**Modules implemented in Portal for Collaborating Communities**

The system has been implemented in Drupal, so it is easily extensible with additional modules. Modules provide functionality to the System, and the 'core modules', which are included in a standard installation,that provides all of the basic functions for installation. The addition module installed in the system are used based on the requirements and these addition modules may have some dependency modules which are require to be installed additionally for proper functioning of the system.

1. **Content Creating, Editing and Publishing tools**

Following modules are used for implementing the Workflow in the system are-

**Workbench –** Workbench is a module for implementing the workflow in the system. It provides the core functionality to the additional workbench modules. So the system just requires to install the module.

Workbench gives us three important solutions:

* A simplified workbench interface for users who only have to work with content.
* The ability to control who has access to edit any content based on an workflow structure.
* A customizable editorial workflow that integrates with the access control feature described above or works independently on its own

Dependency modules: Views , Ctools

Configuration:

Step 1: Goto Structure -> Content type -> Articles

Step 2: Select “Publishing options” -> Enable Create new revision -> Enable Moderation of revision

**Workbench Access –** This module provides access right to the user for accessing the workflow. This module is require by Workbench module for implementing the access rights. When creating and editing content, users will be asked to place the content in an editorial section. Other users within that section or its parents will be able to edit the content. A user may be granted editorial rights to a section specific to his account or by his assigned role on the site.

Configuration: Workbench Access creates editorial access controls based on hierarchies. So We have to create some taxonomy (categories) for implementing the access rights.

**Step 1**: Goto Structure -> Taxonomy and create a taxonomy name Workflow and add the terms.

* Workflow
* Create
* Publish

**Step 2**: Goto Configurating -> Workbench Access -> Settings

In the “Active access scheme”- Select “Taxonomy” and select the taxonomy that have created in the previous step.

In Enable the Content type on which the content is going to work – select “Articles”

**Step 3**: Select the “Role” tab at Workbench Access

Create – add Author, Editor and Reviewer roles to the create section.

Publish – add Reviewer role to the publish section.

**Workbench Moderation**: This module is use for adding states and transtions for implementing the workflow. It adds arbitrary moderation states between node "creation" and "published" node states, and affects the behavior of node revisions when nodes are published. Moderation states are tracked per-revision; rather than moderating nodes, so it moderates the revisions.

Some of the states added in the system are Draft, Needs Review, Publishable and Publish.

Configuration:

**Step 1**: Goto Configuration -> Workbench Moderation

Select “States” tab – where we can add as many state as we want.

**Step 2**: Select “Transition” tabs under “Workbench Moderation”

We can add all the probable Transition that can occur from one states to another in this section. The basic transition that we add are –

* Draft to Needs Review
* Needs Review to Draft
* Needs Review to Publishable
* Publishable to Needs Review
* Publishable to Publish

**Dependency modules**: Drafty, Entity API

1. **Localization support for Indian Languages:**

Modules used for implementing the translation are

# Locale: This module helps us to localize and enable, Multi-Lingual Content in the system. Using this module translation can take place local to the system without any of internet. The site interface can be displayed in different languages, as well as nodes can have languages assigned.

# Configuration: This module is available in the core of the system. It gives two new features Language and Translate interface in configuration.

# For adding languages - Goto Configuration-> Regional and Language -> Language

# List of language available to be added in the system are –

# Assamese

# Bengali

# Bihari

# English

# Gujarati

# Hindi

# Kannada

# Kashmiri

# Malayalam

# Marathi

# Nepali

# Oriya

# Punjabi

# Sanskrit

# Tamil

# Telugu

# Title: This module is use for translating the title of a content. It provides an option called “replace” at “Structure->Content types->Articles-> Manage field” for translating node titles to the Field API in order to make nodes fully translatable.

# Dependency module: System (a core module of Drupal)

# Entity Translation: This module is used for translating a content and store that translation in the form of revision for each of the content. In this system each piece of content consists of a single node, but each field on the node can have multiple copies(revision), in different languages, all attached to the same node.

# Configuration:

# Goto Configuration -> Regional and Language -> Entity Translation

# Enable – “Enable translation workflow permission “

# Select Translatable Entity Types as – “Nodes”, “Taxanomy”

# Select default language for Articles as – “English”

# Dependency module: Locale

**Internationalization**: This module is use for translating not only the content of the fields, but also the field labels and description. This module contain a number of sub modules which provides various functionalities to translation. We need to enable the Field Translation” module form the sub module section.

This module fills the gaps that still exist in locale module. A few of the important features which the Internationalization module are:

* A proper multilingual menu system
* Multilingual blocks
* Multilingual taxonomy

# Dependency modules: Variable, Locale

# Configuration: Goto Structure -> Content types -> Edit “Article”

# Select “Publishing options” -> Select “Enable with field translation”

# Localization Update: It automatically downloads and updates our translations by fetching from localize.drupal.org. When a new module is installed all the translation of string for that module is downloaded by this module. This module takes care of update require for all translation.

# Dependency module: Locale

# Language Icons: This module adds icons to language links.

# Dependency module: Locale

1. **The Reputation Model:**

The different modules are implemented for achieving the point based and voting system, and combining them to form the reputation model.

The module used for point based system are -

**Userpoints :** This module is based on point based system. It provides an API for users to gain or lose points for performing certain actions on your site. This is a module that works in conjunction with many other User points contributed module in order to implement complete point based reputation model.

**User points Nodes and Comments** : User points Nodes and Comments uses the Userpoints Core module underneath and grant points to a user when they post a node or a comment and/or moderate a comment.

**Userpoints cap :** This module limit the number of points a user can earn in a system.

**Userpoints Role:** This module set transition of roles when a user earn/lose certain points threshold.

**Userpoints Role Exempt :** This module exempts certain roles from earning userpoints. For eg: Exempt admin from taking part in reputation system.

The module used for Voting system are-

**Voting API :** VotingAPI provides an api and schema for storing, retrieving, and tabulating votes for nodes and comment. It provides api for handling the rating system that includes rating of any content, multi criteria voting and also provides automatic tabulation of results. This module does not expose voting mechanism to end users.

**Rate :**  This modules has been implemented for providing voting mechanism to end user. It provides flexible voting widgets for nodes and comments. This module is dependent on voting api and uses its schema for storing the results. This module provides UI for voting in nodes and comments in the system.

**User Points Voting API :**  This module is a submodule of User point module but it makes it possible for voting widgets that uses Voting API to hook into the User Points ecosystem. This module is both dependent on User points and Voting Api modules. With the help of this module the user can gain or loss points when another user is rating his/her content or comment in the system.

**Configuration for implementing the Reputation Model :**

Step 1: Install all the above mentioned User Points modules. Then

Goto -> Configuration -> Points -> Points Settings

* Set Points that the user will gain/lose from content, comments, revision in the respective Content, Comment, Revision, Content Visits tab.
* For transition of roles set the threshold for each role in Role Change tab .
* If a you want to give different points for different content type then -
* Goto -> Structure -> Content type -> and Select the content type which we want to give different points.

Step 2: Install the Voting API, Rate and User Points Voting API modules. Then

Goto -> Configuration -> User Point Voting API ->

* Set the Points for the User Who is voting to be 0 ( we can set this point to any number if we want to give a user who is voting on a content of another user).
* Set the Point for the user who has created the content in the “Points for author receiving votes” tab.

Goto -> Structure -> Rating Widgets -> Select the widget type -> Set the content on which the widget is going to display.

# **4. Sharing the contents on social networks**

**Share buttons for Drupal** including the AddToAny universal share button, Facebook, Twitter, Google+, Pinterest, WhatsApp and many more.

Vector share buttons use **AddToAny** SVG sharing icons. AddToAny vector icons load efficiently, are mathematically precise, scalable to any size, and stunning on High-PPI screens such as Retina and Retina HD displays.

## **Features:**

* Universal sharing button
* Modern SVG sharing buttons
* Elegant Drupal integration
* AJAX support
* Display in articles, pages, teasers, blocks, custom nodes
* Display via custom modules, themes, templates

# 

## **Configuration:**

**step 1:** First install the “**AddToAny”** module on the site and enable it.

**Step 2:** Next goto configuration option of admin toolbar.

**Step 3:** Under Web services option we can see the “**AddToAny**” option , click on it.

**Step 4:** Here all the AddToAny default settings are available, like:-How the buttons are appears on website, in which content type we want to place, which place we want to established, etc.

**Step 5:** Finally all the settings are done then click on “ Save configuration”.

# **6. Discussion System (Talk)**

By default the discussion link is created at the end of any node. But we want to add a menu tab instead of link then we can use the *Talk* module add a menu tab for discussion added at the top of the every content.

## **Configuration:**

**Step 1:** First install the “**Talk”** module on the site and enable it.

**Step 2:** Next goto configuration option of admin toolbar.

**Step 3:** Under the **CONTENT AUTHORING,** we will get the **Talk page** option, click on it**.**

**Step 4:** Here, you can change the settings of **Talk** module as per the requirements, like: Name of the **Title** (as Discussions), **Links**, etc.

**Step 5:** Then click on **“ Save configuration”**.

**Step 6:** Now, go to the **Structure** then **content type** and then edit the content type (Article, Basic page, etc.) in which content type we want to use this module.

**Step 7:** Then go to the **Comment settings** and do the check mark on the **“Display comments on separate talk page”.**

**Step 8:** All settings are done now, click on **Save content type.**

**7. Advance Search Module**

**Installation and configuration of apache solr with drupal 7**

What is Apache Solr?

Apache solr is an open source server used for making the search feature for the site faster than normal search facility available.

Solr uses an XML (Extensible Markup Language) based query and result language.

Features provided by solr are:

* Advanced Full-Text Search Capabilities -Solr enables powerful matching capabilities including phrases,wildcards and grouping across any data type
* Highlighting the search results
* Optimized for high volume traffic
* Spell Suggestions
* Near real time indexing

**Steps to install Apache solr (5.2.0) on ubuntu**

Step 1: Update the system

First execute the following command to update the system with latest patches and updates.

“sudo apt-get update && apt-get upgrade -y”

Step 2: Setting up the Java Runtime Environment-

* Solr is a Java application, so the Java runtime environment needs to be installed first in order to set up Solr.
* We have to install Python Software properties in order to install the latest Java 8. Run the following command to install the software and Press Y to continue.

*“sudo apt-get install python-software-properties”*

* After executing the command, add the webupd8team Java PPA repository in your system by running following command in terminal:

“ *sudo add-apt-repository ppa:webupd8team/java*”

* Press [ENTER] when requested. Now, you can easily install the latest version of Java 8 with apt.
* First, update the package lists to fetch the available packages from the new PPA:

*“sudo apt-get update”*

* Then install the latest version of Oracle Java 8 with this command

*“sudo apt-get install oracle-java8-installer”*

* Press Y to continue and agree to the license by clicking on the OK button.
* check the version of Java installed by running the following command

*“java -version”*

* It will display the version of java as follows:

Java version "1.8.0\_91"

Java(TM) SE Runtime Environment (build 1.8.0\_91-b14)

Java HotSpot(TM) 64-Bit Server VM (build 25.91-b14, mixed mode)

Step 3: Installing the Solr application

* First type the following command to download the solr 5.2.0 version from the website

*cd /tmp*

*sudo wget https://archive.apache.org/dist/lucene/solr/5.2.0/solr-5.2.0.tgz*

* Now, run the following command to extract the service installation file:

*“sudo tar xzf solr-5.2.0.tgz solr-5.2.0/bin/install\_solr\_service.sh --strip-components=2”*

* And install Solr as a service using the script:

*sudo ./install\_solr\_service.sh solr-5.2.0.tgz*

* Use this command to check the status of the service

*sudo service solr status*

Step 4:Use the Solr Web Interface

* The Apache Solr is now accessible on the default port, which is 8983.
* The admin UI should be accessible at *http://your\_server\_ip:8983/solr*. The port should be allowed by the firewall to run the links. For eg in our case http://127.0.0.1:8983/solr/

Step 5: Configuring Solr with Drupal 7

* Get the latest version of Search API Solr Search from Drupal by executing the following command in terminal

*cd /opt*

*sudo wget https://ftp.drupal.org/files/projects/search\_api\_solr-7.x-1.12.tar.gz*

*sudo tar -zxf search\_api\_solr-\*.tar.gz*

* Next copy Solr configuration files from “Search API Solr” module to solr configuration directory by executing following command in terminal

*sudo rsync -av search\_api\_solr/solr-conf/5.x/ /opt/solr/example/files/conf/*

* Create new Solr core for your website.

*sudo mkdir /opt/solr/server/solr/drupal\_core* //drupal\_core is name of Solr Core

* Copy Solr configuration files to new core

*sudo cp -r /opt/solr/example/files/conf/ /opt/solr/server/solr/drupal\_core/*

* Restart Solr by following command:

cd /opt

sudo service solr restart

* Open Solr admin page - http://your\_ip\_address:8983/solr

For eg in our case it is http://localhost:8983/solr

* To create solr core execute the following command in the terminal

*“sudo su - solr -c "/opt/solr/bin/solr create -c core1 -n data\_driven\_schema\_configs"*

Where **core1** is the name of the core.

* Once again restart Solr.
* Now we have Solr configured, secured with new core.
* Now configure our Drupal site to use Solr for search. Install below modules -
  + Search Api
  + Apache Solr Search
  + Facet Api
  + Apache Solr Autocomplete
* Enable these modules in the site module list page.
* Now we will create one search server and one search index to be used with Solr.
* For Creating server and index we have to navigate to Search Api config page

Go to Configuration and inside Search and metadata block click on Search Api.

For creating Search Server:

* Now click on Add Server link
* Enter the Server name (solr\_server).
* Select HTTP protocol as http.
* In solr host option give your server IP-address. In our case it is localhost.
* In Solr port give the port number on which solr instance is running. In our case it is 8983
* In solr path give the path of the solr core. For e.g in our case */solr/drupal\_core*
* Click on Save Settings button.

For creating Search Index:

* Click on Add Index.
* Give the index name(we have given solr\_index).
* In server option select the name of server we have created before i.e solr\_server.
* In Datasource option select the Bundles(Content types) which we want to index.
* Check on index item immediately.
* Give the cron batch size as per the requirement.
* Click on Save Settings.
* Now goto FIELDS tab and check the required field to be index.
* Click on save changes.
* Navigate to FILTERS tab select the filter criteria as per requirements.
* Click on save configuration.
* Now navigate to the FACETS tab.
* Here we can see all the fields we have selected.
* By going to the configure display option we can change the display behaviour of the facets.

**8. Registration role**

This module allows the system to automatically assign a role when a user is a new resigtered user in the system. The selected role will be assigned to new registrants which can be configured by the administrator. It is very lightweight module which does not install any database tables.

**Configuration:**

**Step 1:** First install the “**Registration role”** module on the site and enable it.

**Step 2:** Click on the **Configuration** option under **PEOPLE** block select **Registration role** option.

**Step 3:** Now, here we have a **“Role to automatically assign to new registrants”** option and below all the roles of your website. Select the specific role for the newly registered user.

**Step 4:** Finally, click on **Save Configuration** button.

This role does not have any privileges that should not be given out to just anyone who registers. Because we shouldn't give away any real abilities above an authenticated user by default, this module really has only a few known useful applications:

1. If you want to assign all people who sign up after (or before) a certain time to a role to distinguish them - simply set the roles as appropriate at the appropriate time.
2. If you have multiple sites with a shared user database table, and you want to assign users a role based on the site at which they register.
3. If you want people who register themselves to have a distinct role from users invited by an administrator.
4. If paired with a custom or contrib token system to restrict registration.

**9. Avoiding Spams (CAPTCHA)**

# A CAPTCHA is a challenge-response test often placed within web forms to determine whether the user is human. The purpose of CAPTCHA is to block form submissions by spam, which are automated scripts that post spam content everywhere they can.

## **Configuration:**

**step 1:** First install the “**captcha”** module on the site and enable it.

**Step 2:** Next goto configuration option of admin toolbar.

**Step 3:** under people option we can see the “captcha” option , click on it.

**Step 4:** On the top we can select the Default challenge type: Either Math or Image. We have selected Image.

**Step 5:** Next we can see the different form id available on our site and next to each form id select the challange type from drop down.

**Step 6:** Finally click on “ Save configuration”