

Optimization of WordPress Multisite Data to MVC based Sites for Multilingual Analysis

Manikanta V*

SRP in CIIL, Mysore, Karnataka, India and Research Scholar in Bharathiar University, Coimbatore, Tamilnadu, India
E-mail: ciil.manikanta@gmail.com

Abstract--- WordPress is open source software to implement websites effectively with many plugins and many codes belong to PHP. The WordPress plugins are supported in many issues of form activities. But many of the plugins are not supported to the longer times, updates are required and license is required. So, from this all phenomenon, MVC based modules are easily develop multisite language and developer create, manage data and code easily compare to WordPress dependency. The language speakers of population are easily explored the activities of WordPress multisite data and MVC based data.

Keywords--- WordPress, MVC, Multilingual, Multisite Data

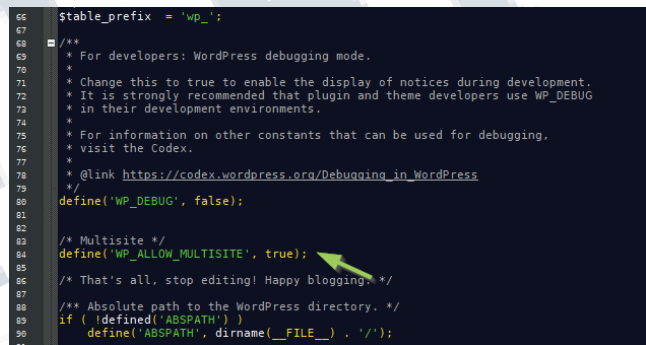
I. INTRODUCTION

Multisite is create multiple websites in a single website features and manages all the networks. In WordPress dashboard is easily implement multisite. Multisite users are mostly used same plugins and themes. The multisite provide separate folder for uploading or downloading data. The one site user cannot access different site data of images or files. The multisite networks used in many corporations, universities, institutes, news, online business and more. The multisite is mainly activated to blog from blog_id throughout the programming.

How to enable a multisite network in WordPress?

In the location of wp-config.php file to add the following lines

```
define('WP_ALLOW_MULTISITE', true);
define('WP_DEBUG', false);
define('MULTISITE', true);
define('SUBDOMAIN_INSTALL', true);
define('DOMAIN_CURRENT_SITE',
'sitename.com');
define('PATH_CURRENT_SITE', '/');
define('SITE_ID_CURRENT_SITE', 1);
define('BLOG_ID_CURRENT_SITE', 1);
define('WP_MEMORY_LIMIT', '5126M');
```



```

66 $table_prefix = 'wp_';
67
68 /**
69  * For developers: WordPress debugging mode.
70  *
71  * Change this to true to enable the display of notices during development.
72  * It is strongly recommended that plugin and theme developers use WP_DEBUG
73  * in their development environments.
74  *
75  * For information on other constants that can be used for debugging,
76  * visit the Codex.
77  *
78  * @link https://codex.wordpress.org/Debugging_in_WordPress
79  */
80 define('WP_DEBUG', false);
81
82
83 /* Multisite */
84 define('WP_ALLOW_MULTISITE', true);
85
86 /* That's all, stop editing! Happy blogging. */
87
88 /** Absolute path to the WordPress directory. */
89 if ( !defined('ABSPATH') )
90     define('ABSPATH', dirname(__FILE__) . '/');
91

```

Figure 1 – wp-config.php

In the WordPress dashboard choose tools inside Network setup to make or configure multisite network. The network provides sites will be own subdomain or subdirectory type.

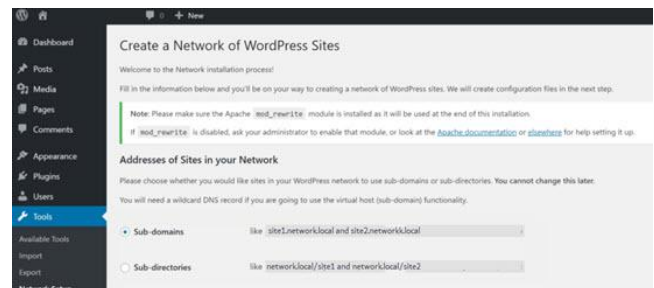


Figure 2 – Create a Network of WordPress Sites

Settings of .htaccess for multisite

```
RewriteEngine On
RewriteBase /
RewriteRule ^index.php$ - [L]
```

```
# add a trailing slash to /wp-admin
RewriteRule ^([_0-9a-zA-Z-]+)/wp-admin$ $1wp-admin/ [R=301,L]

RewriteCond %{REQUEST_FILENAME} -f [OR]
RewriteCond %{REQUEST_FILENAME} -d
RewriteRule ^ - [L]
RewriteRule ^([_0-9a-zA-Z-]+)/?(wp-content|admin|includes).* $2 [L]
RewriteRule ^([_0-9a-zA-Z-]+)/?(.*\.php)$ $2 [L]
RewriteRule . index.php [L]
```

The multisite network admin are used to select themes, page, etc. The setting from multisite network admin it will effect to all the sites once they are changing properties. The multisite provides add, delete, archive and deactivate of any sites. The WordPress is easiest way to manage content and much functionality of themes and plugins. Multisite website is needs when a similar feature of the functionality appears in other sites also. The WordPress somebody express ‘one install many sites’. The WordPress provides multiple user level access functionality. The WordPress requirement.

- 1) Many sites executes by one installation.
- 2) Only one time installation and no need to several times for others.
- 3) User level accesses are easily manages the sites.
- 4) Similar themes and plugins are added.
- 5) WordPress site is affordable in business sites.
- 6) The multisite is adopting shared code.
- 7) The new updates of streamlined through upcoming products.

The files of wp-config.php, the .htaccess and uploads folder files are exist in WordPress site. In multisite, uploads folder are uploading files each sites with their blog_id folder name when any files mechanism. The database of WordPress contains 12 tables in which settings and content of page is stored. In network multisite website 10 tables are duplicated. The tables are changed with their blog_id index of all the table names. Many sites is required more number of tables are generated with their blog_id.

WordPress Data and Scheme structures

In the WordPress Database diagram are use tables like wp_termmeta, wp_users, wp_terms, wp_term_taxonomy, wp_posts, wp_term_relationships, wp_options, wp_postmeta, wp_usermeta, wp_comment, wp_commentmeta, wp_links. In multiste, we are using wp_blogs, wp_blogs_version, wp_registration_log,

wp_signups, wp_site, wp_sitemeta.
wp_blogs -> created of each site and stored in data based on blog_id.
wp_blogs_version-> Database version status in represent the each site.
wp_registration_log-> the table will stored when admin creating a new site.
wp_signups-> this table performs registration process.
wp_site-> Sites address is stored.
wp_sitemeta-> Site information.
 When we create multisites WordPress then the database tables are created multiple tables of wp_posts, wp_postmeta, wp_terms, etc.

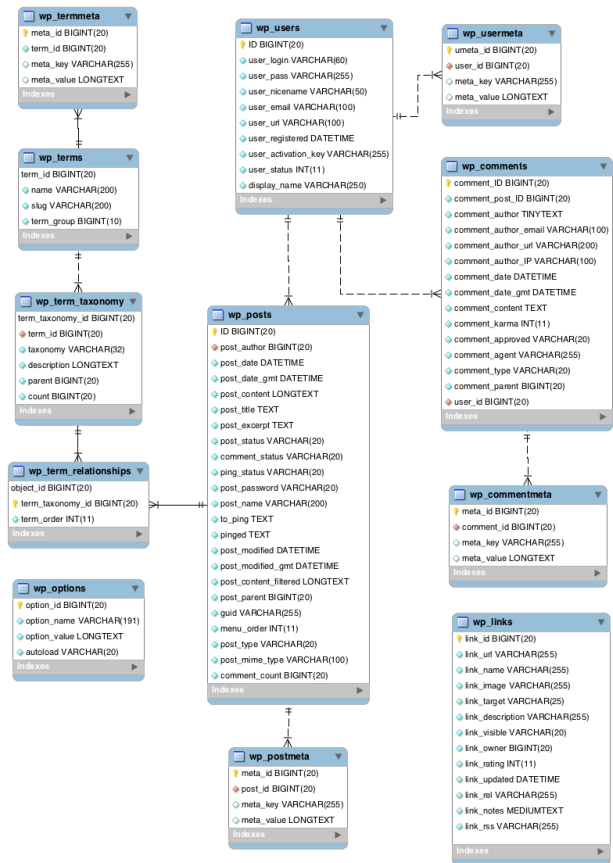


Figure 3 – Database Diagram

Tables of WordPress single site

- wp_commentmeta
- wp_comments
- wp_links
- wp_options
- wp_postmeta
- wp_posts

- *wp_terms*
- *wp_termmeta*
- *wp_term_relationships*
- *wp_term_taxonomy*
- *wp_usermeta*
- *wp_users*

Tables of WordPress Multisite

- *wp_blogs*
- *wp_blogs_versions*
- *wp_commentmeta*
- *wp_comments*
- *wp_links*
- *wp_options*
- *wp_postmeta*
- *wp_posts*
- *wp_registration_log*
- *wp_signups*
- *wp_site*
- *wp_sitemeta*
- *wp_terms*
- *wp_termmeta*
- *wp_term_relationships*
- *wp_term_taxonomy*
- *wp_usermeta*
- *wp_users*

SITE SPECIFIC TABLES

- **Site Specific Tables**
- *wp_commentmeta*
- *wp_comments*
- *wp_links*
- *wp_options*
- *wp_postmeta*
- *wp_posts*
- *wp_terms*
- *wp_termmeta*
- *wp_term_relationships*
- *wp_term_taxonomy*

MVC

The PHP is introduced MVC framework platform to design application in many platforms that will divide the application into data, presentation and business logic method. The MVC is looks for model, view and controller. The controller is handling between the models and views. The student activity observation of view from teacher and teacher handles and controls the student giving brain storm techniques.

Student activities - > View

Teacher -> Controller
Techniques applied -> Models
PHP MVC Frameworks is working with many difficult technologies

1) Hiding all the implementation details.
2) It provides standard methods to build application in our requirement.

3) The developer are using technology of MVC is high. The connectivity database, sanitizing user input, PDO database objects and it are partially implemented.

4) THE MVC is one of professional standard codes.

MVC Design pattern

Model: This is mainly deals with business logic and the application data. This will be used as validation, data process and store. The Data handle based on databases, XML or other data sources.

Controller: The controller handles with user webform handling with their name based method. The forms and loads the model can be initiated in controller and links the views and models based on the requested resources.

Views: The view handles the data to presenting to the user. This is normally websites location files. The MVC provides lot of framework like CodeIgniter, Kohana, CakePHP, Zend, etc.

II. PROBLEMS

The sites of the individual data based on *blog_id* and the tables are adding to continuously *wp_posts*, *wp_1_posts*, *wp_2_posts*, etc in all the tables of their respective tasks. The later is very difficult to manage the data when more than 100 sites and each time we look up the table and when we search the data belonging to all is very complex. In multisite, each site has its own table and structure and optimizing the features are not good. The wordpress site always depends on the plugins and it is difficult to handle without free version. The WordPress consists of many tables with many sites for using individually. The WordPress site is not easy to implement user's requirement for design and coding. The coding is open source and it will take time to resolve the problem. The wodpress sites error log use when we use multisites and server may be problem. The WordPress site is mainly belonging to the themes and changing theme is required more time to our requirement. The WordPress sites are used to third party tools when we are using multisite. The WordPress multisite belonging to SSL wildcard domains are slow performed compares to main domain.

III. METHODOLOGY

While visualizing, Indian Languages is consists of 121 languages from which 23 scheduled and 98 non scheduled languages and many subsites are required for viewing

language wise with all translations. From, this translation of each word is required and updating translations are required using optimizing techniques of the WordPress multisite data to optimized mysql data.

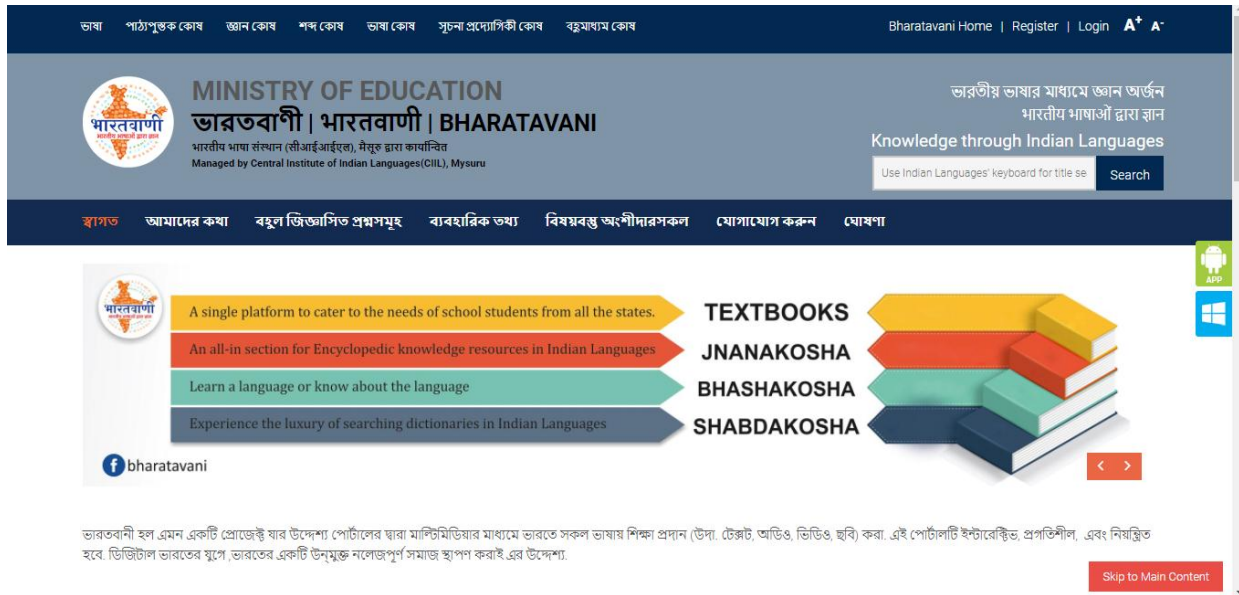


Figure 4: Bengali language

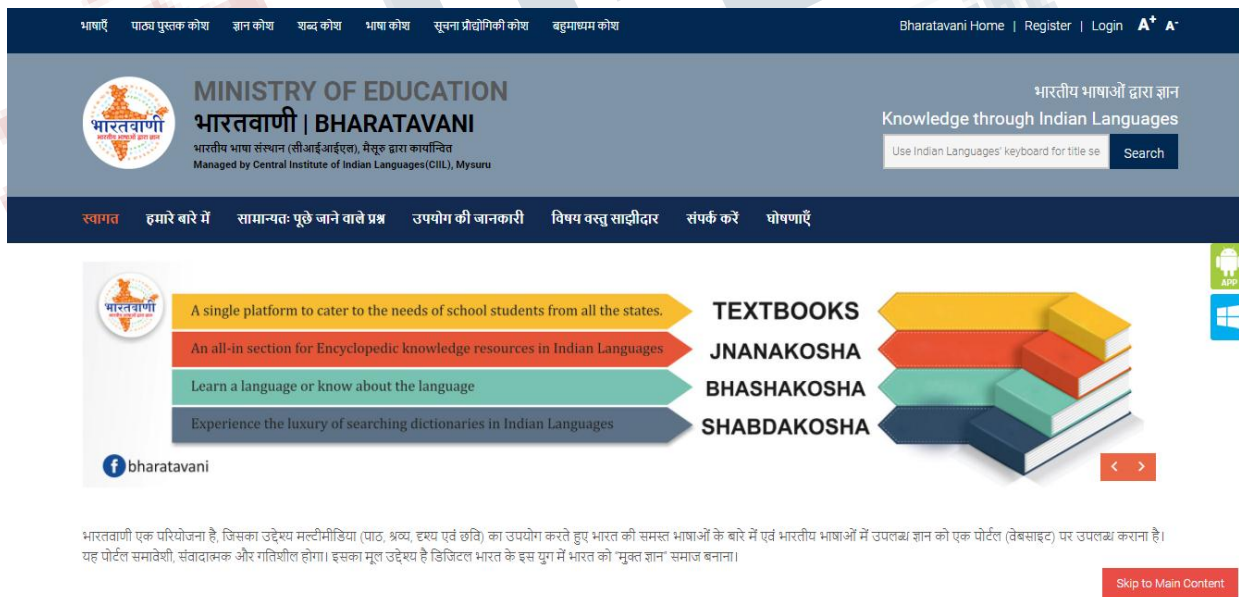


Figure 5: Hindi language

From Fig. 4 and 5 represents the WordPress multisite of different languages with same features.

Creating .htaccess file
Options -MultiViews
RewriteEngine On
Options -Indexes


```
RewriteBase /language/
RewriteCond %{REQUEST_FILENAME} !-d
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-l
RewriteRule ^(.+)$ index.php?url=$1 [QSA,L]
```

```
index.php
<?php
if (file_exists('vendor/autoload.php'))
{
    require 'vendor/autoload.php';
}
require 'application/config/config.php';
require 'application/libs/application.php';
require 'application/libs/controller.php';
$app = new Application();
```

Inside application folder to creating model, view, controller and config files wherever necessary.

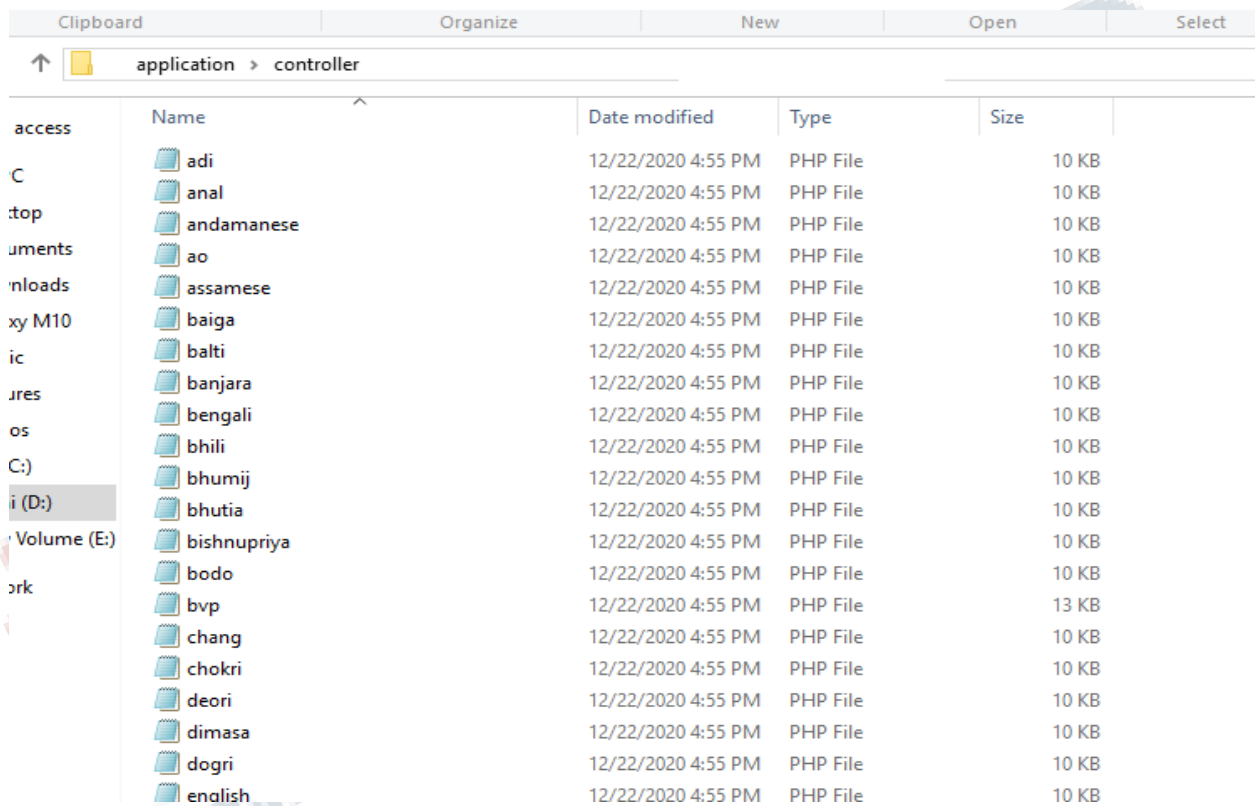


Figure 6: Controller

The controller creates the different languages with similar features of languages and blog_id is required to change their respective language site id from wp_blogs table.

Ex: Adi language creates controller for loading

modules (MainModel for index), pages requirement and blog_id. From Adi language blog_id=109, function to load pages index, fileview, editdata, editsearch, etc from view pages of PHP. With reference of blog_id and it easily executes same features for all the languages.

**International Journal of Engineering Research in Computer Science and Engineering
(IJERCSE)
Vol 8, Issue 4, April 2021**

```

1 <?php
2
3 class Adi extends Controller
4 {
5     public function index()
6     {
7         $blogid=109;
8         $fimages = $this->loadModel('MainModel');
9         $fb=$fimages->getblogs($blogid);
10        $fs = $fimages->getFimages($blogid);
11        require 'application/views/home/index.php';
12    }
13
14    public function fileview()
15    {
16        $blogid=109;
17        require 'application/views/home/fileview.php';
18    }
19
20    public function editdata()
21    {
22        $blogid=109;
23        require 'application/views/home/editdata.php';
24    }
25
26    public function editsearch()
27    {
28        $blogid=109;
29        require 'application/views/home/editsearch.php';
30    }
31
32    public function book()
33    {
34        $blogid=109;
35        $bd = $this->loadModel('MainModel');
    
```

Creating Models: 'MainModel' model is creating functions along with database configuration. The models are created with requirement of designing the website. The model showing that getblogs() function in the following.

So, we create all the functionality to execute the website works. Similarly, create to many models and load models based on website requirements.

```

1 <?php
2
3 class MainModel
4 {
5     /**
6      * Every model needs a database connection, passed to the model
7      * @param object $db A PDO database connection
8      */
9     function __construct($db)
10    {
11        try {
12            $this->db = $db;
13        } catch (PDOException $e) {
14            exit('Database connection could not be established.');

```

Views: All the websites related pages.

application > views > home >

Name	Date modified	Type	Size
assets	12/22/2020 8:08 PM	File folder	
classes	12/22/2020 8:08 PM	File folder	
core	12/22/2020 8:08 PM	File folder	
css	12/27/2020 1:40 PM	File folder	
db	12/22/2020 8:08 PM	File folder	
functions	12/22/2020 8:08 PM	File folder	
img	12/27/2020 1:40 PM	File folder	
includes	12/22/2020 8:08 PM	File folder	
jquery.ime-master	12/22/2020 8:09 PM	File folder	
js	12/27/2020 1:40 PM	File folder	
scss	12/27/2020 1:40 PM	File folder	
upload	12/22/2020 8:09 PM	File folder	
vendor	12/27/2020 1:37 PM	File folder	
a_bhashakosha	12/27/2020 4:34 PM	PHP File	15 KB
a_index	12/24/2020 5:50 PM	PHP File	7 KB
a_jnanakosha	12/26/2020 5:57 PM	PHP File	15 KB
a_multimedia	12/24/2020 5:50 PM	PHP File	13 KB
a_textbook	12/27/2020 1:04 AM	PHP File	15 KB
about	12/22/2020 4:55 PM	PHP File	3 KB
afterlogin	12/22/2020 4:55 PM	PHP File	1 KB
afterregister	12/22/2020 4:55 PM	PHP File	0 KB
announcements	12/22/2020 4:55 PM	PHP File	2 KB
author_delete	12/22/2020 4:55 PM	PHP File	1 KB

Figure 7: Views of the pages

Optimization Techniques

1) Integrate the sql data based on blog_id

From this, site_id is represents based on blog_id and their tables are represented in blog_id.

2) Converts the integrated of respective tables to a single site table with blog_id like wp_posts, wp_postmeta, etc. of tables using many SQL Queries.

3) The same views can be used in many sites through controller.

4) The controller is easy to maintain application to mention the webforms and models. The PDO objects are easily resolve many techniques compared to others.

5) Create controller page for each language with same page name with their respective blog_id.

6) Based on blog_id to implement the models and view of the pages.

7) The model is consists of PDO datasets to select and retrieve the data function based on their respective functionalities.

8) The views are like one common theme for all the sites controls from the controller.

The pages are belonging to broadcast then the broadcast sites are not easy to maintain in WordPress sites. From this extract broadcast data and based on broadcast to select the languages accordingly w.r.to the blog_id. The optimization of WordPress multisite data to mysql data based on blog_id and broadcast using many queries with their respective categories, taxonomies, posts, term meta, comments, etc. The optimization of data belongs to the easiest way to implement the code and manages fastly, accurately.

IV. RESULTS

The MVC architecture is easy to optimize the data and create the same sites and execution based on blog_id. This one idea brings all the functionalities works compares to WordPress multisite approach. The Languages are translation and updated easy based on blog_id. The pages are belonging to broadcast sites are easily maintain compare to WordPress. The languages are translated with same features with all functionality to select blog_id of Bengali is 6 in Fig. 8 and blog_id of Hindi is 3 in Fig. 9.

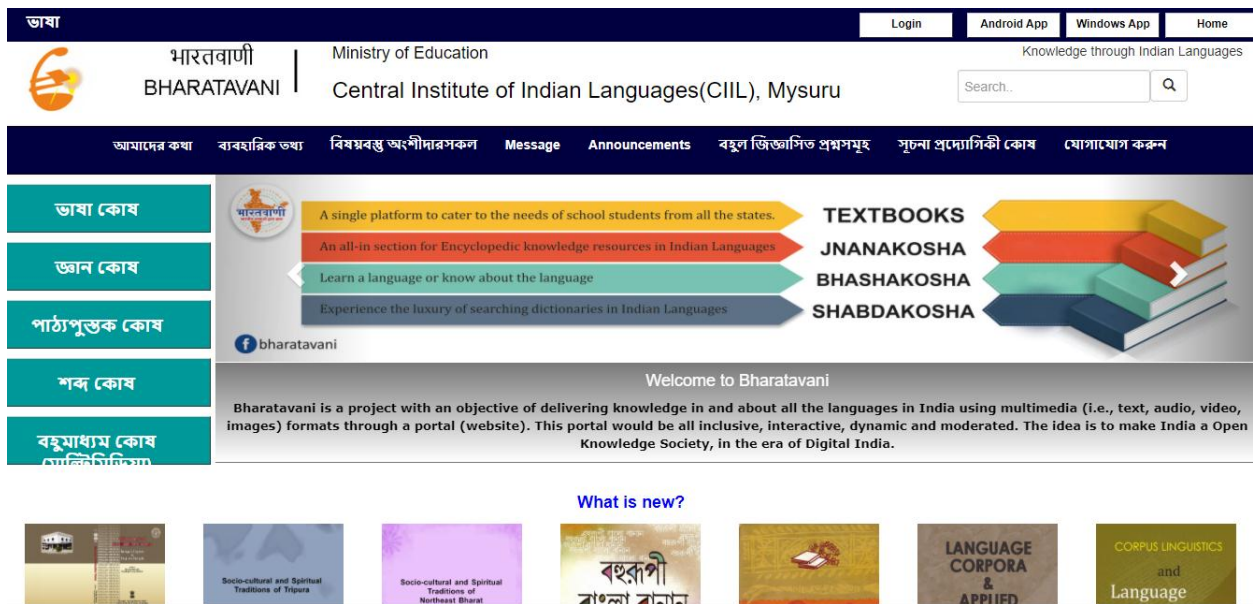


Figure 8: MVC based site of Bengali language (languages/home/Bengali)

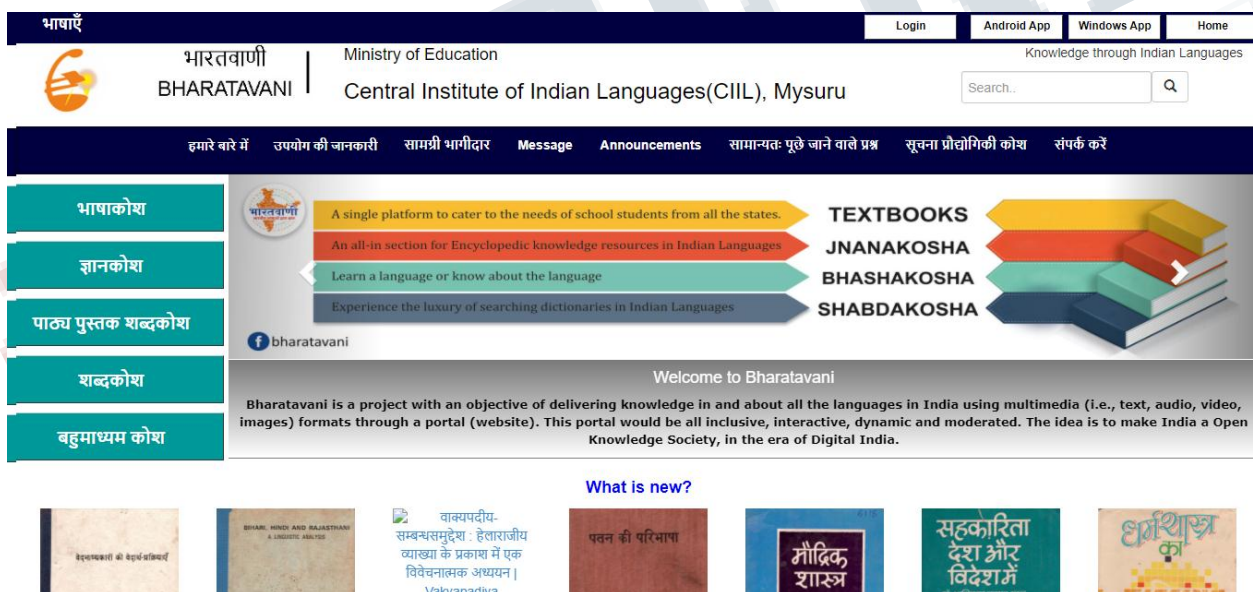


Figure 9: MVC based site of Hindi language (languages/home/hindi)

From our development aspects easy to handle the data and manages according to our requirement of the website activities. The multisite activity is easy to deployed the features and functionality easily comparing to any other multisite. The approach of MVC based sites is visible with controller name of the file in website. The MVC website is easily handled with the models whenever our requirement depends. The WordPress site is mainly belonging to feature with one site is good for working and data with

more site is very difficult to manage their functionality. The data with huge amount of information in multi website is difficult to extract the features of the data immediately. The optimization of data is easily provides when we executes the query based on blog_id. The model, view, controller will handles huge numbers of functionality to create our requirement easy. Mapping and changing of data is very difficult to maintain when we using custom query.

V. CONCLUSION

The idea belongs to blog_id is one of the great tools to support any platforms while creating multisite. The similar features are approached like themes and also we can change any pages based on our requirement. The normalized data are easy to handle the data with our requirement belonging to knowledge development approach. The visualization of data from WordPress multisite data is difficult to manage in while the site requirement of plugin failed updates, etc. The WordPress site is always updated day to day. The technologies are enhanced in multiple features while changing, updates, the theme compatibility are not suitable and searching with many options are complex. The file uploads option are mandatory for their requirement and all the WordPress table of URL belonging to the sub site with their featured image and content url also website with site name. The sites are loading with their URL with domain name. The optimization of the featured image and URL location based on directive requirement using mysql data query. The websites are fastly executing compare to URL location mentioning. The functionality of URL with their aspect is not good comparing to multisite. Each and every sites are mentioned with their respective subsite location instead of single sites upload locations. The multisite website is easily explored in all the way. The SSL is easy when we are subdomains without wildcard SSL. The wildcard SSL it runs slowly in server compare to normal SSL. So, we create single SSL for many websites using MVC model. The framework is providing set of libraries for partial implementation. The PHP is very much suitable platform of tested MVC. This movement is good approach develop application in separates the data and presentation.

REFERENCES

- [1] Brain Messeniechner and Jason Coleman, Building Web Apps with WordPress
- [2] James Mallison, Mastering PHP 7.
- [3] Krishna Shasankar, Zend Framework 2.0 by Example.
- [4] Nathan George. WordPress FOR BEGINNERS: A Visual Guide to Building Your WordPress Site. 2018 Edition.
- [5] Arun K Pujari, Data Mining Techniques, University press. 2001.
- [6] Jaiwei Han, Michelinne Kamber. Data Mining : Concepts and Techniques.
- [7] C S R Prabhu. Data Warehousing Concepts, Techniques and Applications 2nd Edition. Prentice Hall of India, 2002.
- [8] <http://www.newhope.com/food-amp-beverage/role-functional-foods-promoting-gut-health>