Summer Fellowship Report

On

Fossee Web Development

Submitted by

Piyush Kumar

Under the guidance of

Prof.Kannan M. Moudgalya

Chemical Engineering Department

IIT Bombay

Piyush Kumar

June 4, 2020

Acknowledgment

First and foremost, I would like to express my sincere thanks to my project head, Prof.Kannan M. Moudgalya, who gave me this valuable opportunity to work in such a learning environment at FOSSEE.

I would like to express my special gratitude and thanks to my project mentors, Mr. Prashant Sinalkar and Ms. Shashi Rekha for their guidance, constant supervision and for imparting their knowledge and expertise in this field.

Finally, I am deeply thankful to my parents and teachers who helped and inspired me throughout this fellowship.

Sincerely
Piyush Kumar
IIIT Manipur

Contents

Chapter 1

Introduction

1.1 Drupal

Drupal is a web-oriented content management system (CMS) that grew out of a personal project started in 2000 at the university of Antwerp, Belgium, by a student named Dries Buytaert. The project founders designed the original software to allow a group of peoples to share their thoughts and files via electronic means. The first public website built with Drupal was Drop.org.

In 2001, the software behind the project was first released with the name Drupal. The word Drupal is pronounced ”droo.puhl” and derives from the English pronunciation of the Dutch word druppel, which means drop. Today thousands upon thousands of web sites are built using Drupal; it is one of the most popular Web content management systems in the world. And it’s this success Story that probably led you to consider Drupal for your project.

Drupal is a free and open-source web content management framework written in PHP and distributed under the GNU General Public License. Drupal provides a back-end framework for at least 2.3% of all websites worldwide – ranging from personal blogs to corporate, political, and government sites. Systems also use Drupal for knowledge management and for business collaboration.

The standard release of Drupal, known as Drupal core, contains basic features common to content-management systems. These include user account registration and maintenance, menu management, RSS feeds, taxonomy, page layout customization, and system administration. The Drupal core installation can serve as a simple website, a single- or multi-user blog, an Internet forum, or a community website providing for user-generated content.

Drupal also describes itself as a Web application framework. When compared with notable frameworks Drupal meets most of the generally accepted feature requirements for such web frameworks.

Although Drupal offers a sophisticated API for developers, basic Web-site installation and administration of the framework require no programming skills.

Drupal runs on any computing platform that supports both a web server capable of running PHP and a database to store content and configuration.

Originally written by Dries Buytaert as a message board, Drupal became an open source project in 2001. The name Drupal represents an English rendering of the Dutch word druppel, which means "drop" (as in a water droplet). The name came from the now-defunct Drop.org website, whose code slowly evolved into Drupal. Buytaert wanted to call the site "dorp" (Dutch for "village") for its community aspects, but mistyped it when checking the domain name and thought the error sounded better.

Interest in Drupal got a significant boost in 2003 when it helped build "DeanSpace" for Howard Dean, one of the candidates in the U.S. Democratic Party's primary campaign for the 2004 U.S. presidential election. DeanSpace used open-source sharing of Drupal to support a decentralized network of approximately 50 disparate, unofficial pro-Dean websites that allowed users to communicate directly with one another as well as with the campaign. After Dean ended his campaign, members of his Web team continued to pursue their interest in developing a Web platform that could aid political activism by launching CivicSpace Labs in July 2004, "...the first company with full-time employees that was developing and distributing Drupal technology." Other companies began to also specialize in Drupal development. By 2013 the Drupal website listed hundreds of vendors that offered Drupal-related services. In the Drupal community, "CORE" refers to the collaboratively built codebase that can be extended through contributory modules and – for versions prior to Drupal 8 – is kept outside of the "sites" folder of a Drupal installation. (Starting with version 8, core is kept in its own 'core' sub-directory.) Drupal core is the stock element of

Drupal. Common Drupal-specific libraries, as well as the bootstrap process, are defined as Drupal core; all other functionality is defined as Drupal modules including the system module itself.

In a Drupal website's default configuration, authors can contribute content as either registered or anonymous users (at the discretion of the administrator). This content is accessible to web visitors through a variety of selectable criteria. As of Drupal 8, Drupal has adopted some Symfony libraries into Drupal core.

1.1.1 Drupal Modules

A Drupal module is a collection of files containing some functionality and is written in PHP. Because the module code executes within the context of the site, it can use all the functions and access all variables and structures of Drupal core. In fact, a module is no different from a regular PHP file that can be independently created and tested and then used to drive multiple functionalities.

This approach allows Drupal core to call at specific places certain functions defined in modules and enhance the functionality of core. The places where code can be executed are called "hooks" and are defined by a fixed interface.

The basic functionality needed to develop a website is provided by Drupal in its core itself. The user just needs to enable it from the extension. The core modules make it easy to add content, publish them and create pages. The module gives the user full control over how they want their website's functionality.

All the additional functions user need in his site are provided by Drupal in the form of modules. Modules are the elements which provide Drupal with its flexibility and make it one of the best CMS out there.

The user can turn the features and functionality on by installing the module and can turn it off by uninstalling the module but before uninstalling, the user may need to remove data and configuration related to the feature or functionality. Each module that is installed adds to the time needed to generate pages on your site, so it is a good idea to uninstall modules that are not needed.

1.1.2 Drupal Themes

Themes are what make a Drupal website look the way it does. Themers, or theme developers, use HTML, CSS, JavaScript, and other front-end assets in order to implement a design for their site. Each individual theme is a collection of files that define the presentation layer for your application. Themes are generally one of the first places where code is customized for a Drupal site, and are in many cases unique to the specific site they were created for.

Rather than starting from scratch, Drupal themes start from an existing HTML framework and make changes as needed by overriding and changing just the necessary templates. Some themes only need to modify a few select bits, while others may choose to override nearly everything. Either way, if it's HTML, you can change it with a theme.

In order for this to work, every component in Drupal that needs to display something in the browser provides a simple, minimal, HTML template for that element. Whether it's the content of a node, the site logo displayed in the header, or even the header region itself, the required HTML is rendered from a template. These templates can be overridden by a theme in order to change the markup they generate.

Themes are used to:

- Change the HTML markup of anything in Drupal
- Add CSS styles to change the layout, color, or typography on one or more pages
- Use JavaScript to enhance the user experience

Most themes will combine changes to HTML markup with new CSS files that provide the layout and overall graphical treatment of a site, and JavaScript that modifies the ways users interact with the content of the page. Combine all of this, and you can make Drupal look, and feel, like anything you can imagine.

1.2 Task Assigned

1.2.1 Task 1(Custom theme development)

Update the Drupal 8 fossee_istos theme.

1.2.2 Task 2(Site Development)

Update the openplc.fossee.in with the fossee_istos theme.

1.2.3 Task 3(Site Development)

Update the scilab-arduino.fossee.in with the fossee_istos theme.

1.2.4 Task 4(Issue Analyzation)

Analyze fossee.in website for different issues.

1.2.5 Task 5(Custom module development)

Create a forum discussion module for Drupal 7.

1.2.6 Task 6(Custom module development)

Create a forum discussion module for Drupal 8.

Chapter 2

Task1

2.1 Walkthrough

My first task was to work on the fossee_istos theme. The layouts were prebuilt for the theme. But there were lots of issue in the theme and I had to fix all those issues.

The first thing I decided to work on was to analyse the theme for all the scenarios. Then I started noticing all the issues it was having.

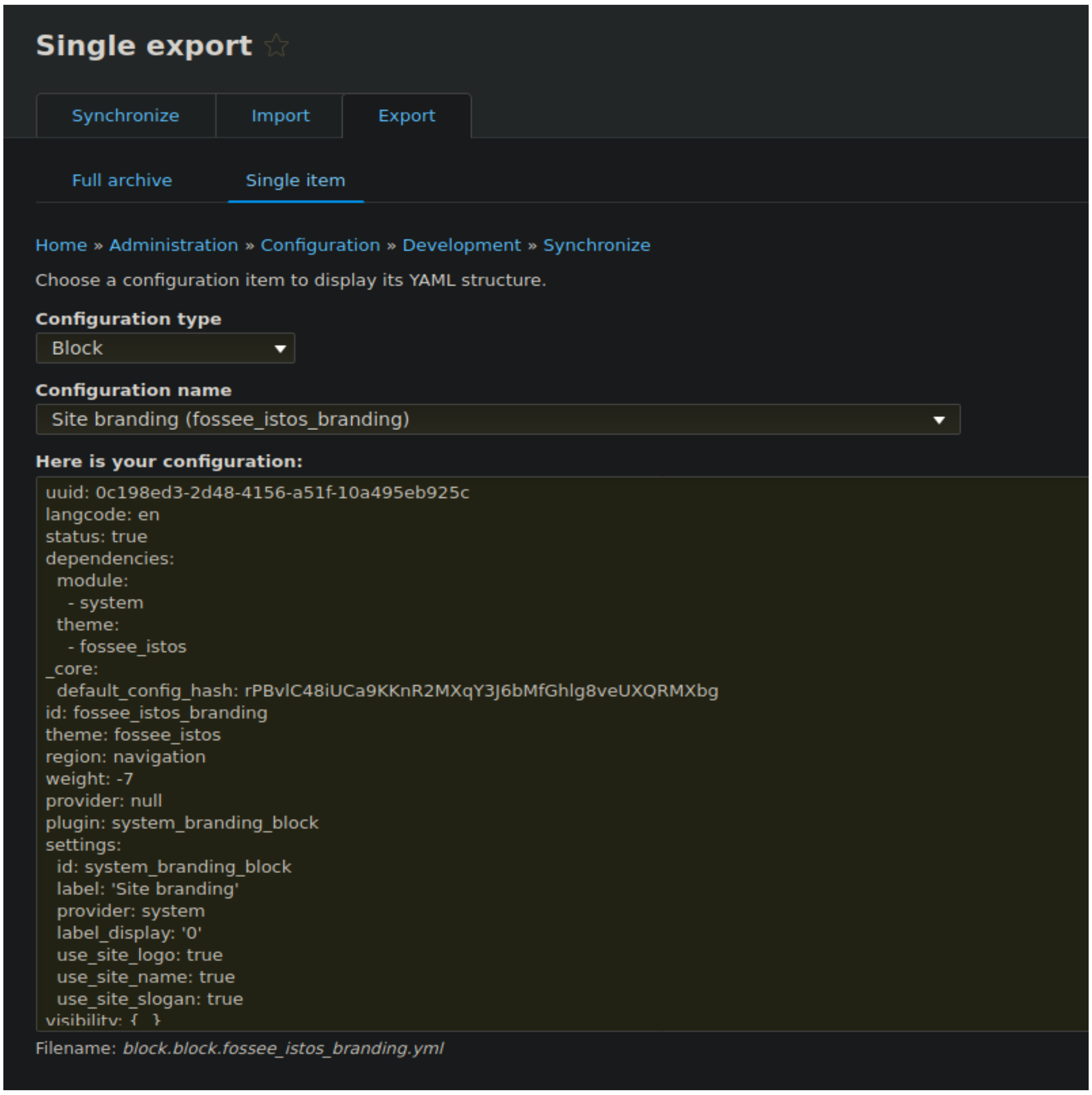
Here are some major issues that I noticed in the fossee_istos theme -

- Some regions of the theme were not positioned correctly.
- The icons were not visible by default in the navbar after installing the theme.
- The footer images were not visible on pages other than homepage.
- Image slider was not displaying by default after the installation.
- The width of different regions were not respective with each other.
- The theme was not perfect for all screen sizes.

So, now after analysing all the issues I started working on each issue one by one.

The first issue was mismatched regions. In Drupal 8 themes to set a block to a specific region/position, we simply need to save the configuration settings of that particular block to our theme directory.

Like if we have a site_branding block in our theme and we want to set it to the top of all other regions, then we need to set its positions manually on our local development environment and then we can save its configuration settings to our theme directory.


Single export
Synchronize
Import
Export
Full archive
Single item
Home » Administration » Configuration » Development » Synchronize

Choose a configuration item to display its YAML structure.

Configuration type: Block

Configuration name: Site branding (fossee_istos_branding)

Here is your configuration:

```
uuid: 0c198ed3-2d48-4156-a51f-10a495eb925c
langcode: en
status: true
dependencies:
  module:
    - system
  theme:
    - fossee_istos
_core:
  default_config_hash: rPBvlC48iUCa9KKnR2MXqY3J6bMfGhlg8veUXQRMXbg
id: fossee_istos_branding
theme: fossee_istos
region: navigation
weight: -7
provider: null
plugin: system_branding_block
settings:
  id: system_branding_block
  label: 'Site branding'
  provider: system
  label_display: '0'
  use_site_logo: true
  use_site_name: true
  use_site_slogan: true
visibility: {  }
```

Filename: *block.block.fossee_istos_branding.yml*

Figure 2.1: Configuration settings for site_branding_block

Now we need to create the following directory:

[your_theme]/config/optional/

In this directory we will be storing all our default configurations for different blocks.

In the Figure 2.1 we can see that at the bottom there is a filename given we can create a file with same filename in the above created directory and there we need to save the content shown in Figure 2.1.

Thats it we have successfully added a default configuration to our block which will be used to position that block correctly on next install.
Similarly I had to add the default configurations for following blocks to position it correctly:

- block.fossee_istos_branding.
- block.fossee_istos_breadcrumbs.
- block.fossee_istos_footer.
- block.fossee_istos_local_tasks.
- block.fossee_istos_main_menu.
- block.fossee_istos_messages.

That's it, the first issue is solved successfully. Now whenever we will install this theme again, all the blocks will be positioned correctly.

The second issue was that the font-awesome icons were not displayed in the site branding block by default. This was a simple issue with the files. In the theme directory the files for font-awesome library was not present, this absence caused this issue. So a quick fix for this issue was to place the required files for font-awesome icons. So I downloaded the font-awesome library files from {URL} and placed it in the following directory:

[**fossee_istos**]**/includes/**

Now the font-awesome icons are successfully displayed in the site_branding block.

The third issue was with the footer image. The image used in the footer block was not loading on pages other than homepage. To fix this issue I had to modify the method of loading images int the template file. Earlier the images were being loaded in this way:

<img src="{{directory}}/image/iitb-logo.png">

A more generalized way to load images can be :

<img src="{{ url('<front>') }}/{{directory}}/image/iitb-logo.png">

Using this method the images were loaded correctly on all pages. Now with all these third issue is also solved now.

The fourth issue was not a blunder. Infact the theme was itself created with this settings. So I had to change this settings from the codebase.
I worked on the templates file to change this settings. Now the flex_slider block will be automatically displayed on the home page whenever the theme is installed.

The fifth issue was with the layout of the theme. The different blocks of the theme were having uncommon width with respect to each other blocks. So for this issue I had to work on css styles of the theme. There is one main stylesheet file which is located under

[**fossee_istos**]**/css/style.css**

So I had to work on this stylesheet file. To fix the layout issue I decided to modify the stylesheets according to viewport width. This approach will always lead to a perfect responsive page.

I modified different rules to work according to the viewport width, as a result the theme now have a perfect layout. All different blocks are sharing same behaviour as other blocks.

Now the last issue with the theme was the unresponsive behaviour and this was the most issue. Noone wants their site to look good on only one screen size. Every one wants their site to look good on all screen sized devices. So I had to fix this issue to make this theme responsive.

The main issue arises on devices with screen size bigger than mobile devices so I had to work for screen bigger than mobile devices. I basically started adding media queries to the the theme.

@media(min-width:768px){}

This is the main media query that I added to the theme and it changed the stylesheet rules for all screen bigger than mobile devices. I started adding different rules for all blocks under this media query and as a result the theme is now fully responsive.

So till now all the issues to the theme were fixed and the theme already started looking good and ready to ship.

2.2 New Features

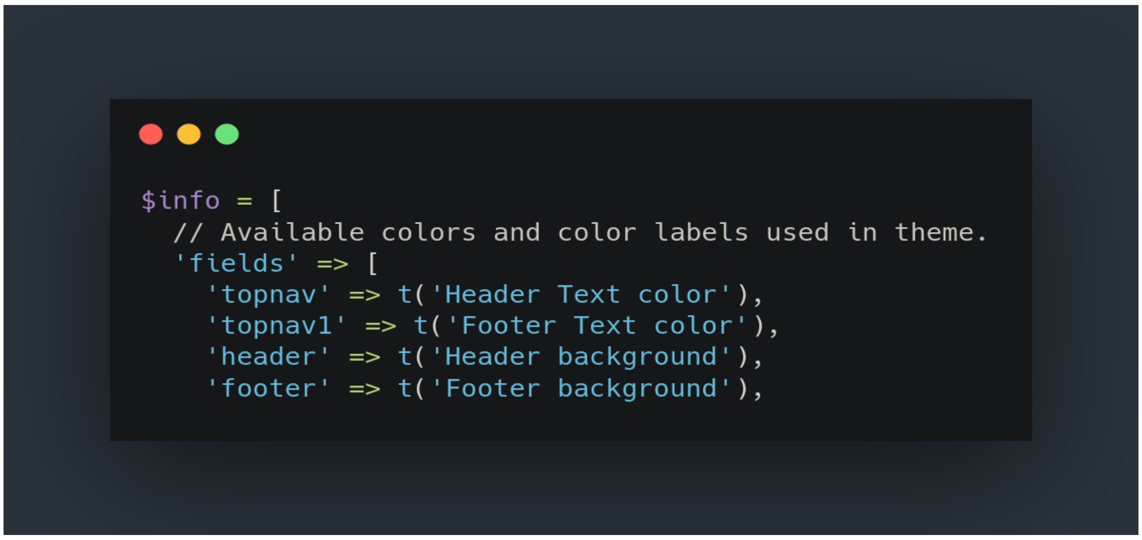
Now I had to work on adding new features to the theme.
Below are the new features that I had to implement -

- Separate color scheme for header and footer background from theme settings.
- Separate color scheme for header and footer color from theme settings.
- Add two more content regions as left_sub_content and right_sub_content.
- Add 3 sub_footer regions by replacing the footer_1 region.
- Modify the overlay for image_slider block.

First and second feature is somewhat similar to each other. Both of these are related to color module build into Drupal 8 core.
I had to modify the prebuilt settings for color module for our theme.
To add a specific color to any block we need to create a info for that block in color.inc file under color directory. After creating the info for that block we need to add the color for that block under schemes array in the same file. Similarly we need to add those colors to preview.css file under the color directory.

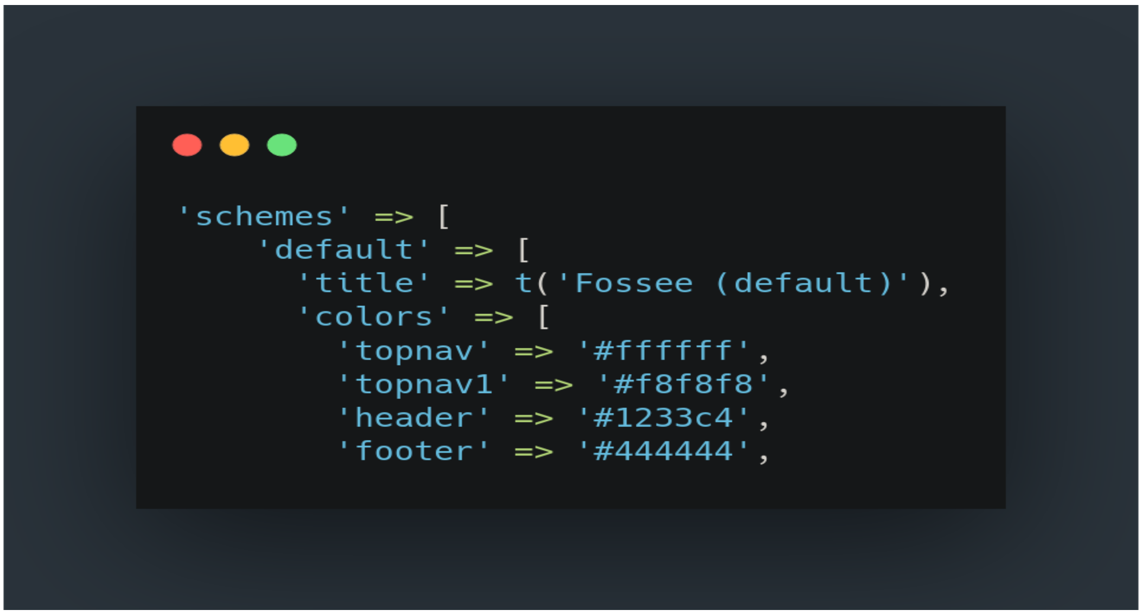
After adding the colors to the respective classes, we now need to edit the preview.js file under the color directory. In the main function of this javascript file we need to find the css class and attach the input color from theme settings to that class.

After doing all this, our theme will start taking effect. Now we can assign different colors for header and footer from our theme settings successfully.



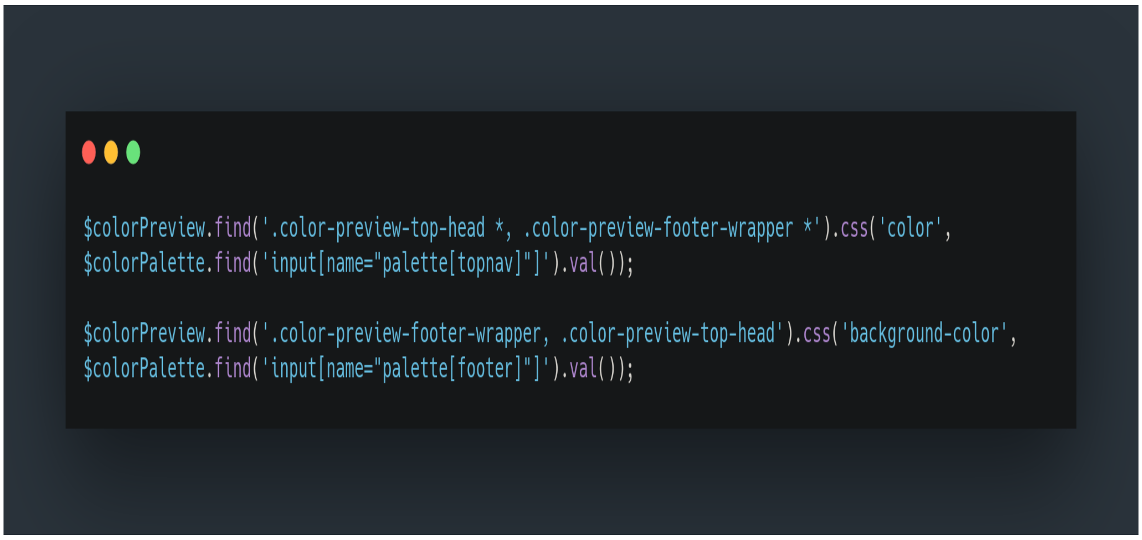
```
$info = [
  // Available colors and color labels used in theme.
  'fields' => [
    'topnav' => t('Header Text color'),
    'topnav1' => t('Footer Text color'),
    'header' => t('Header background'),
    'footer' => t('Footer background'),
```

Figure 2.2: Configuration settings for site_branding_block



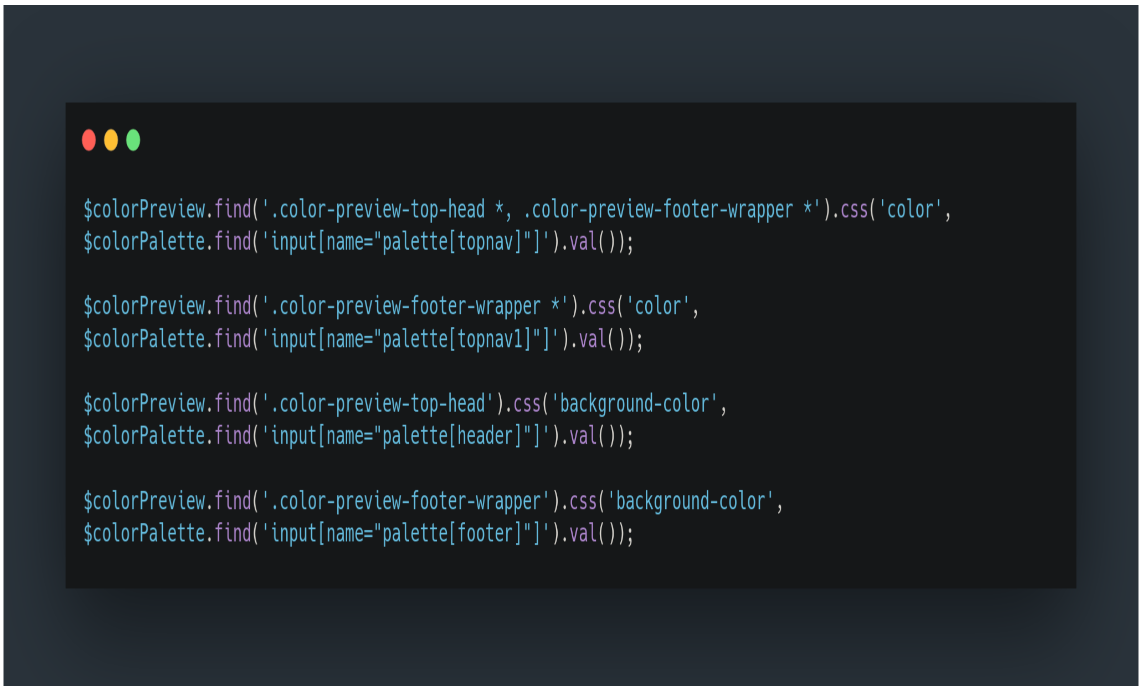
```
'schemes' => [
    'default' => [
      'title' => t('Fossee (default)'),
      'colors' => [
        'topnav' => '#ffffff',
        'topnav1' => '#f8f8f8',
        'header' => '#1233c4',
        'footer' => '#444444',
```

Figure 2.3: Configuration settings for site_branding_block



```
$colorPreview.find('.color-preview-top-head *, .color-preview-footer-wrapper *').css('color',
$colorPalette.find('input[name="palette[topnav]"]').val());

$colorPreview.find('.color-preview-footer-wrapper, .color-preview-top-head').css('background-color',
$colorPalette.find('input[name="palette[footer]"]').val());
```

Figure 2.4: Configuration settings for site_branding_block

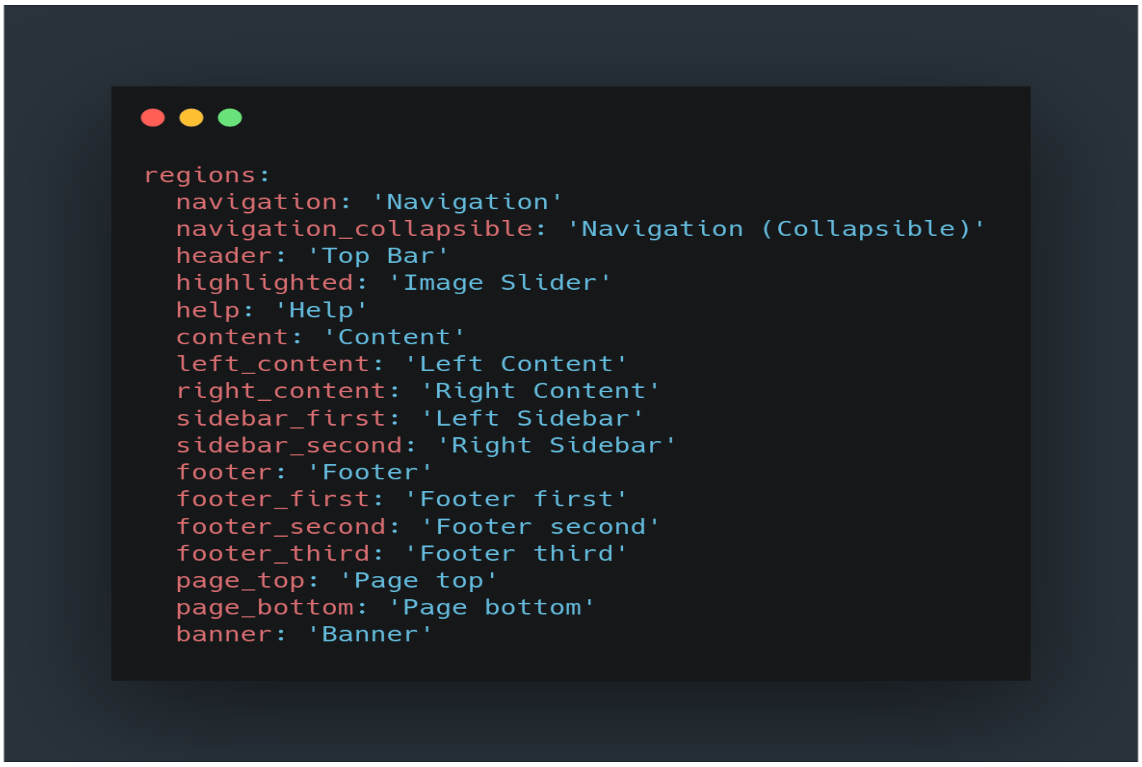


```
$colorPreview.find('.color-preview-top-head *, .color-preview-footer-wrapper *').css('color',
$colorPalette.find('input[name="palette[topnav]"]').val());

$colorPreview.find('.color-preview-footer-wrapper *').css('color',
$colorPalette.find('input[name="palette[topnav1]"]').val());

$colorPreview.find('.color-preview-top-head').css('background-color',
$colorPalette.find('input[name="palette[header]"]').val());

$colorPreview.find('.color-preview-footer-wrapper').css('background-color',
$colorPalette.find('input[name="palette[footer]"]').val());
```

Figure 2.5: Configuration settings for site_branding_block

Third and fourth improvements are also similar to each other. Both the improvements are related to custom region. We need to add two extra content regions below the main content region, as left_sub_content and right_sub_content region. Along with these two regions we also need to add 3 sub_footer region by replacing the footer_1 block.

To add a new custom block to our custom theme firstly we need to edit the **.yml** file for our theme which is **fossee_istos.info.yml**. This file is located in the root directory of our custom theme.



```yaml
regions:
  navigation: 'Navigation'
  navigation_collapsible: 'Navigation (Collapsible)'
  header: 'Top Bar'
  highlighted: 'Image Slider'
  help: 'Help'
  content: 'Content'
  left_content: 'Left Content'
  right_content: 'Right Content'
  sidebar_first: 'Left Sidebar'
  sidebar_second: 'Right Sidebar'
  footer: 'Footer'
  footer_first: 'Footer first'
  footer_second: 'Footer second'
  footer_third: 'Footer third'
  page_top: 'Page top'
  page_bottom: 'Page bottom'
  banner: 'Banner'
```

Figure 2.6: Configuration settings for site_branding_block

I have added two extra sub_content regions as **left_content** and **right_content**. Along with these two regions I also added three sub_footer regions as **footer_first**, **footer_second**, & **footer_third** region.

After adding the regions in the **.yml** file we also need to edit the template files for page. We need to edit the following two files to take our region in effect:

- **page.html.twig**.
- **page–front.html.twig**.

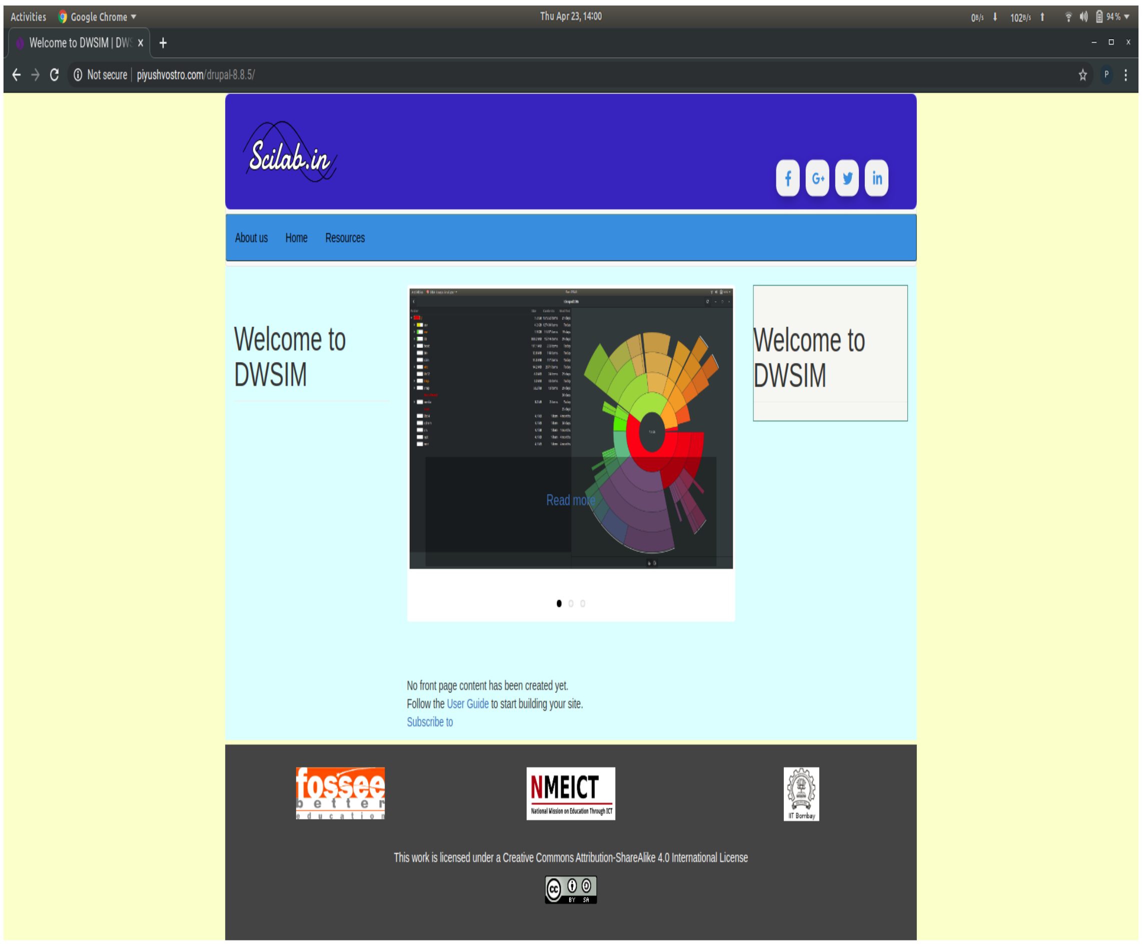
After adding our regions to these files, we can successfully notice the change from the block settings.

So, with all these changes I have successfully added the new regions to our custom theme.

Now the last change that was to be made was to change the overlay for image_slider block. First thing is that we need to remove the link from the overlay. I had to modify the overlay in such a way that only clicking on the overlay, it should redirect to the specified URL. Width of the layout was also not good, so I also had to change the width of the layout. Quick fix to these changes were to edit the respective css files. After modifying the respective files, I changed the behaviour of the block.

So with all these modifications and changes, I successfully completed my first task.

2.3 Final Results



No front page content has been created yet.
Follow the User Guide to start building your site.
Subscribe to

This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License

Figure 2.7: Final fossee_istos theme

Chapter 3

Task2

3.1 Walkthrough

My second task was to update the openplc.fossee.in website with the updated fossee_istos theme. This task was not a time taking task, the site was already built. I only had to update everything with fossee_istos theme.

I began my work by making a fresh Drupal 8 installation on my local development environment. After installing Drupal 8 I installed the fossee_istos theme on my site. Then I started adding all the pages. There were not too many pages on the original openplc.fossee.in website.

I started adding all the different pages to my local site.

3.2 Different Pages

3.2.1 Home

The main page of any website is its homepage. So its the duty of the site admin to make their homepage really standout from other page. But for my case this was not task. I only had to do something like clone of the website. So I just created the same page for my local site but with a different theme.

3.2.2 Resources

For the resources page, there was the same case. I only had to clone the original webpage to my local site but with a different theme.

3.2.3 Downloads

For the Downloads page, there was the same case. I only had to clone the original webpage to my local site but with a different theme.

3.2.4 Fellowship 2020

For the Fellowship 2020, there was the same case. I only had to clone the original webpage to my local site but with a different theme.

3.2.5 Contact Us

For the Contact Us page, there was the same case. I only had to clone the original webpage to my local site but with a different theme.

With all these pages creation, I created the original openplc.fossee.in with the updated fossee_istos theme. Everything was looking good and perfect. Along with all these the site was also fully responsive for all screen size devices.

Chapter 4

Task3

4.1 Walkthrough

My third task was to update the scilab-arduino.fossee.in website with the updated fossee_istos theme. This task was not a time taking task, the site was already built. I only had to update everything with fossee_istos theme.

I began my work by making a fresh Drupal 8 installation on my local development environment. After installing Drupal 8 I installed the fossee_istos theme on my site. Then I started adding all the pages. There were not too many pages on the original openplc.fossee.in website.

I started adding all the different pages to my local site.

4.2 Different Pages

4.2.1 Home

The main page of any website is its homepage. So its the duty of the site admin to make their homepage really standout from other page. But for my case this was not task. I only had to do something like clone of the website. So I just created the same page for my local site but with a different theme.

4.2.2 Resources

For the resources page, there was the same case. I only had to clone the original webpage to my local site but with a different theme.

4.2.3 Downloads

For the Downloads page, there was the same case. I only had to clone the original webpage to my local site but with a different theme.

4.2.4 Forum

For the Forum page, there was the same case. I only had to clone the original webpage to my local site but with a different theme.

4.2.5 Contact Us

For the Contact Us page, there was the same case. I only had to clone the original webpage to my local site but with a different theme.

With all these pages creation, I created the original scilab-arduino.fossee.in with the updated fossee_istos theme. Everything was looking good and perfect. Along with all these the site was also fully responsive for all screen size devices.

Chapter 5

Task4

5.1 Walkthrough

My fourth task was to analyze **fossee.in** website for different issues. This task was a bit related to user_interface and user_experience only. I only had to recognize the layout, style, design issues on the website.

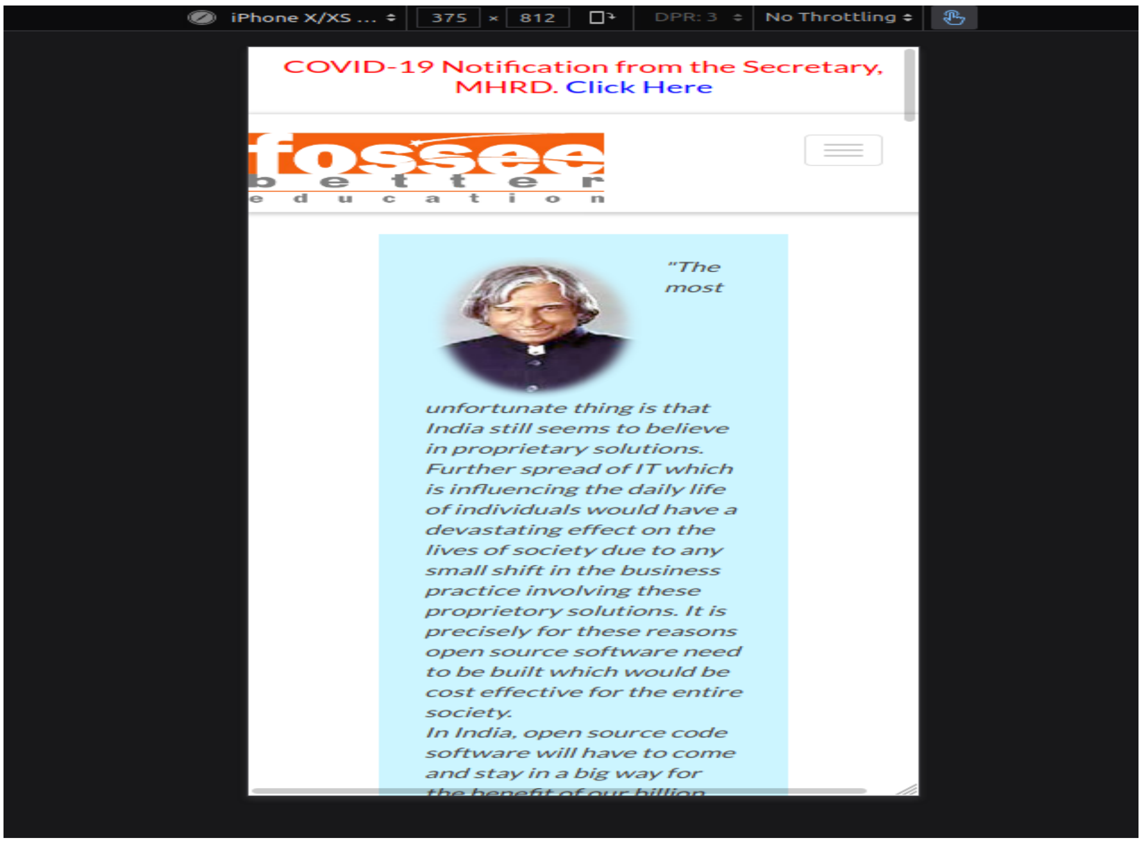
5.2 Analysis

I started working on it by analysing the site for bigger screen sizes. On bigger screen sizes there weren't that much issues to the layouts. Then I started analyzing it for all different screen sized devices, then I started noticing all the major issues to the wesite.

Here is a list of all the issues that I noticed on the website.

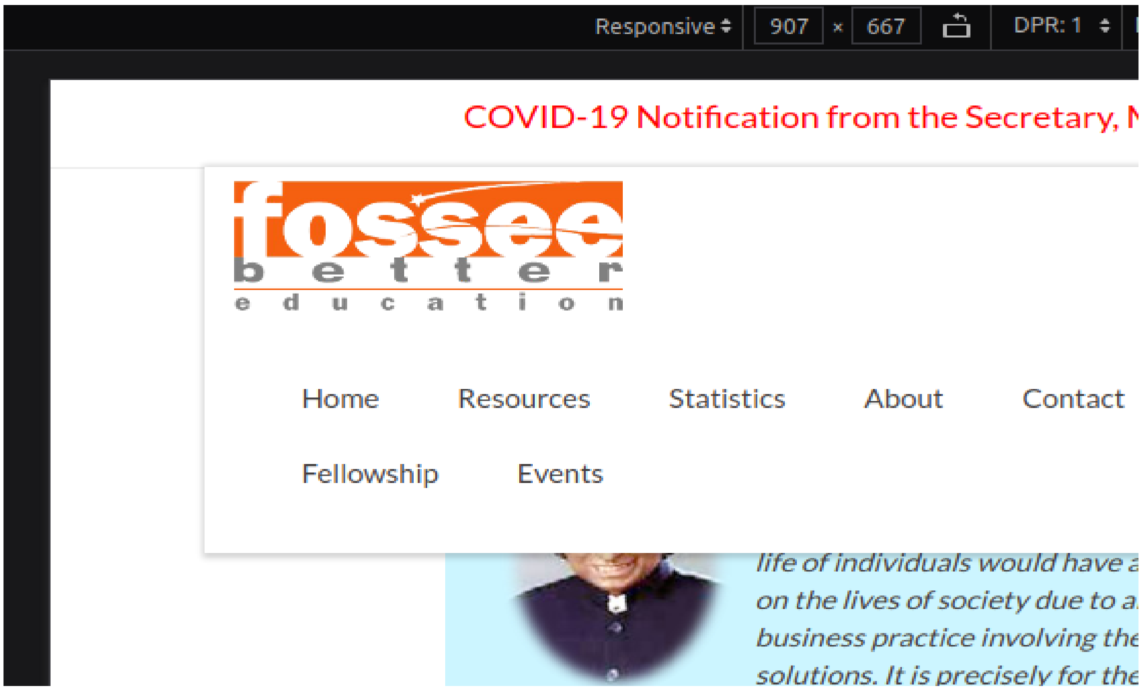
- First of all, the site logo isn't positioned correctly, It doesn't looks perfect on smaller devices.
- The text in the quote div isn't aligned perfectly with respect to the image, on smaller devices it isn't dispalyed perfectly.
- The navbar/sitebranding doesn't have a perfect layout, on smaller devices it overlaps the quote div.

- Some containers don't have perfect padding due to which it gets cut out from the viewport.
- The project wrappers don't have a good layout. On smaller screen the overlay is overflown. Even the overlay text is not displayed completely.
- Tab styles are also not correct, on hovering and on active status the styles applied to these tabs are not correctly configured.
- The layout of the active tab pane in Activities div is also not good. On smaller screen it shrinks too much making the text unreadable.
- The footer layout is also not up to the mark. It isn't a good responsive footer. Inner content doesn't have a good layout.
- Also the overall width of the site is not perfect, it is exceeding the viewport width, which is a really bad UX.



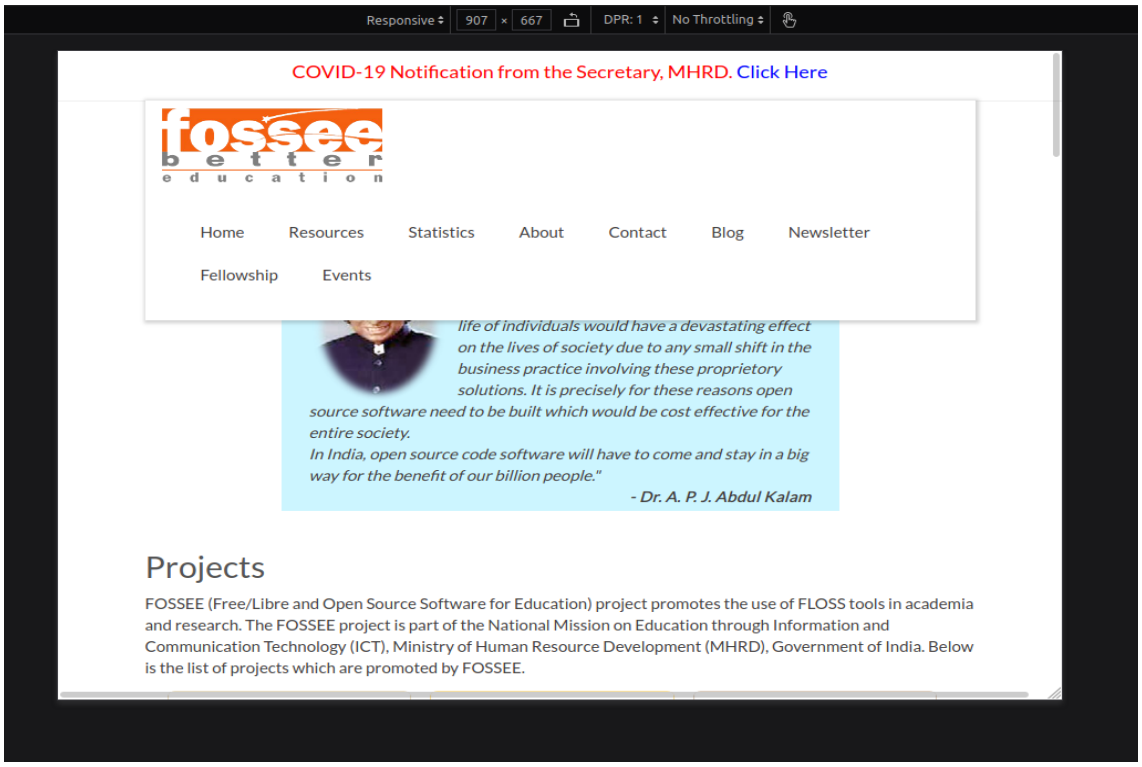
"The most unfortunate thing is that India still seems to believe in proprietary solutions. Further spread of IT which is influencing the daily life of individuals would have a devastating effect on the lives of society due to any small shift in the business practice involving these proprietory solutions. It is precisely for these reasons open source software need to be built which would be cost effective for the entire society.
In India, open source code software will have to come and stay in a big way for the benefit of our billion

Figure 5.1: Wrong logo positioning



life of individuals would have a
on the lives of society due to a
business practice involving the
solutions. It is precisely for the

Figure 5.2: Overlapping elements.

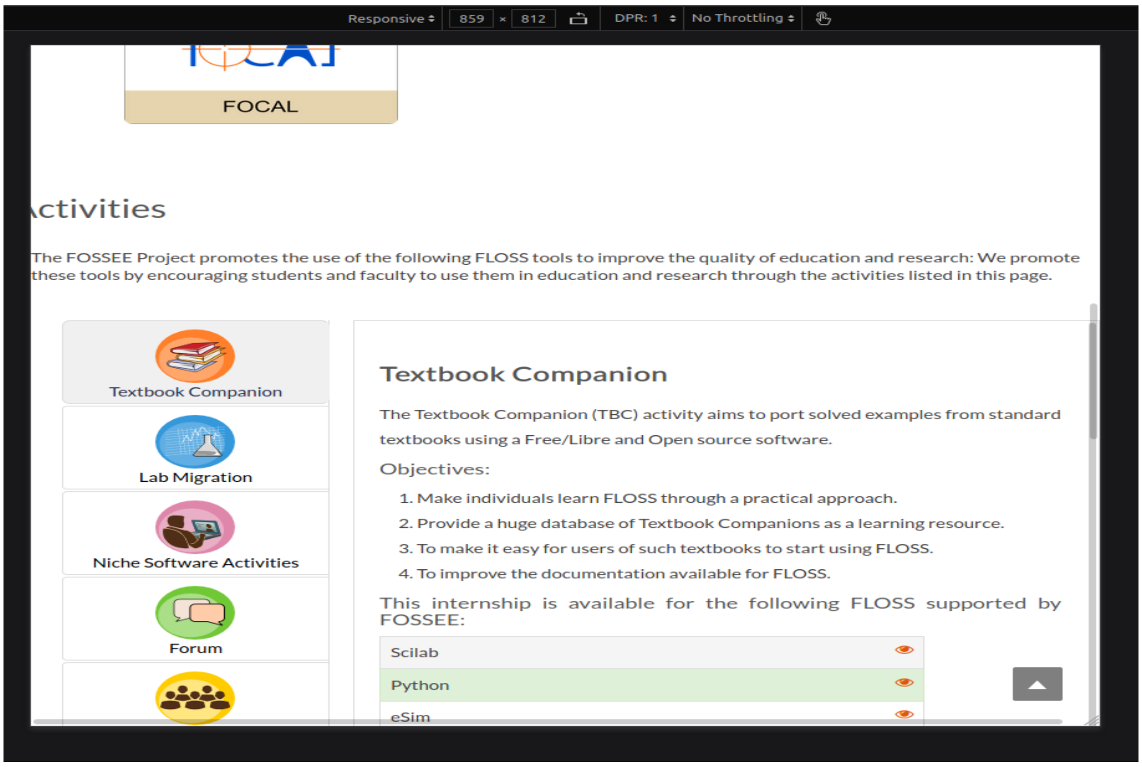


life of individuals would have a devastating effect on the lives of society due to any small shift in the business practice involving these proprietory solutions. It is precisely for these reasons open source software need to be built which would be cost effective for the entire society.
In India, open source code software will have to come and stay in a big way for the benefit of our billion people."
- Dr. A. P. J. Abdul Kalam

Projects

FOSSEE (Free/Libre and Open Source Software for Education) project promotes the use of FLOSS tools in academia and research. The FOSSEE project is part of the National Mission on Education through Information and Communication Technology (ICT), Ministry of Human Resource Development (MHRD), Government of India. Below is the list of projects which are promoted by FOSSEE.

Figure 5.3: Overlapping elements



ctivities

The FOSSEE Project promotes the use of the following FLOSS tools to improve the quality of education and research: We promote these tools by encouraging students and faculty to use them in education and research through the activities listed in this page.

Textbook Companion

The Textbook Companion (TBC) activity aims to port solved examples from standard textbooks using a Free/Libre and Open source software.

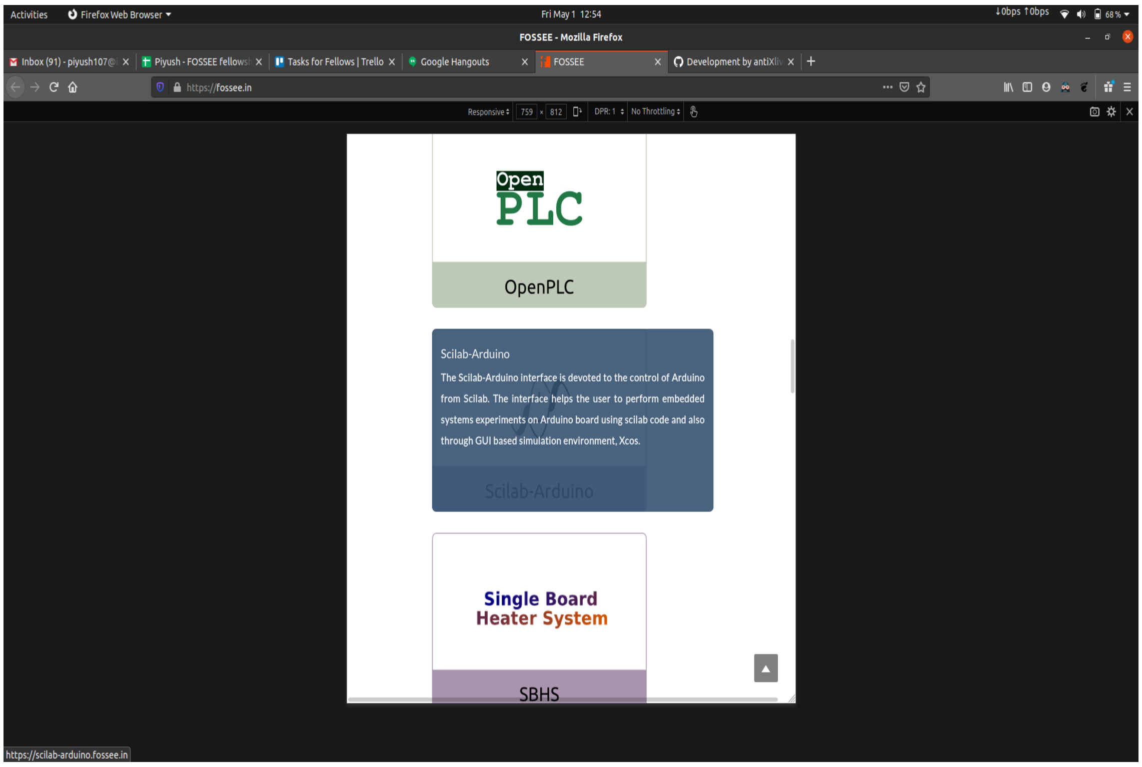
Objectives:

1. Make individuals learn FLOSS through a practical approach.
2. Provide a huge database of Textbook Companions as a learning resource.
3. To make it easy for users of such textbooks to start using FLOSS.
4. To improve the documentation available for FLOSS.

This internship is available for the following FLOSS supported by FOSSEE:

Scilab
Python
eSim

Figure 5.4: Incorrect width and padding of elements

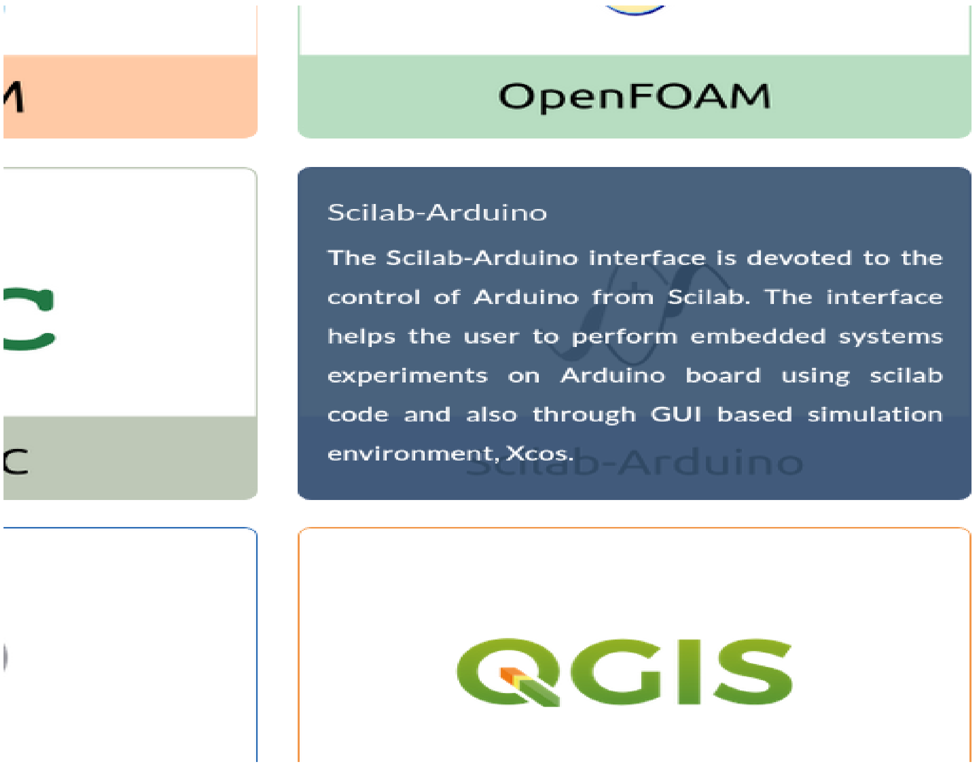


Scilab-Arduino

The Scilab-Arduino interface is devoted to the control of Arduino from Scilab. The interface helps the user to perform embedded systems experiments on Arduino board using scilab code and also through GUI based simulation environment, Xcos.

Scilab-Arduino

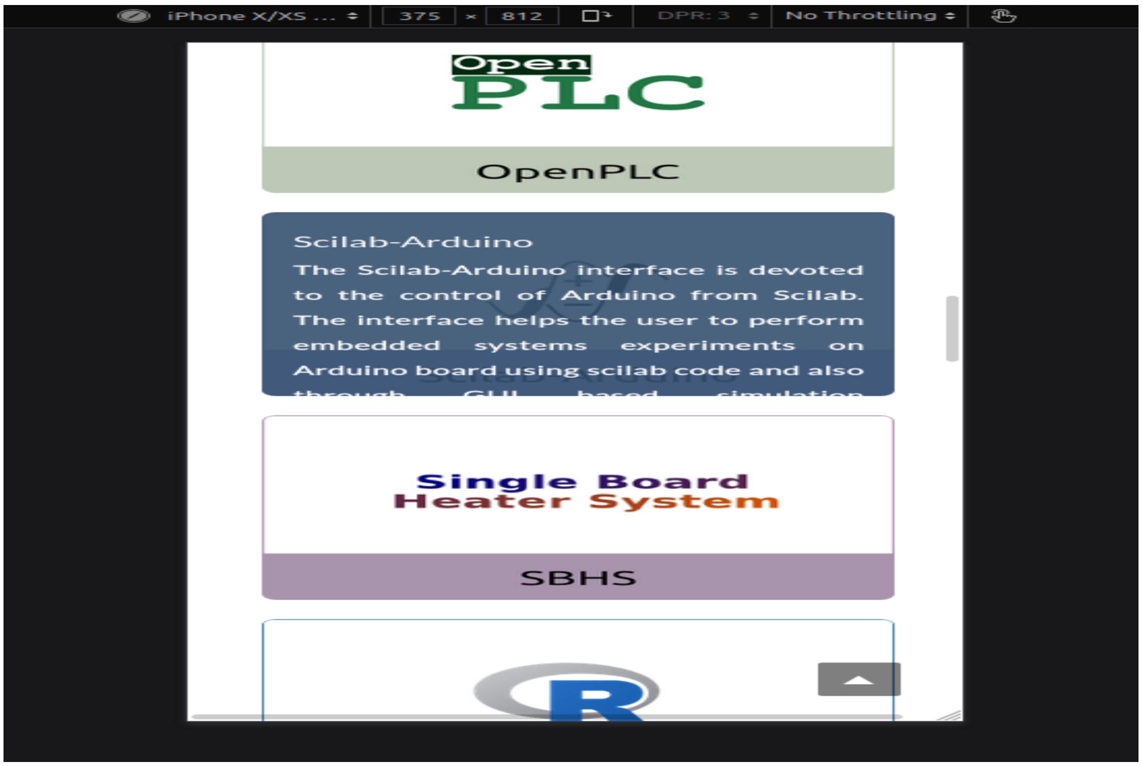
Figure 5.5: Bad width of overlays.



Scilab-Arduino

The Scilab-Arduino interface is devoted to the control of Arduino from Scilab. The interface helps the user to perform embedded systems experiments on Arduino board using scilab code and also through GUI based simulation environment, Xcos.

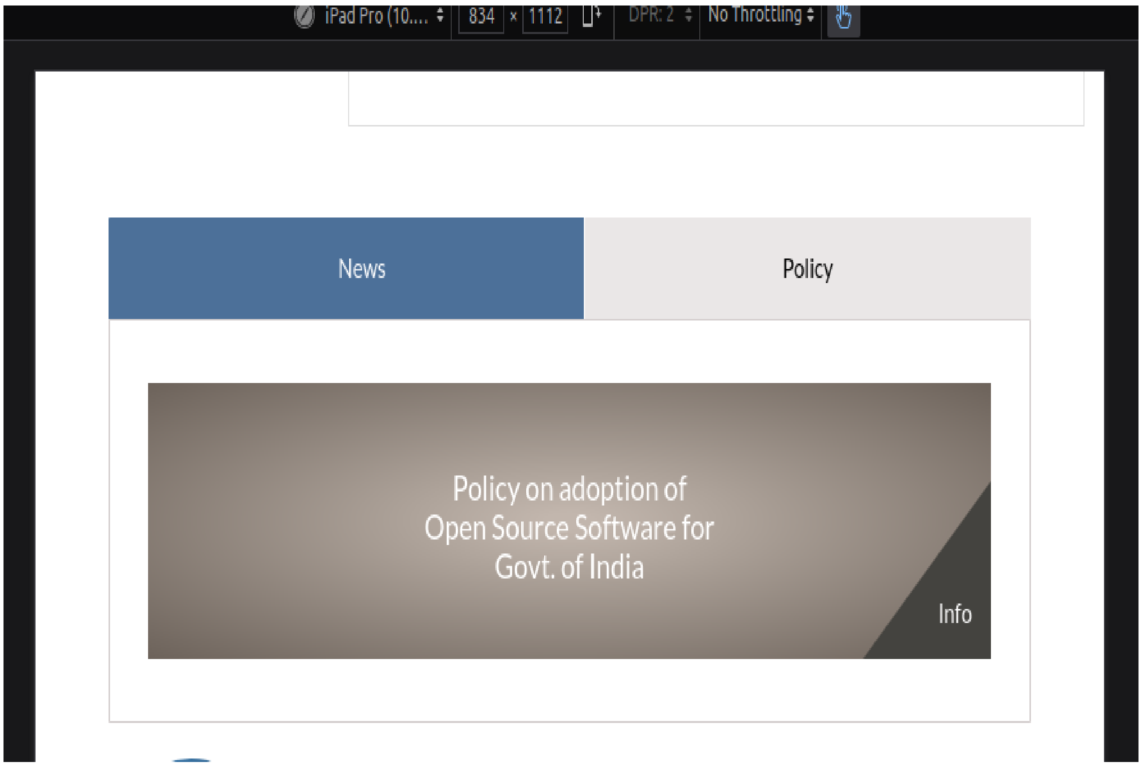
Figure 5.6: Complete text on bigger screen



Scilab-Arduino

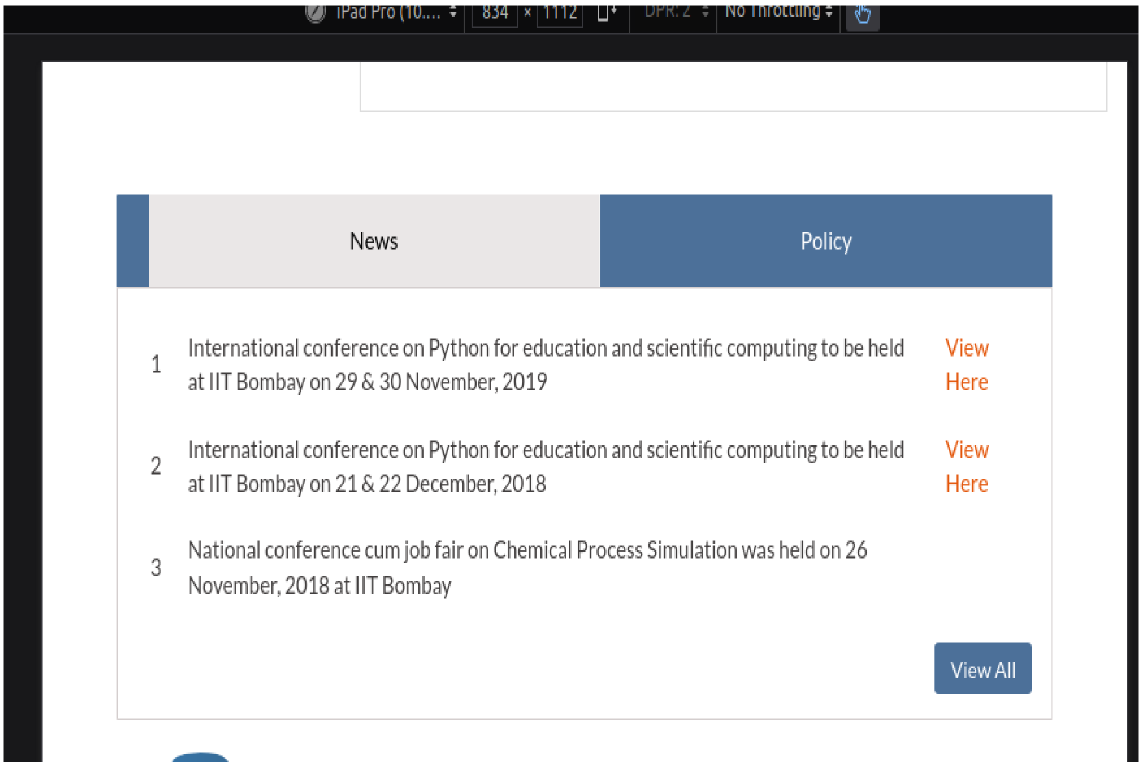
The Scilab-Arduino interface is devoted to the control of Arduino from Scilab. The interface helps the user to perform embedded systems experiments on Arduino board using scilab code and also

Figure 5.7: Incomplete text on smaller screen



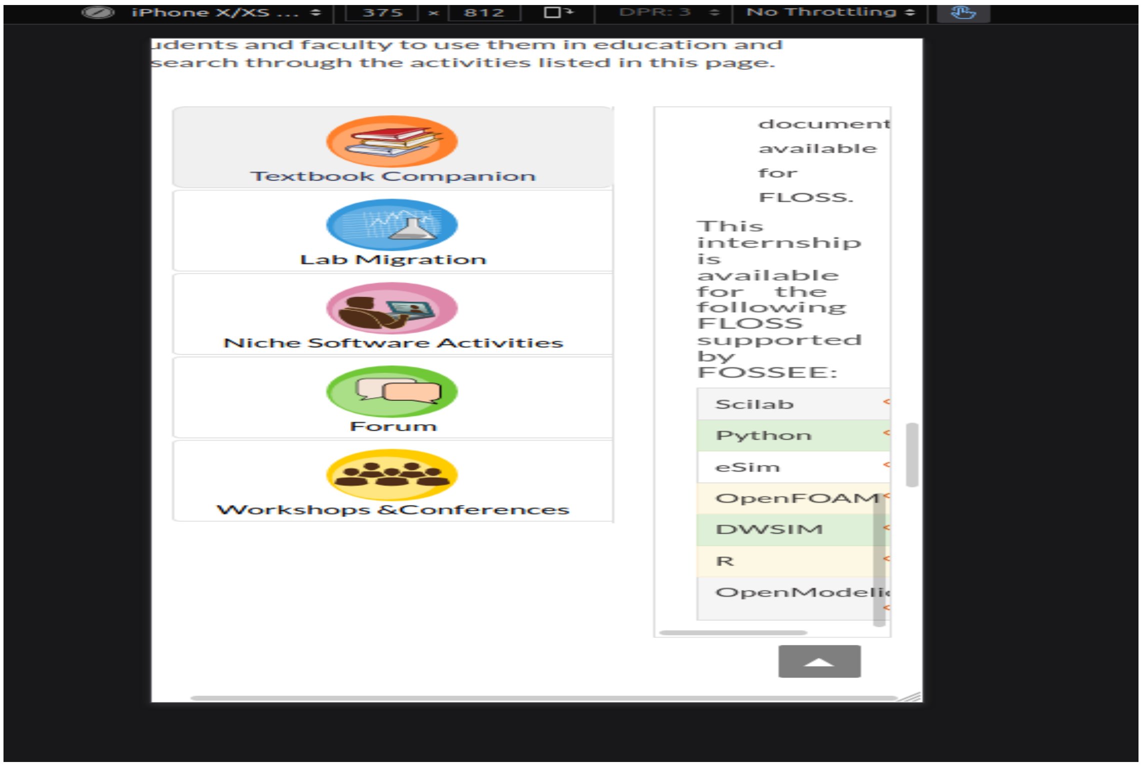
Policy on adoption of
Open Source Software for
Govt. of India

Figure 5.8: Bad hover styles



1	International conference on Python for education and scientific computing to be held at IIT Bombay on 29 & 30 November, 2019	View Here
2	International conference on Python for education and scientific computing to be held at IIT Bombay on 21 & 22 December, 2018	View Here
3	National conference cum job fair on Chemical Process Simulation was held on 26 November, 2018 at IIT Bombay	

Figure 5.9: Bad hover styles


iPhone X/XS ... 375 × 812 DPR: 3 No Throttling

udents and faculty to use them in education and
search through the activities listed in this page.

Textbook Companion
Lab Migration
Niche Software Activities
Forum
Workshops &Conferences

document
available
for
FLOSS.
This
internship
is
available
for the
following
FLOSS
supported
by
FOSSEE:

Scilab
Python
eSim
OpenFOAM
DWSIM
R
OpenModeli

Figure 5.10: Bad layout

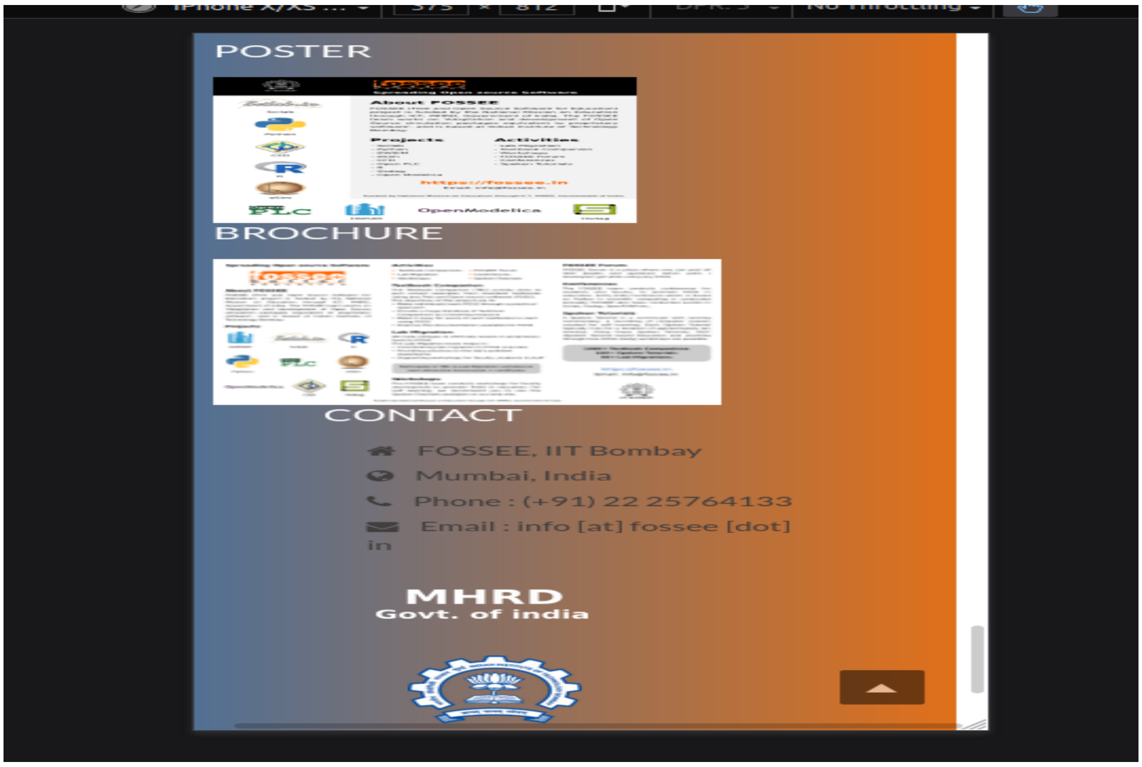

POSTER
BROCHURE
CONTACT
FOSSEE, IIT Bombay
Mumbai, India
Phone : (+91) 22 25764133
Email : info [at] fossee [dot] in
MHRD
Govt. of india

Figure 5.11: Bad overall width of site.

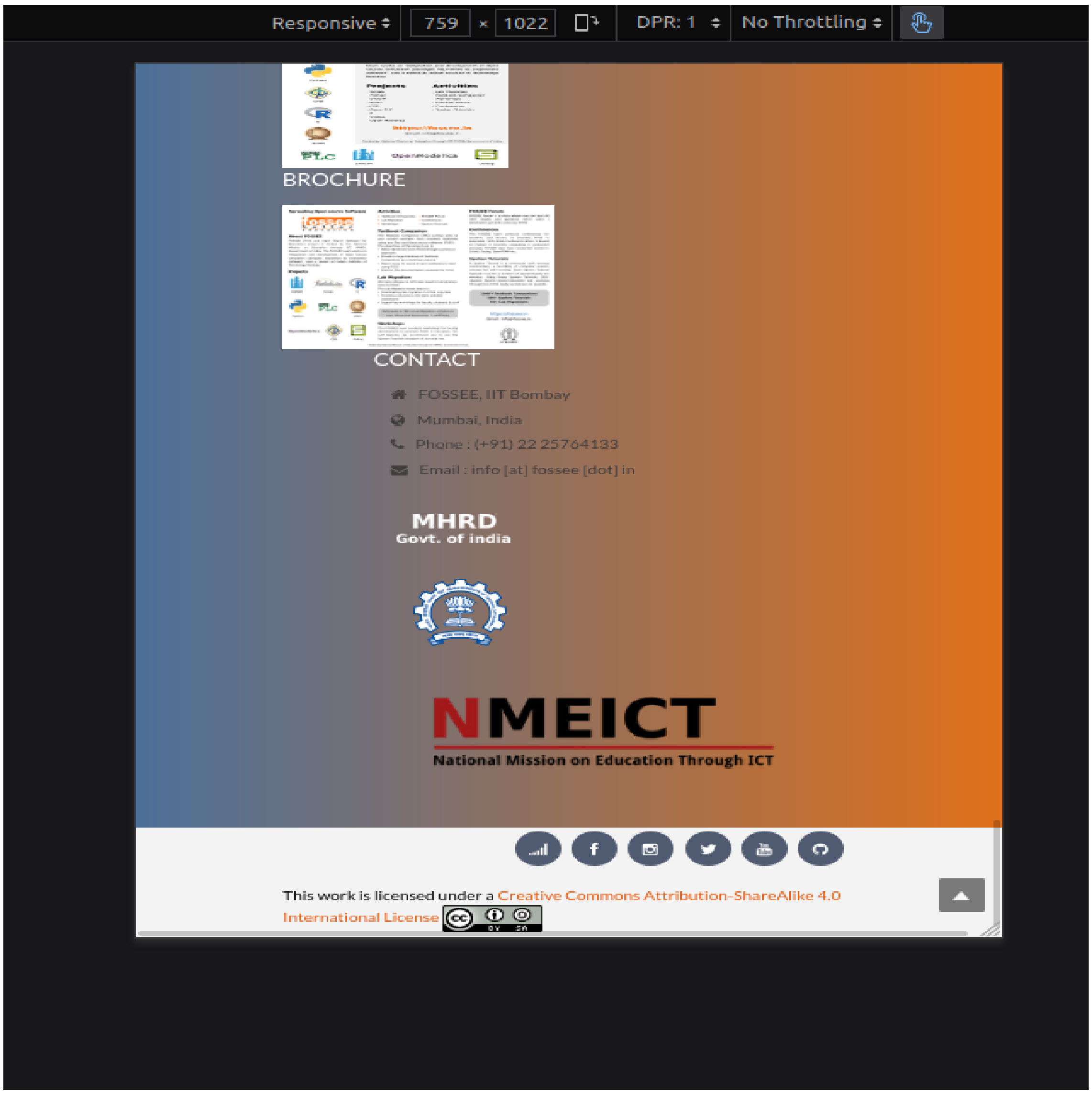

Responsive
759 × 1022
DPR: 1
No Throttling
BROCHURE
CONTACT
FOSSEE, IIT Bombay
Mumbai, India
Phone : (+91) 22 25764133
Email : info [at] fossee [dot] in
MHRD
Govt. of india
NMEICT
National Mission on Education Through ICT
This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License

Figure 5.12: Footer elements don't have a good alignment

Chapter 6

Task5

6.1 Walkthrough

My fifth task was to create a forum discussion module for Drupal 7. According to me, this task was something more like a realistic and real world task. I had to develop a custom forum discussion module from scratch.

6.2 Database

The first thing I decided to work on was to design database for our module. According to a basic functionality of a normal or standard forum discussion module, I began designing my database. There were two main tables that I decided to create.

The first table I created, was to store all the different comments of different users.
The second table I created, was to store all the different replies given to a specific comment according to their ids.
This is the database that I created to implement in our forum_discussion module.

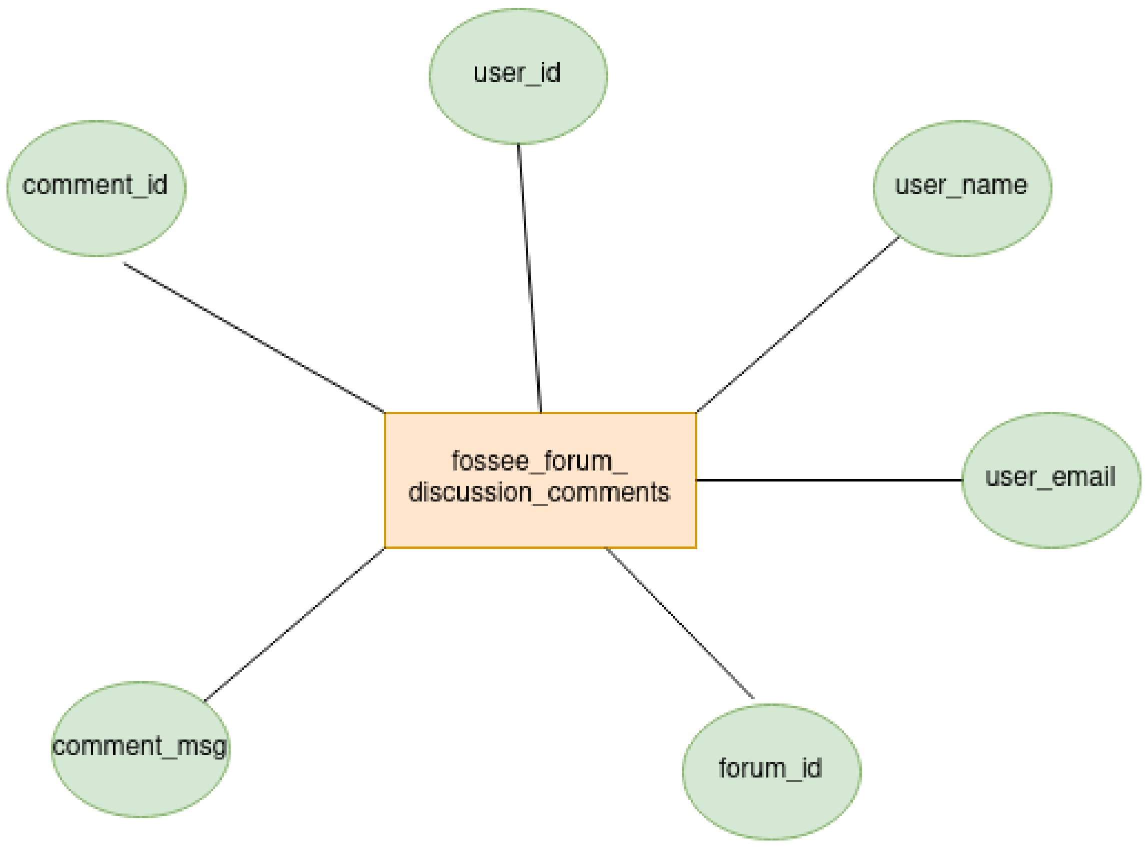


Figure 6.1: ER diagram for our custom module

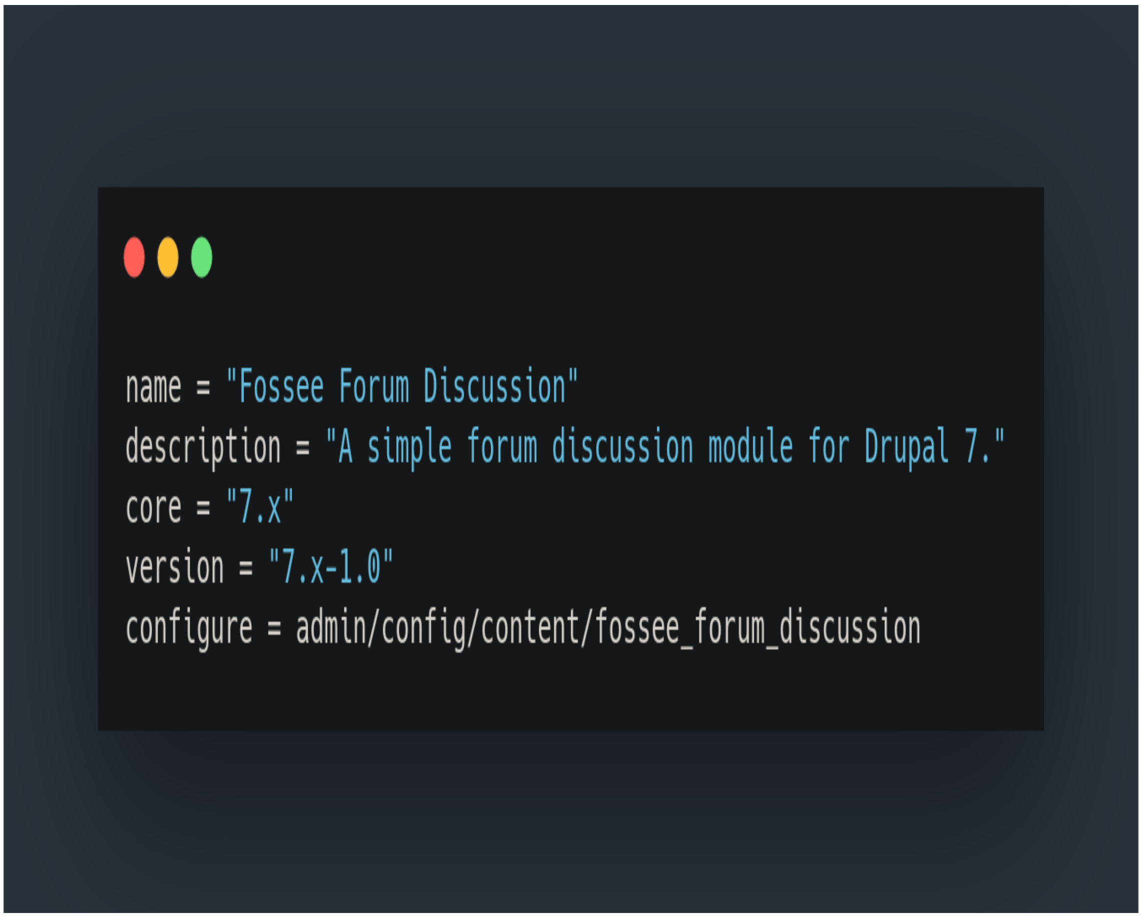
6.3 File Structure

The first task of any development is to have a proper file structure. For our forum discussion module we also need a proper file structure. So, Let's start with file structure for our forum discussion module.

6.3.1 fossee_forum_discussion.info

Drupal 7 uses .info files to store metadata about themes and modules.
For modules, the .info file is used for:

- rendering information on the Drupal Web GUI administration pages;
- providing criteria to control module activation and deactivation;
- notifying Drupal about the existence of a module;
- specifying the module's dependencies on other Drupal projects
- general administrative purposes in other contexts.



```
name = "Fossee Forum Discussion"
description = "A simple forum discussion module for Drupal 7."
core = "7.x"
version = "7.x-1.0"
configure = admin/config/content/fossee_forum_discussion
```

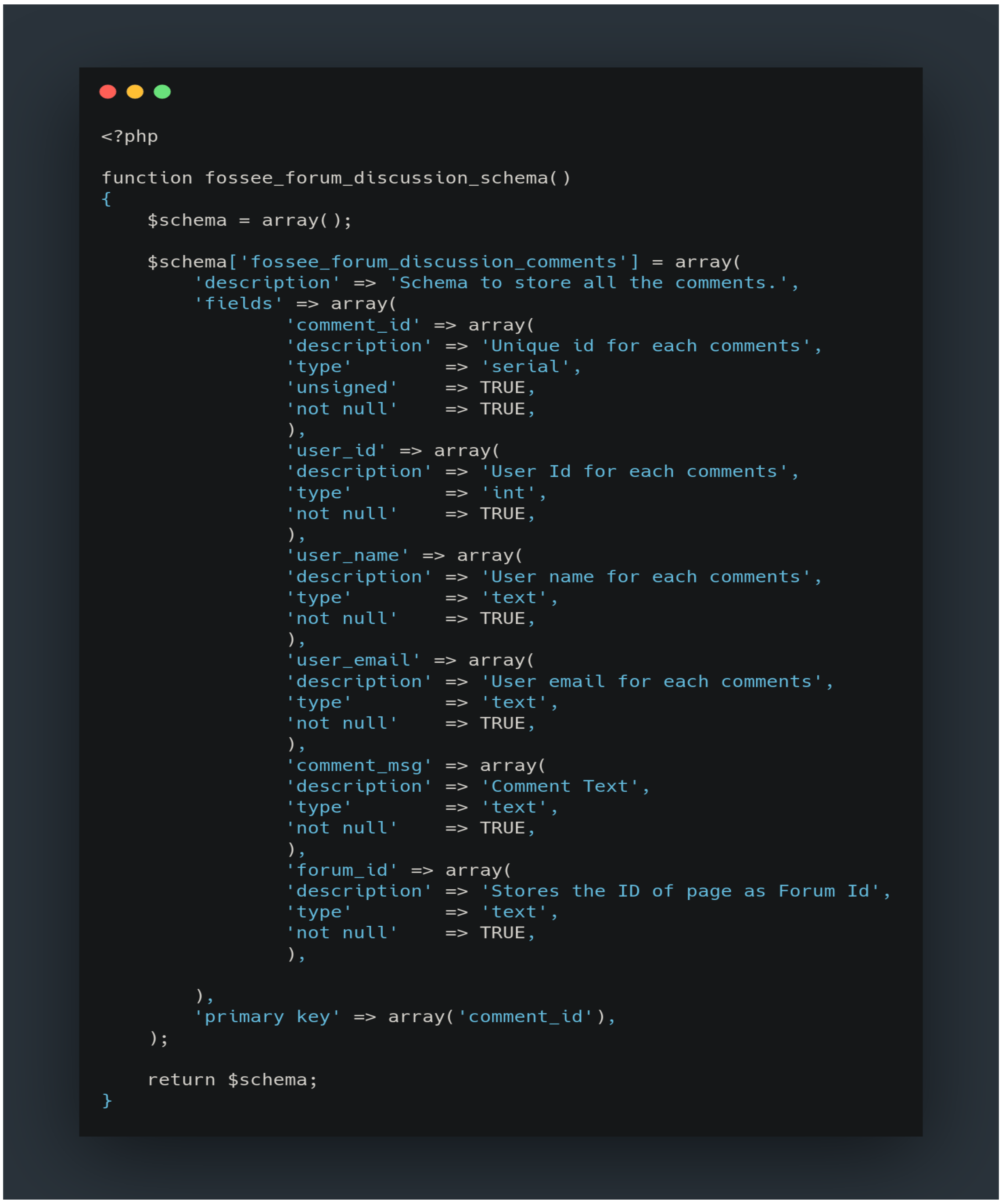
Figure 6.2: Info file for forum_discussion module

6.3.2 fossee_forum_discussion.install

Drupal 7 uses .install files to create database tables and fields, and to insert data. Drupal 7 .install files can also provide updates to change the database structure and content. Drupal 7 also uses .install files and includes the Field module in core. Field provides a different way to send data to nodes than in Drupal 6 and could replace some .install files.

6.3.3 fossee_forum_discussion.module

A .module file is what Drupal uses for all basic functionalities. All the main codes are written here. All different functions and templates are getting in action from this module file.



```php
<?php

function fossee_forum_discussion_schema()
{
    $schema = array();

    $schema['fossee_forum_discussion_comments'] = array(
        'description' => 'Schema to store all the comments.',
        'fields' => array(
                'comment_id' => array(
                'description' => 'Unique id for each comments',
                'type'        => 'serial',
                'unsigned'    => TRUE,
                'not null'    => TRUE,
                ),
                'user_id' => array(
                'description' => 'User Id for each comments',
                'type'        => 'int',
                'not null'    => TRUE,
                ),
                'user_name' => array(
                'description' => 'User name for each comments',
                'type'        => 'text',
                'not null'    => TRUE,
                ),
                'user_email' => array(
                'description' => 'User email for each comments',
                'type'        => 'text',
                'not null'    => TRUE,
                ),
                'comment_msg' => array(
                'description' => 'Comment Text',
                'type'        => 'text',
                'not null'    => TRUE,
                ),
                'forum_id' => array(
                'description' => 'Stores the ID of page as Forum Id',
                'type'        => 'text',
                'not null'    => TRUE,
                ),

        ),
        'primary key' => array('comment_id'),
    );

    return $schema;
}
```

Figure 6.3: Schema for comment table

6.3.4 fossee_forum_discussion.tpl.php

I have created a separate template files to render all comments and replies. In this template file I have written code in Javascript and PHP to achieve all the access to make the functionalities easier to implement.

6.3.5 fossee_forum_discussion_email.inc

I have created a separate email file to implement hook_mail function. This file contains all the email templates along with the email message. We can easily edit all the email templates from here according to our needs.

6.3.6 fossee_forum_discussion_helpers.php

I have created a separate php file to write different helper functions that are needed through out our module. This php file makes our module a bit more modular.
We have access to everything means every resources in this helpers file. This way we can easily use the database api of drupal to store and retrieve all the data related to our module in this php file.
I have also implemented custom functions to call drupal_mail to trigger emails on various events.

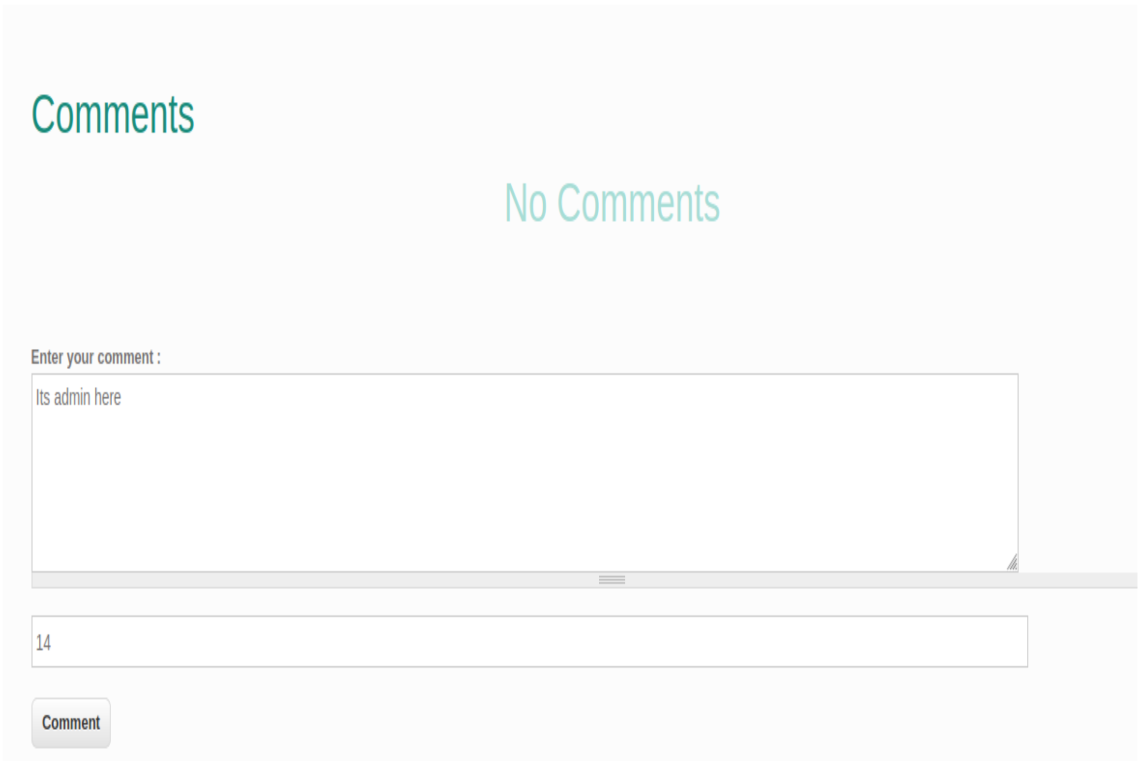
6.4 Module Functionalities

In our custom module, I basically had to implement forum_discussion functionality. But there were also other functionalities that I had to implement.
Here is a list of all those basic functionalities that need to be there in our custom module:

- There should be separate forums for different pages according to the page_id.
- Only authorised user should be able to post a comment or reply.
- Unauthorised users should be redirected to login page on post actions.
- Multiple spaces should be filtered out from the comment or reply.
- There should be a limit for minimum and maximum length of comment and reply messages.
- There should be an option to set these limits from module configuration.
- A custom mail should be triggered to comment author when they posts a new comment.
- A custom mail should be triggered to reply author when a user posts a new reply.

- A custom mail should be triggered to forum members when a user posts a new reply.
- All emails should be sent to the emails mentioned in the bcc and cc variables in module configuration.

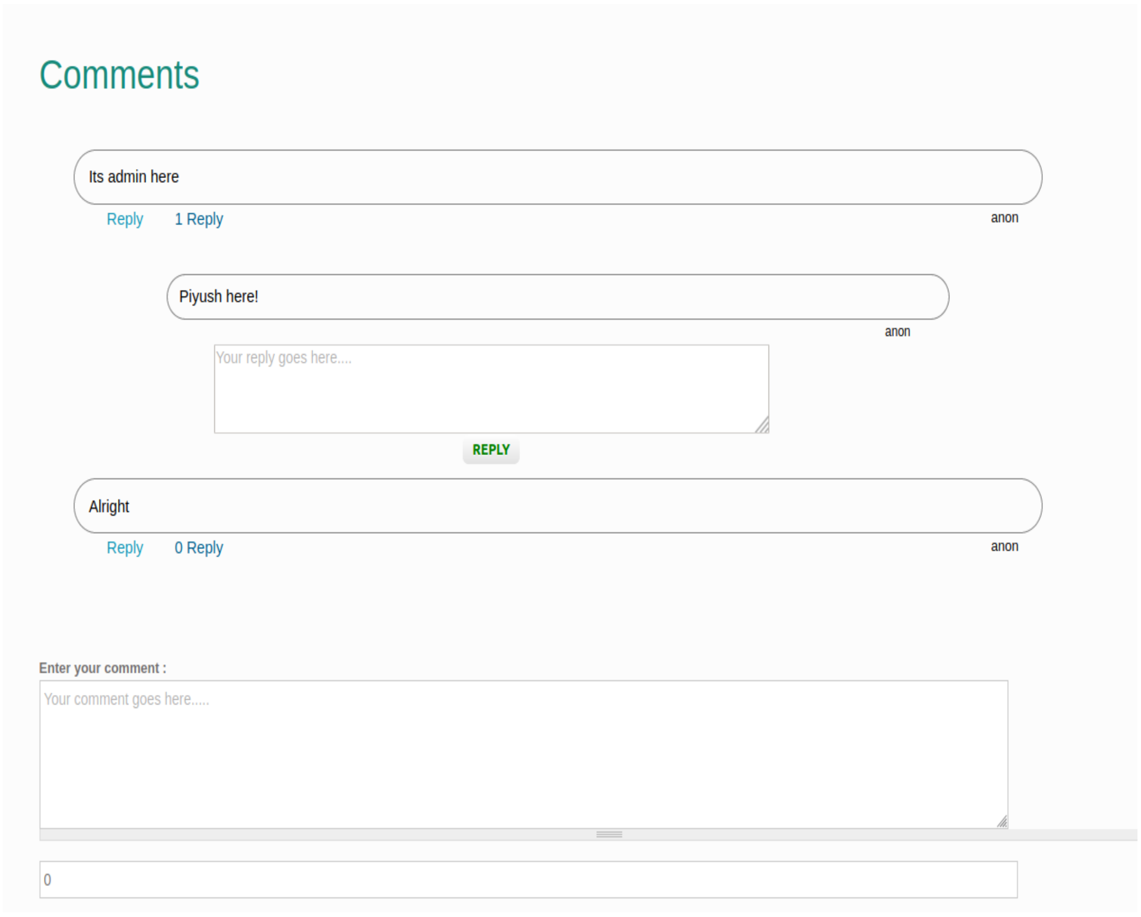
6.5 Final Results



Its admin here

14

Figure 6.4: No comments view & message count



Comments

Its admin here

Reply 1 Reply anon

Piyush here!

anon

Your reply goes here....

REPLY

Alright

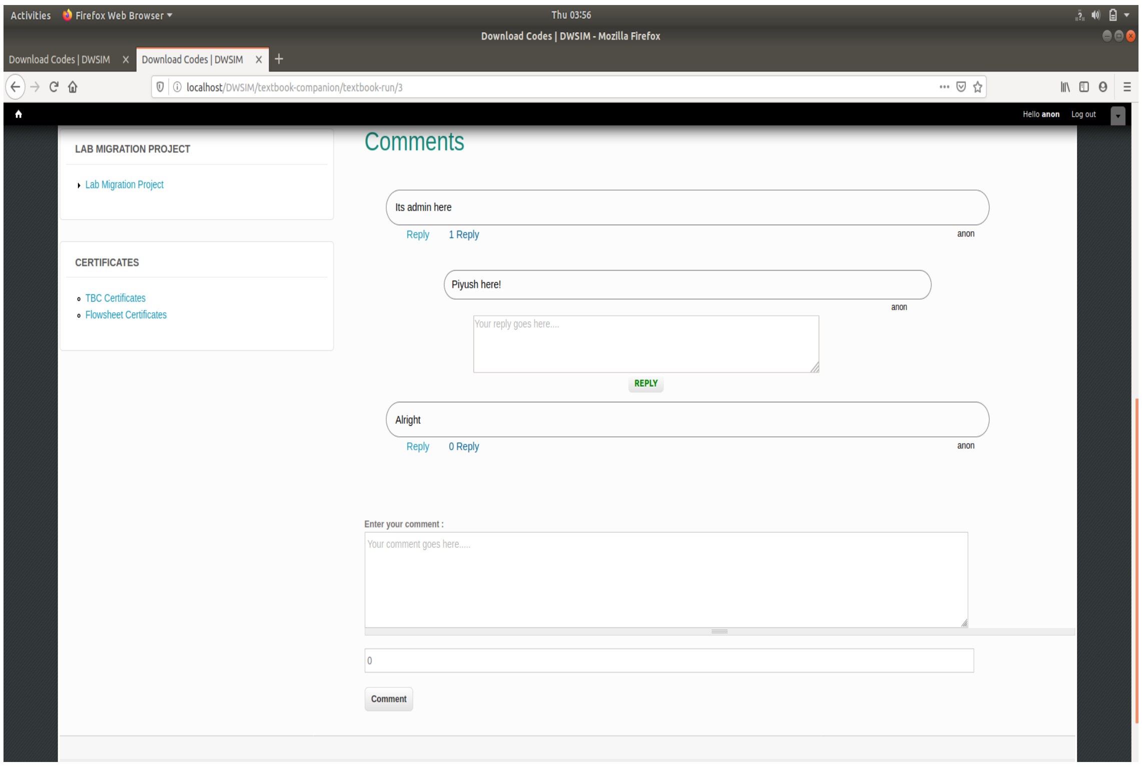
Reply 0 Reply anon

Enter your comment :

Your comment goes here.....

0

Figure 6.5: Expanded view



Comments

Its admin here

Reply 1 Reply anon

Piyush here!

anon

Your reply goes here....

REPLY

Alright

Reply 0 Reply anon

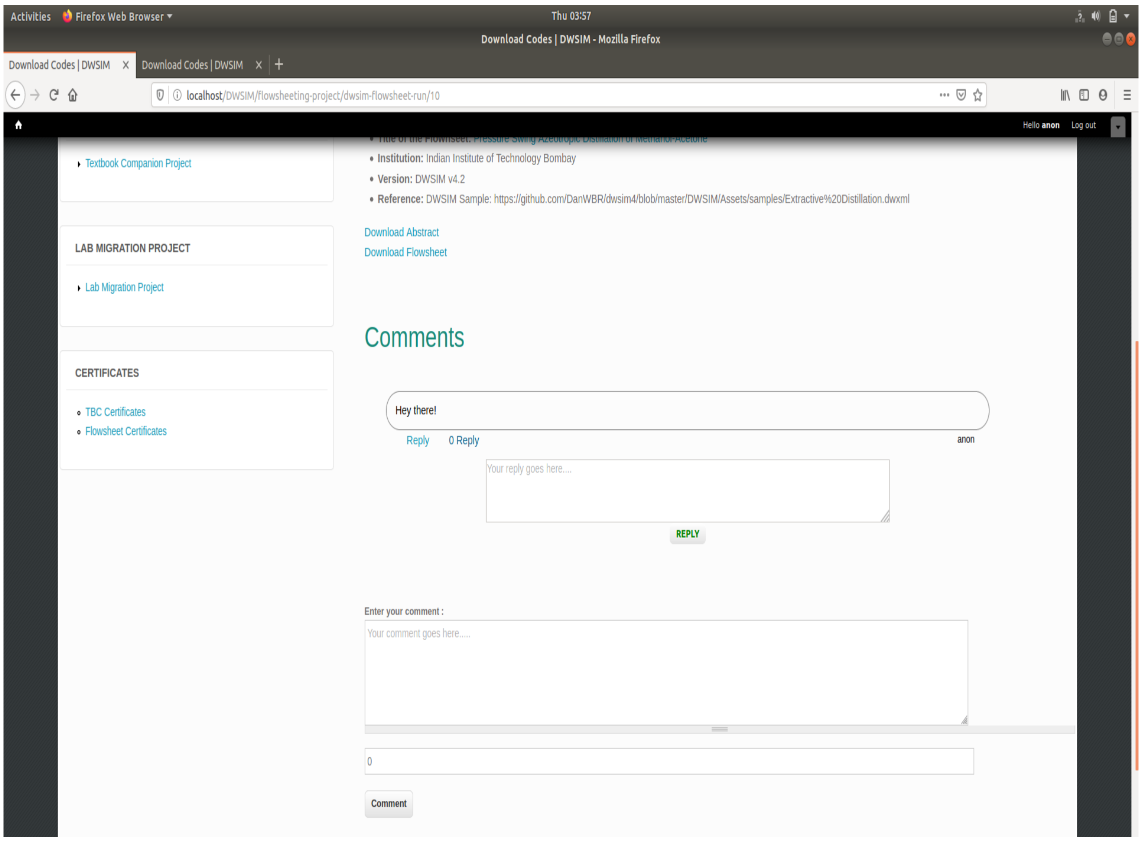
Enter your comment :

Your comment goes here.....

0

Comment

Figure 6.6: Module implanted on flowsheeting category



- Title of the Flowsheet: Pressure Swing Azeotropic Distillation of Methanol-Acetone
- Institution: Indian Institute of Technology Bombay
- Version: DWSIM v4.2
- Reference: DWSIM Sample: https://github.com/DanWBR/dwsim4/blob/master/DWSIM/Assets/samples/Extractive%20Distillation.dwxml

Download Abstract
Download Flowsheet

Comments

Hey there!

Reply 0 Reply anon

Your reply goes here....

REPLY

Enter your comment :

Your comment goes here.....

0

Comment

Figure 6.7: Module implanted on textbook-companion category

Chapter 7

Task6

7.1 Walkthrough

My sixth and last task was to create a forum discussion module for Drupal 8.

7.2 Database

The first thing I decided to work on was to design database for our module. According to a basic functionality of a normal or standard forum discussion module, I began designing my database. There were two main tables that I decided to create.

The first table I created, was to store all the different comments of different users.
The second table I created, was to store all the different replies given to a specific comment according to their ids.
This is the database that I created to implement in our forum_discussion module.

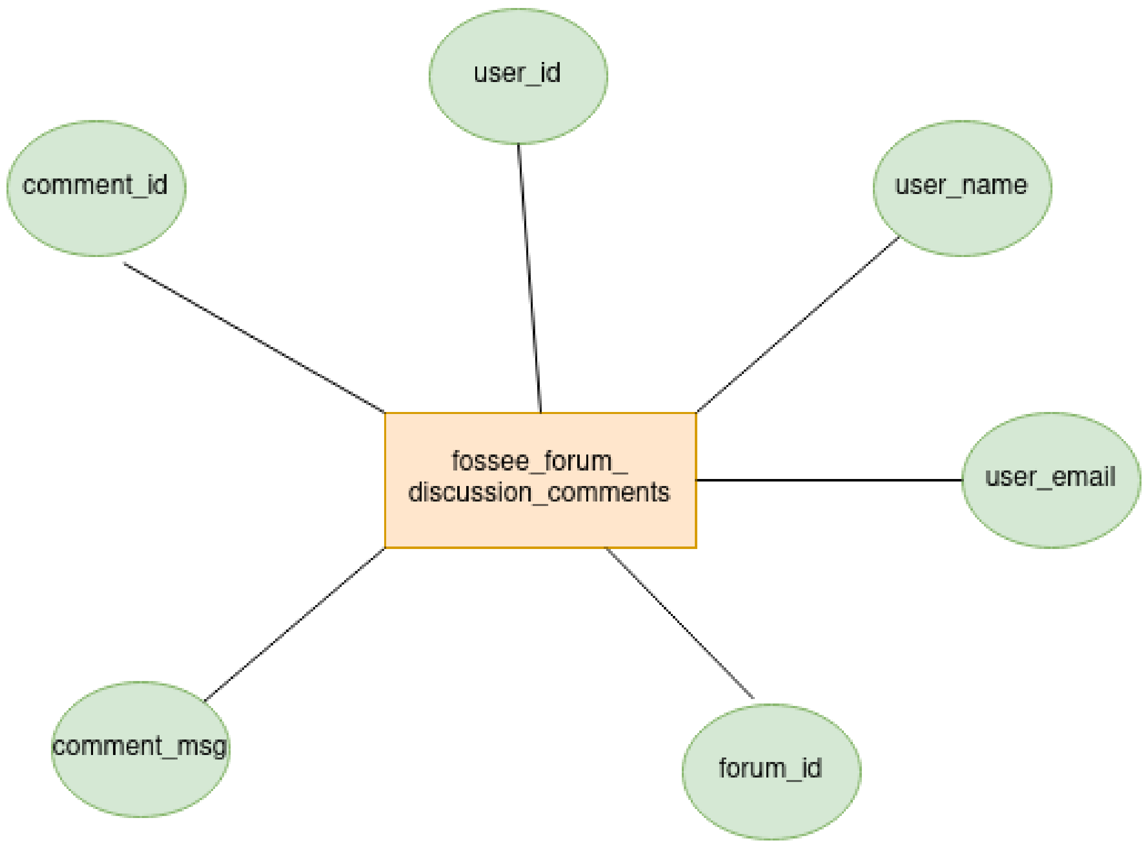


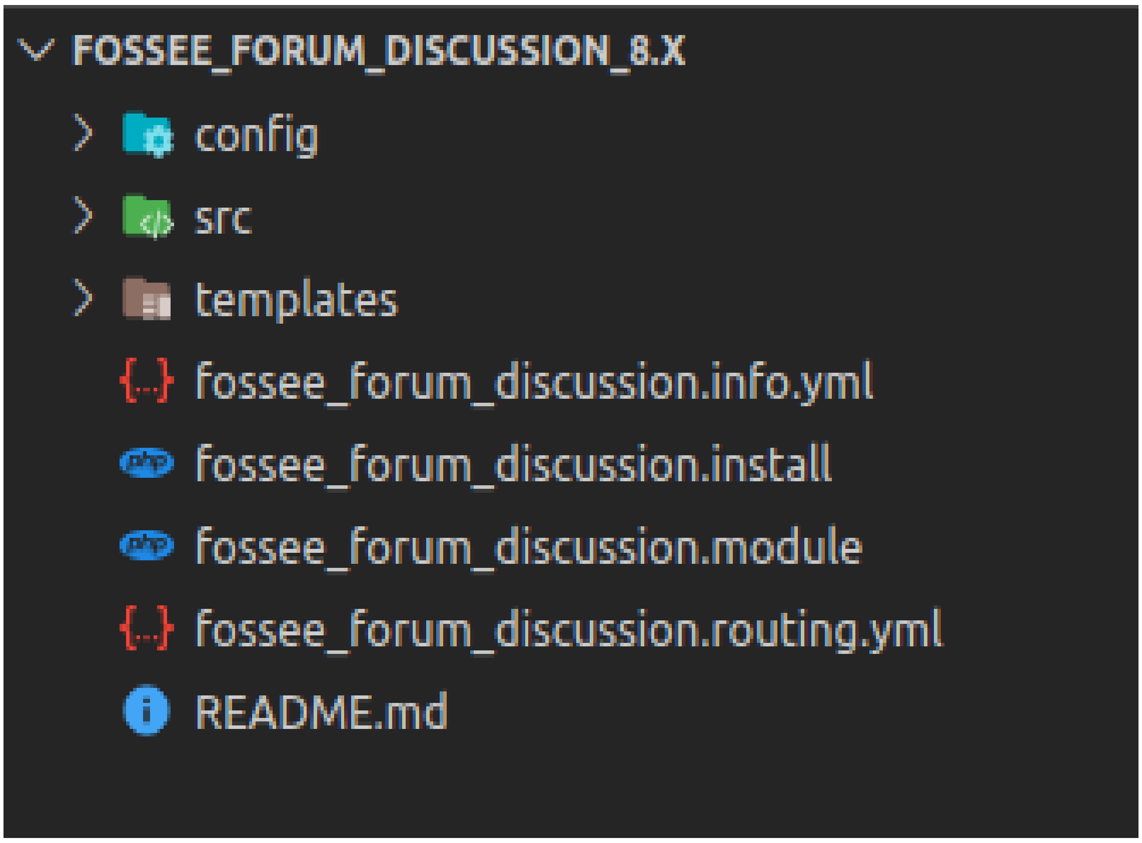
Figure 7.1: ER diagram for our custom module

7.3 File Structure

7.3.1 fossee_forum_discussion.info.yml

Drupal 8 uses .info files to store metadata about themes and modules.
For modules, the .info file is used for:

- rendering information on the Drupal Web GUI administration pages;
- providing criteria to control module activation and deactivation;
- notifying Drupal about the existence of a module;
- specifying the module's dependencies on other Drupal projects
- general administrative purposes in other contexts.



```
FOSSEE_FORUM_DISCUSSION_8.X
  config
  src
  templates
  fossee_forum_discussion.info.yml
  fossee_forum_discussion.install
  fossee_forum_discussion.module
  fossee_forum_discussion.routing.yml
  README.md
```

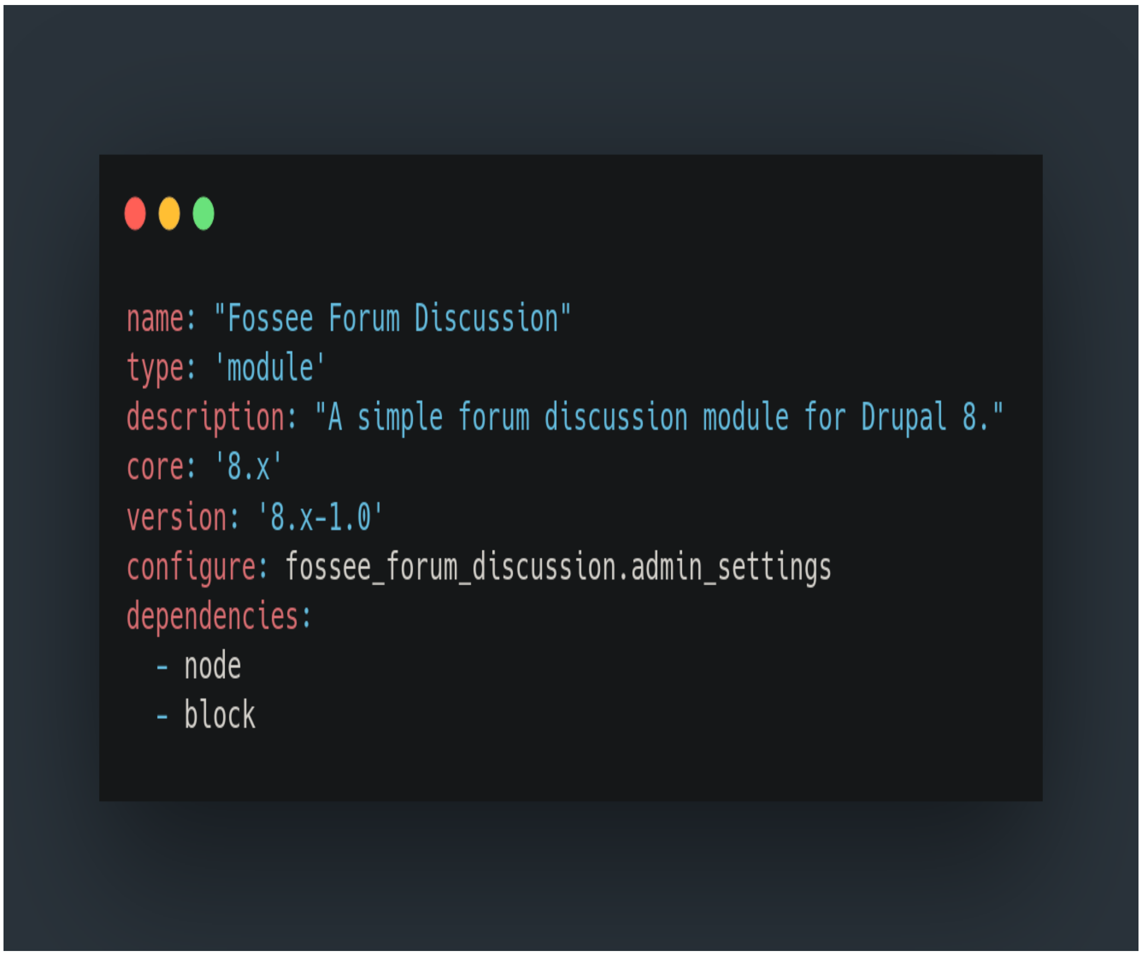
Figure 7.2: File structure for forum_discussion module

7.3.2 fossee_forum_discussion.install

Drupal 8 uses .install files to create database tables and fields, and to insert data. Drupal 8 .install files can also provide updates to change the database structure and content. Drupal 8 also uses .install files and includes the Field module in core. Field provides a different way to send data to nodes than in Drupal 6 and could replace some .install files.

7.3.3 fossee_forum_discussion.module

In Drupal 8 a .module file is not that much important. It is only used when we need to implement some specific hooks.
In our custom module we will be using .module file to implement hook_theme() and hook_mail() function. With the help of hook_theme(), I have added a custom theme to our module to display all the comments and respective replies.



```
name: "Fossee Forum Discussion"
type: 'module'
description: "A simple forum discussion module for Drupal 8."
core: '8.x'
version: '8.x-1.0'
configure: fossee_forum_discussion.admin_settings
dependencies:
  - node
  - block
```

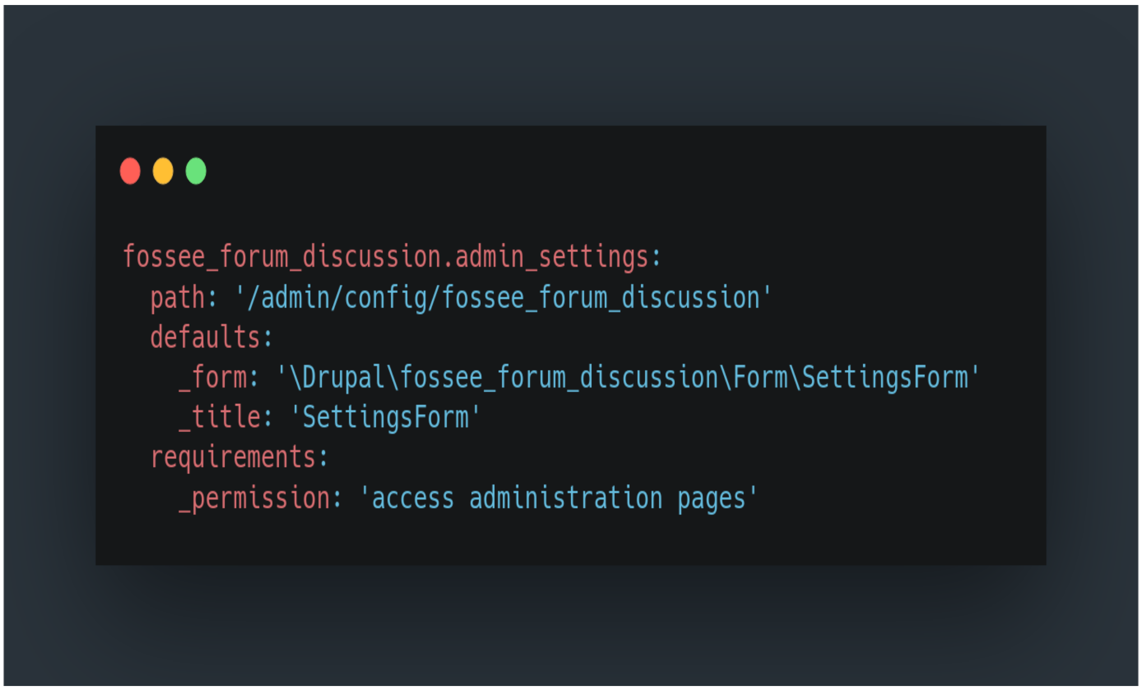
Figure 7.3: Info file for forum_discussion module

7.3.4 fossee_forum_discussion.routing.yml

A routing file is created to register different routes for a custom module. In our custom module, I will be using this routing file to register the route for our module configuration form.

7.3.5 \config

In the config directory, I had to create one more directory and that is the install directory. Inside this install directory I had to create **fossee_forum_discussion.settings.yml** file. This is the file that will be storing all our default values for our custom module configuration variables.



```yaml
fossee_forum_discussion.admin_settings:
  path: '/admin/config/fossee_forum_discussion'
  defaults:
    _form: '\Drupal\fossee_forum_discussion\Form\SettingsForm'
    _title: 'SettingsForm'
  requirements:
    _permission: 'access administration pages'
```

Figure 7.4: Routing file for forum_discussion

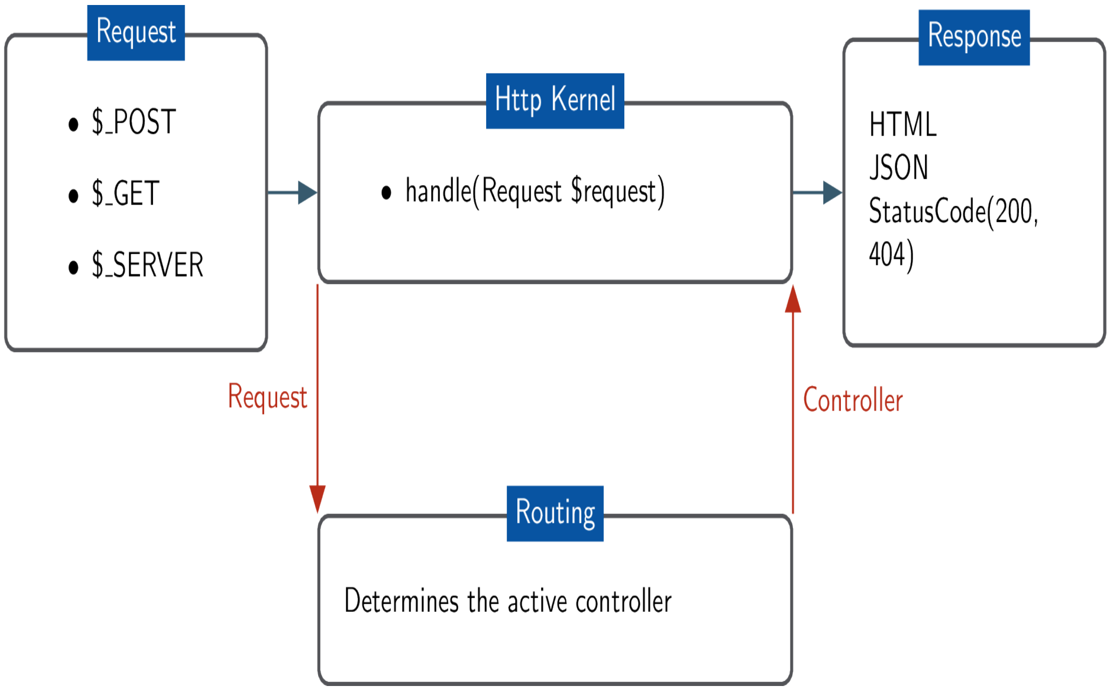


Figure 7.5: Drupal 8 routing

7.3.6 \src

In the src directory I created two sub directories as Form and Plugin. In the Form directory I created all the forms that are needed throughout our module. In our module I created three forms. First two forms to make form for comment and reply and the third form is to render the settings form.

In the Plugin directory I created the files for our custom blocks. There were two main blocks that are needed by our custom module:

- CommentFormBlock
- CommentViewBlock

7.3.7 \templates

In the templates folder, I have created a twig template file to display all the comments and respective replies according to the forum ID or page ID.

Twig is a template engine for PHP and it is part of the Symfony2 framework.
In Drupal 8 Twig replaces PHPTemplate as the default templating engine. One of the results of this change is that all of the theme_* functions and PHPTemplate based *.tpl.php files have been replaced by *.html.twig templating files.
In twig template file we also have the option to write javascript. This feature opens all the way to write super functional codes.

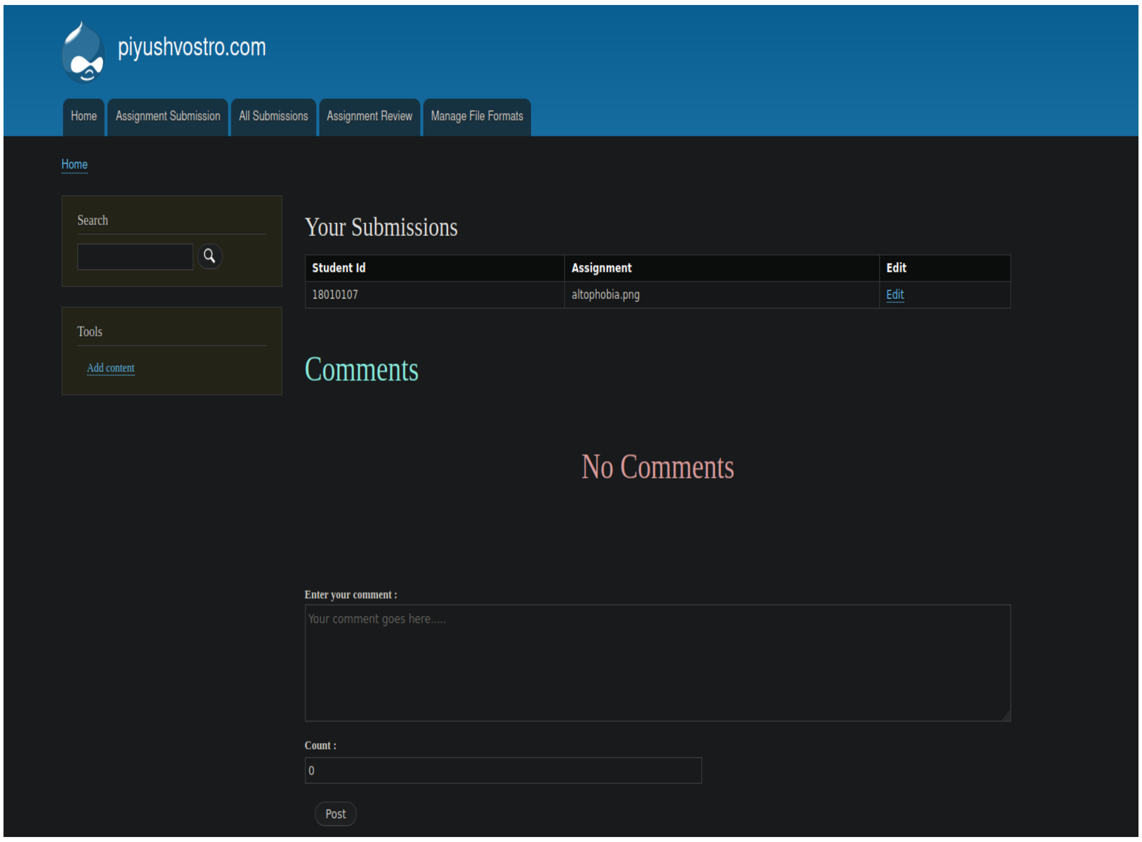
7.4 Module Functionalities

In our custom module, I basically had to implement forum_discussion functionality.
But there were also other functionalities that I had to implement.
Here is a list of all those basic functionalities that need to be there in our custom module:

- There should be separate forums for different pages according to the page_id.
- Only authorised user should be able to post a comment or reply.
- Unauthorised users should be redirected to login page on post actions.
- Multiple spaces should be filtered out from the comment or reply.
- There should be a limit for minimum and maximum length of comment and reply messages.

- There should be an option to set these limits from module configuration.
- A custom mail should be triggered to comment author when they posts a new comment.
- A custom mail should be triggered to reply author when a user posts a new reply.
- A custom mail should be triggered to forum members when a user posts a new reply.
- All emails should also be triggered to the emails mentioned in the bcc and cc variables in module configuration.

7.5 Final Results



Your Submissions

Student Id	Assignment	Edit
18010107	altophobia.png	Edit

Comments

No Comments

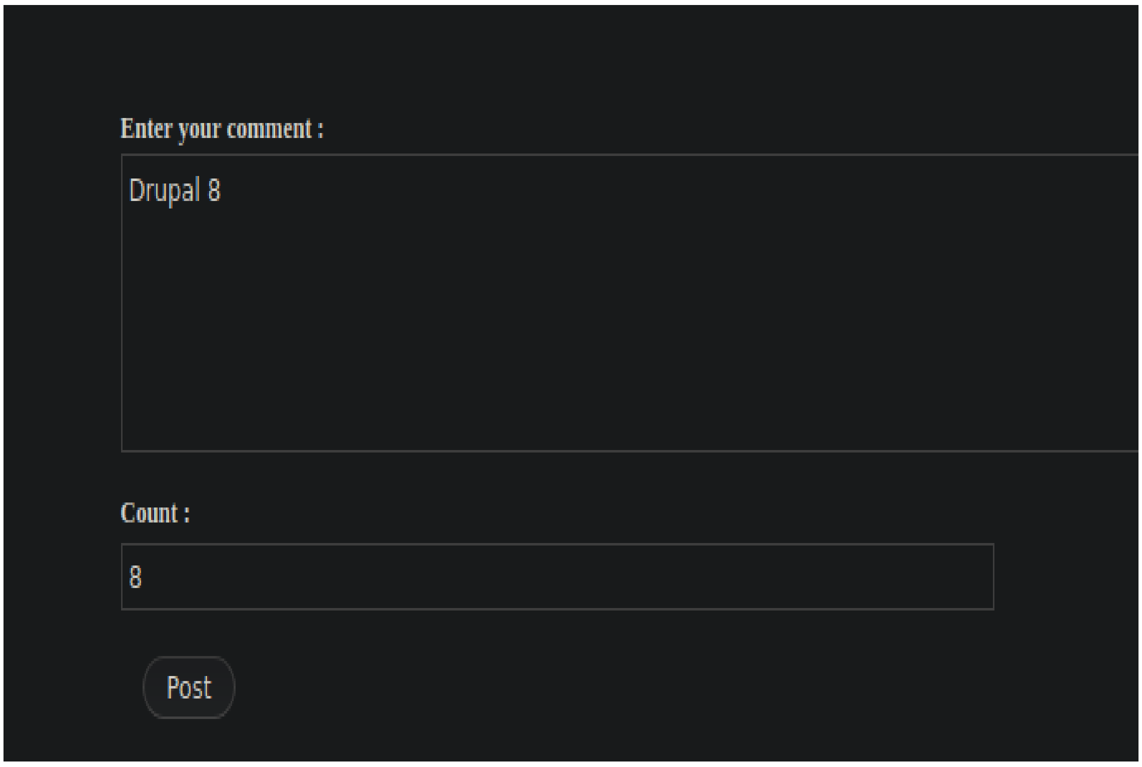
Enter your comment :

Your comment goes here.....

Count :

0

Figure 7.6: No comments view

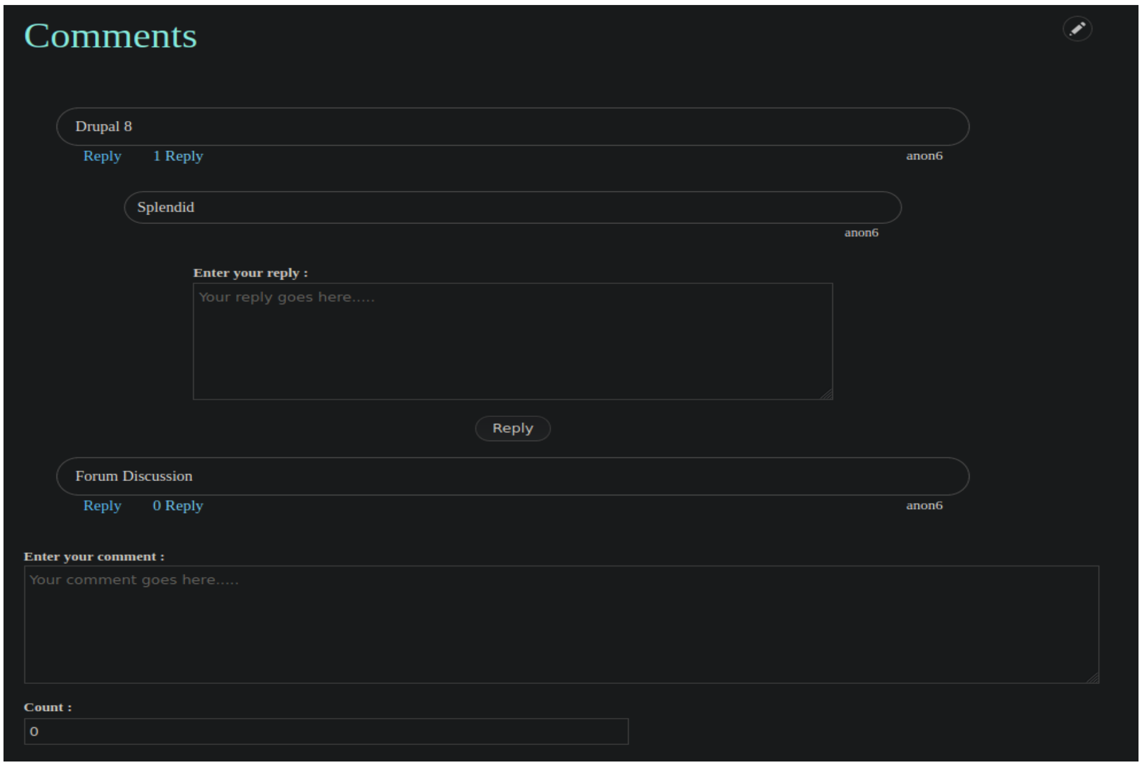

Enter your comment :

Drupal 8

Count :

8

Figure 7.7: Count of the comment message


Comments

Drupal 8

anon6

Splendid

anon6

Enter your reply :

Your reply goes here.....

Forum Discussion

anon6

Enter your comment :

Your comment goes here.....

Count :

0

Figure 7.8: Full expanded view

Reference

- Drupal 8 Docs.
- Drupal 7 Docs