Incorrect:

```
1 .class { /* Avoid missing space and semicolon */
2     background:#fff
3 }
4
5 .class { /* Avoid adding a unit on a zero value */
6     margin: 0px 0px 20px 0px;
7 }
```

Media Queries

- Rule sets for media queries should be indented one level in. It is generally advisable to keep media queries grouped by media at the bottom of the stylesheet.
- An exception is made for the **wp-admin.css** file in core, as it is very large and each section essentially represents a stylesheet of its own. Media queries are therefore added at the bottom of sections as applicable.

Example:

```
1 @media all and (max-width: 699px) and (min-width: 520px) {
2
3     /* Your selectors */
4 }
```

Useful Plugins

As i mentioned earlier, Plugins allow you to add new features to your WordPress blog that don't come standard with the default installation. There are a rich variety of Available Plugins for WordPress. Below are the main plugins we need to add,

Akismet

This plugin integrates the free (for personal use) Akismet **comment spam prevention** service with your WordPress site to direct suspicious looking comments straight into your spam folder. While the Akismet plugin is now pre-installed with WordPress, you still need to sign up for an account and activate the plugin on your site.

Link: <https://wordpress.org/plugins/akismet/>

Advanced Custom Fields: Image Crop Add-on

An image field making it possible/required for the user to **crop the selected image** to the specified image size or dimensions. Its really useful one for Integration.

Link: <https://wordpress.org/plugins/acf-image-crop-add-on/>

WP Smush.it

WP Smush.it strips hidden, bulky information from your images, reducing the file size without losing quality. WP Smush.it meticulously scans every image you upload – or have already added to your site – and cuts all the unnecessary data for you.

Link: <https://wordpress.org/plugins/wp-smushit/>

Regenerate Thumbnails

Regenerate Thumbnails allows you to regenerate the thumbnails for your image attachments. This is very handy if you've changed any of your thumbnail dimensions (via Settings -> Media) after previously uploading images or have changed to a theme with different featured post image dimensions. You can either regenerate the thumbnails for all image uploads, individual image uploads, or specific multiple image uploads.

Link: <https://wordpress.org/plugins/regenerate-thumbnails/>

Envira Gallery

Envira Gallery is the absolute easiest, fastest and most efficient gallery plugin for WordPress. Creating galleries is intuitive using all native WordPress features, and your galleries are responsive right out of the box. You can create global and post dependent responsive WordPress galleries that can be exported and imported anywhere.

Link: <https://wordpress.org/plugins/envira-gallery-lite/>

WordPress SEO by Yoast

WordPress SEO is not just an SEO plugin, it is the complete website optimization package. Apart from adding titles and descriptions to your articles, it also allows you to add open graph meta data, do page analysis, publish xml sitemap, and so much more.

Link: <https://wordpress.org/plugins/wordpress-seo/>

W3 Total Cache

W3 Total Cache helps you optimize your WordPress site for speed and performance. It allows you to easily setup page and browser caching, compress pages for quicker downloads, and set up a content delivery network for your static files.

Link: <https://wordpress.org/plugins/w3-total-cache/>

Share Buttons by AddToAny

The WordPress sharing plugin to help people share, save, and email your posts and pages using any service, such as Facebook, Twitter, Pinterest, Google, Reddit, Tumblr, StumbleUpon, LinkedIn, and well over 100 more sharing and social bookmarking sites. AddToAny's customizable platform and social share icons let you optimize your site's blog posts & pages for the best social sharing engagement.

Link: <https://wordpress.org/plugins/add-to-any/>

Custom Post Type UI

This plugin provides an easy to use interface to create and administer **custom post types** and taxonomies in WordPress. This plugin is created for WordPress 3.x. Needs to do some codes in function.php to emulate those custom post types too.

Link: <https://wordpress.org/plugins/custom-post-type-ui/>

WP Mail SMTP

This plugin reconfigures the wp_mail() function to use SMTP instead of mail() and creates an options page that allows you to specify various options.

You can set the following options:

- Specify the from name and email address for outgoing email.
- Choose to send mail by SMTP or PHP's mail() function.
- Specify an SMTP host (defaults to localhost) and SMTP port (defaults to 25)
- Specify an SMTP authentication, username and password.

Link: <https://wordpress.org/plugins/wp-mail-smtp/>

Debug Bar

Debug Bar adds a new menu called Debug at the admin bar. Click on the menu, and it will neatly display information like cache, total queries, total queried time, and memory usage that will be very useful in the debugging process. In addition, you may also see some PHP Warnings and Notices that occur when you enable WP_DEBUG.

Link: <https://wordpress.org/support/plugin/debug-bar>

Disable WordPress Core Updates

This plugin useful to run in production environments with more than one Administrator-level account. You see, updating themes, plugins and especially the core is a dangerous thing to do on your live website. What I usually do is I run all the updates locally, test them out on a playground (testing) server and then deploy over to production. So this plugin disables core updates and with it activated, I can be sure that the other Administrators don't accidentally hit the upgrade button.

Link: <https://wordpress.org/plugins/disable-wordpress-core-update/>

Reference Links

Topic	Link
Back office	http://codex.wordpress.org/Administration_Screens
CSS Coding Standard	https://make.wordpress.org/core/handbook/coding-standards/css/
Page Templates	http://codex.wordpress.org/Page_Templates
New Breadcrumb	http://thewebtaylor.com/articles/wordpress-creating-breadcrumbs-without-a-plugin

Tips for image issues	http://www.wpbeginner.com/beginners-guide/how-to-fix-common-image-issues-in-wordpress/
Useful functions for function.php	https://css-tricks.com/snippets/wordpress/