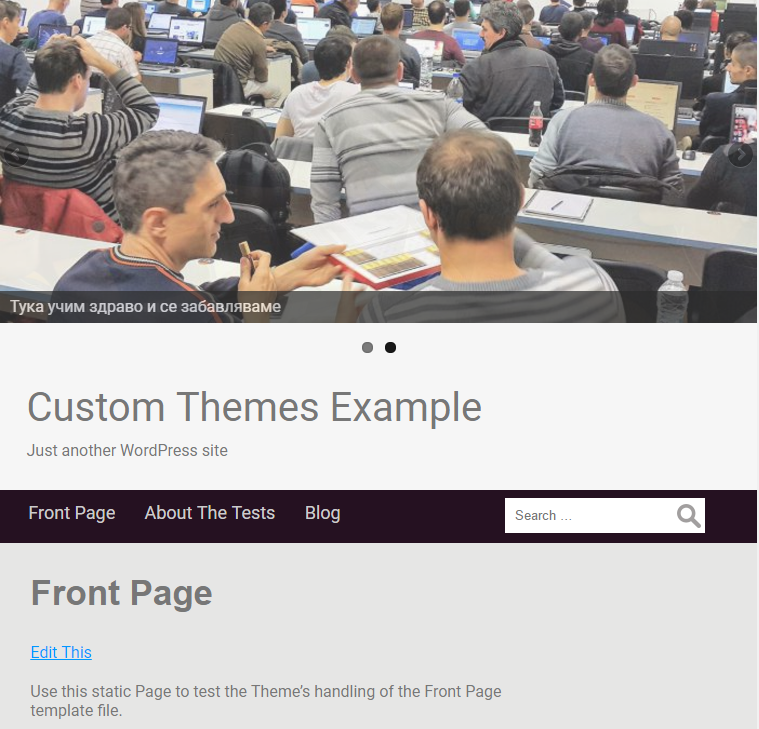
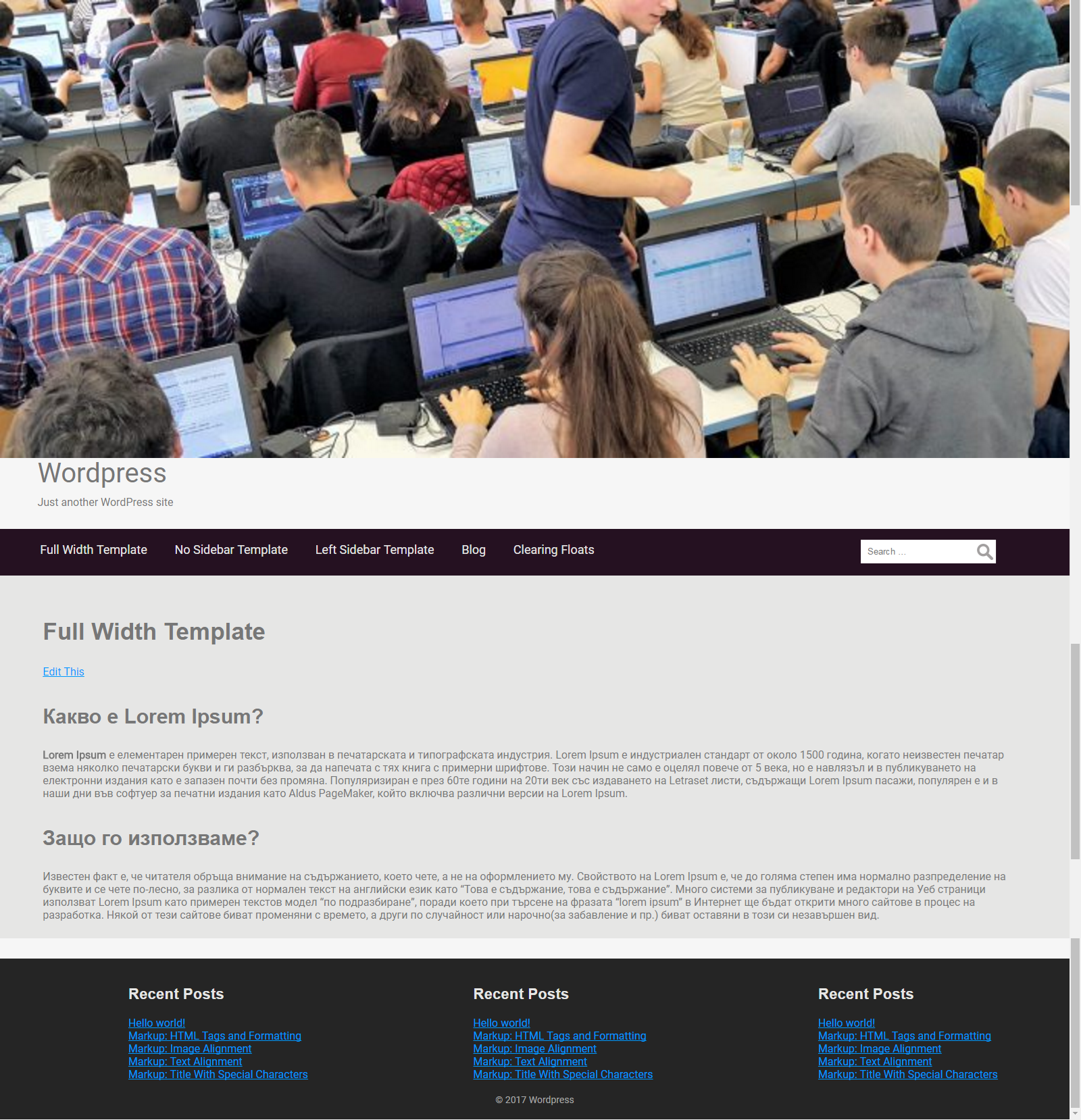
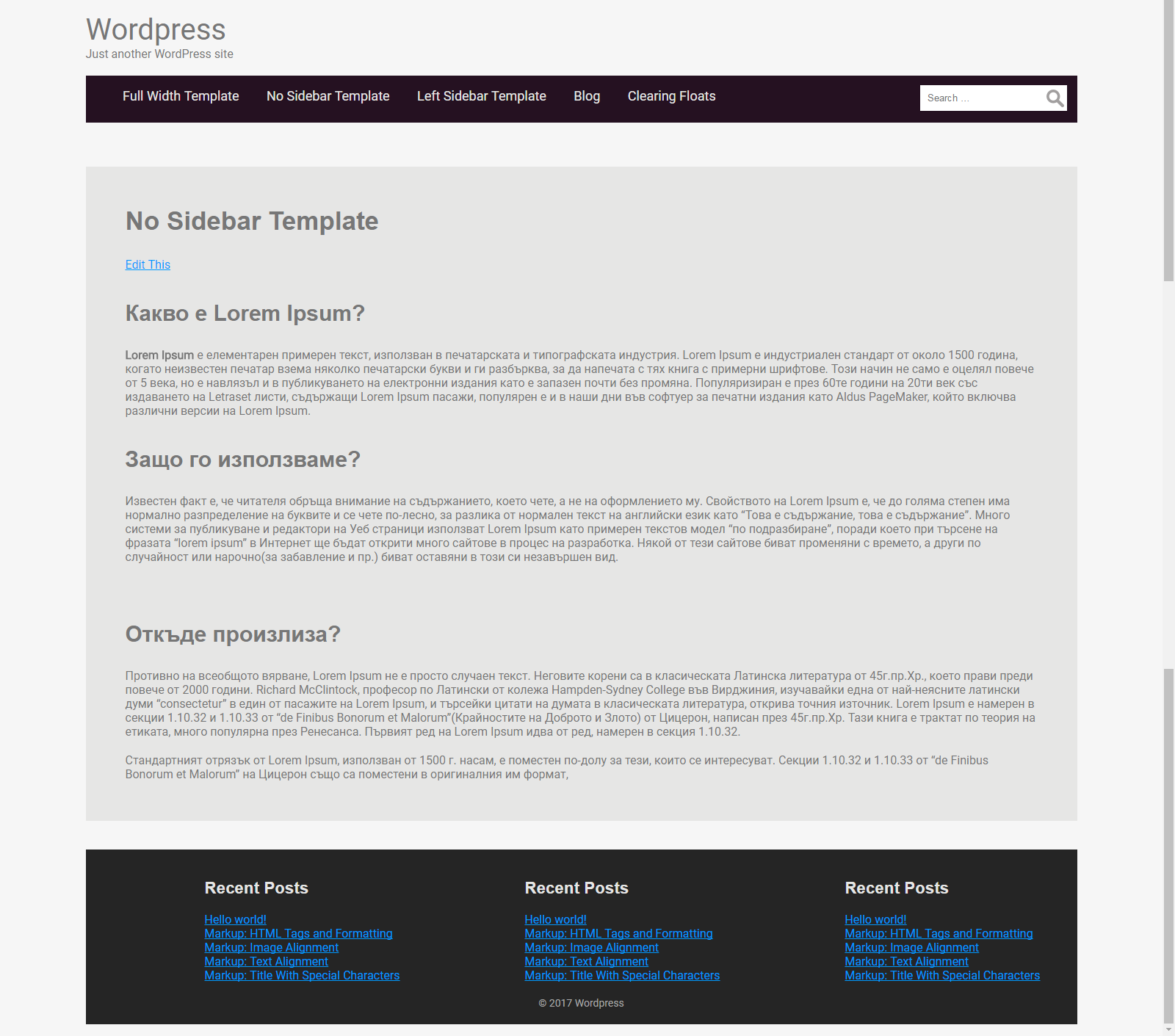
# Exercises: Creating a Custom WordPress Theme

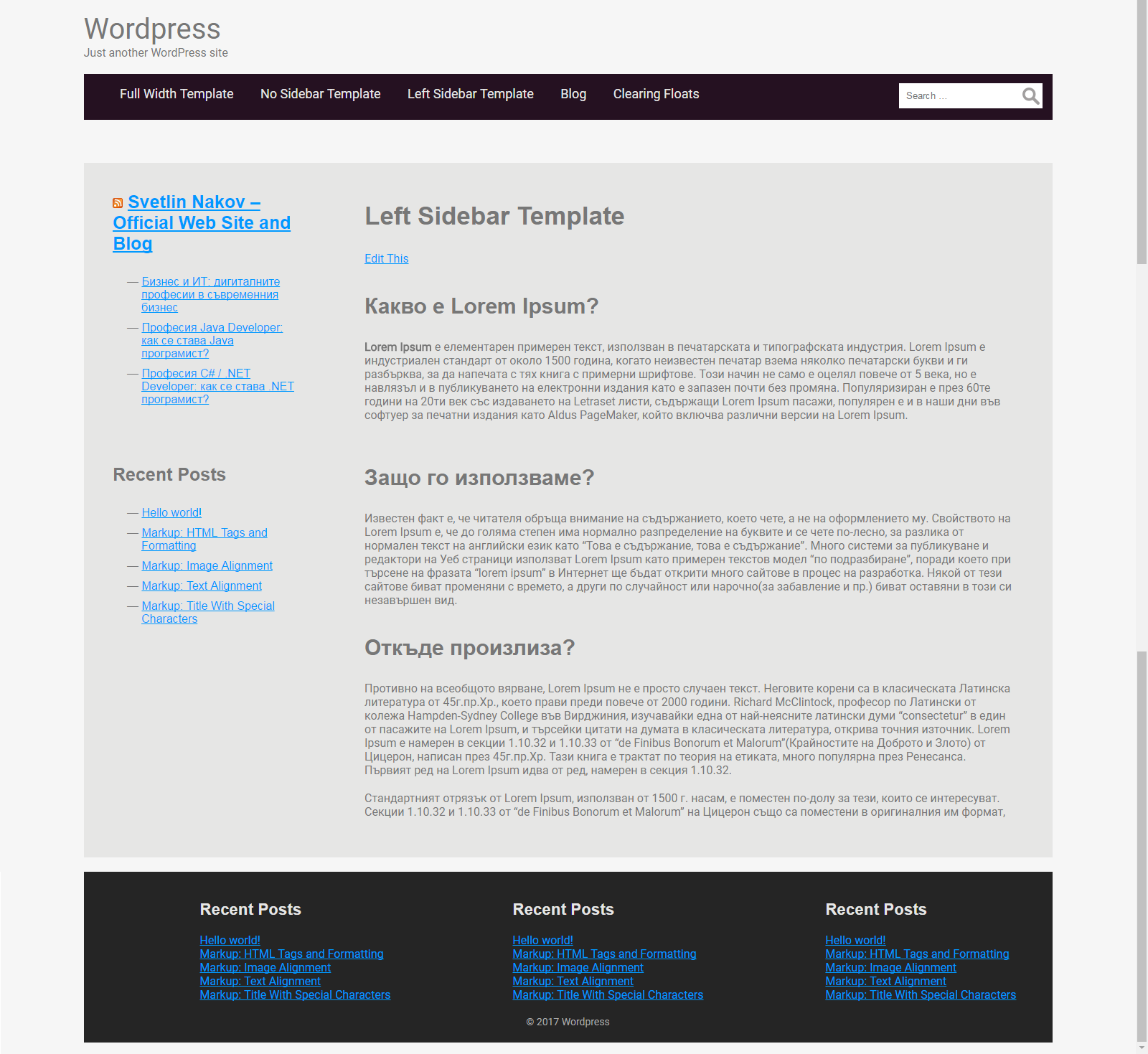
Problems for exercises and homework for the [“WordPress Basics” course @ SoftUni](https://softuni.bg/courses/wordpress-basics).

In this homework, you are supposed to **create a custom WP theme**, following step-by-step instructions. You shall create the following WordPress theme:





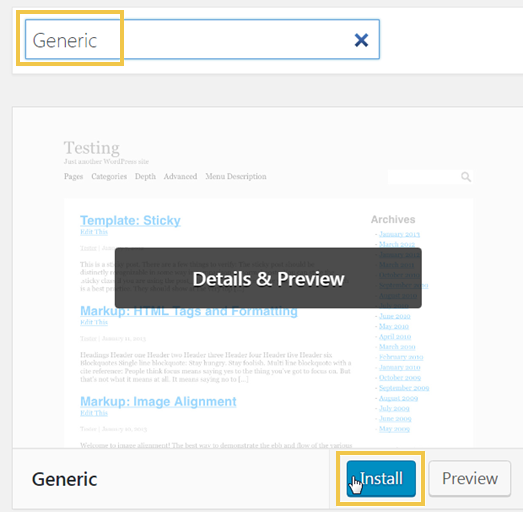
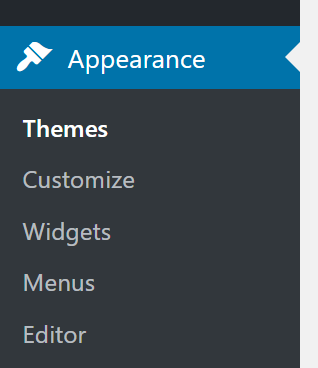




## Install the "Generic" WP Theme

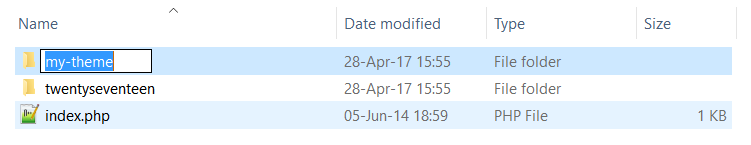
Typically, WordPress themes are built from **starter themes**. A starter theme is taken as base and is modified to achieve the theme goals, design, functionality and user experience. We shall use the “**Generic**” WP theme as starter theme.

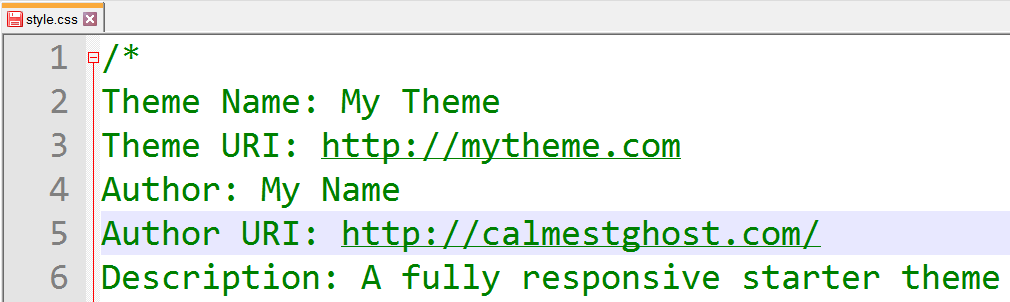
Install and activate the "**Generic**" WP theme from the official WP theme directory:



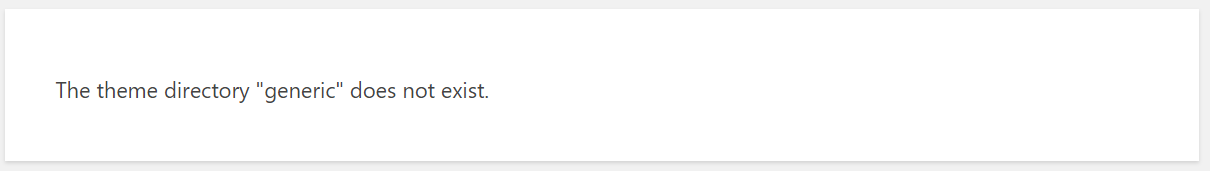
## Rename the Theme

**Rename the theme folder** and edit its metadata in the style.css file.

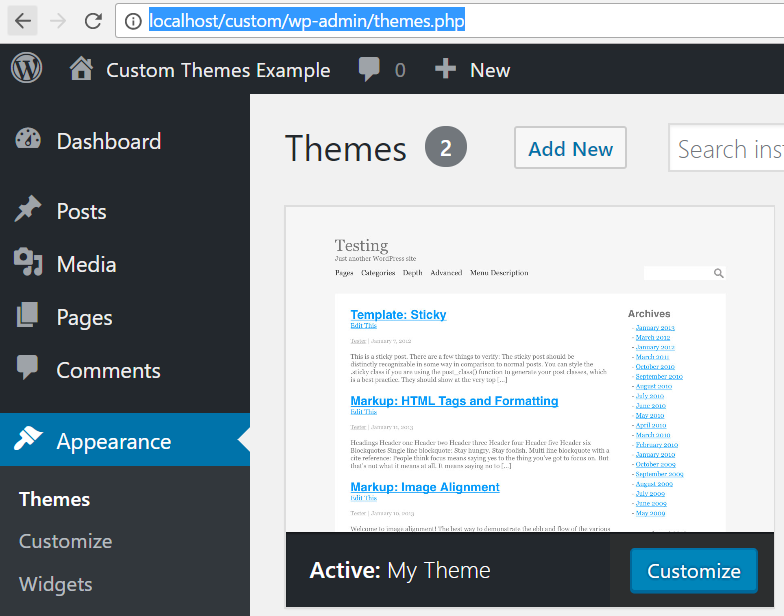




You will get this error:



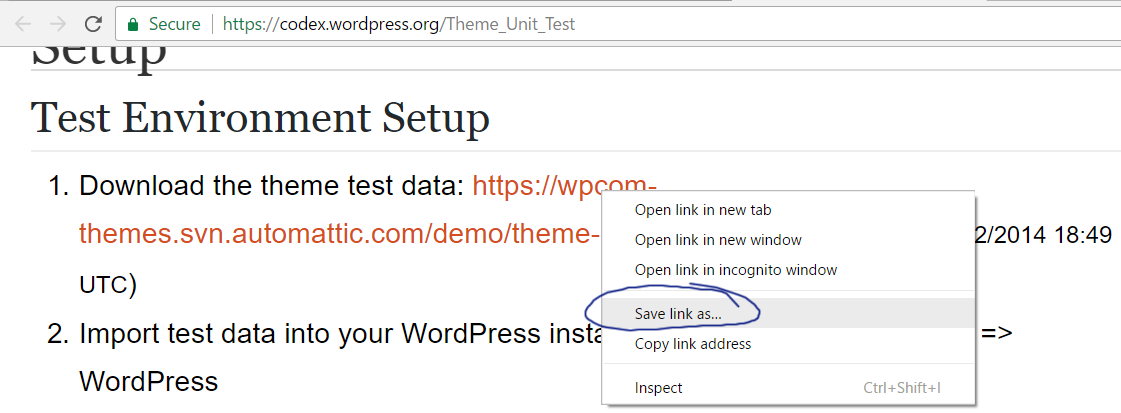
Enter in the **WordPress admin panel** (by opening /wp-admin/ URL of your site) and activate the renamed theme:



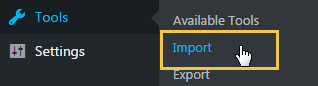
## Import Sample Content

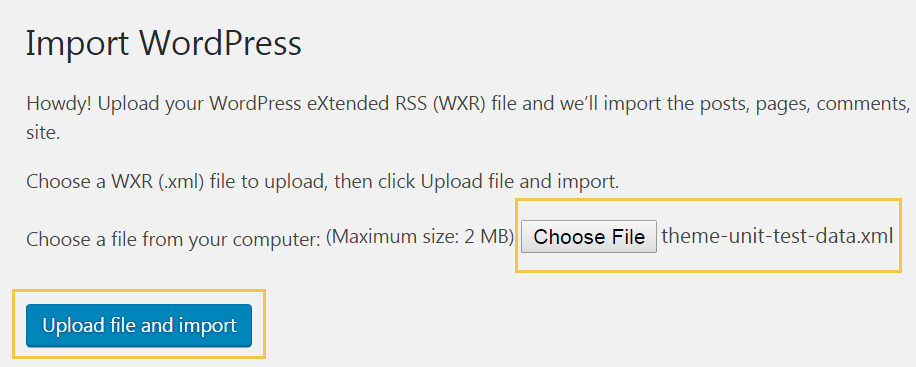
To develop a WP theme, you need to have some content in your WP site: pages, articles (news), categories, tags, links, images, comments, etc. If your site is empty, you will not be able to test it, so please import some content, or create some content by hand.

Download **sample content** from <https://codex.wordpress.org/Theme_Unit_Test>.

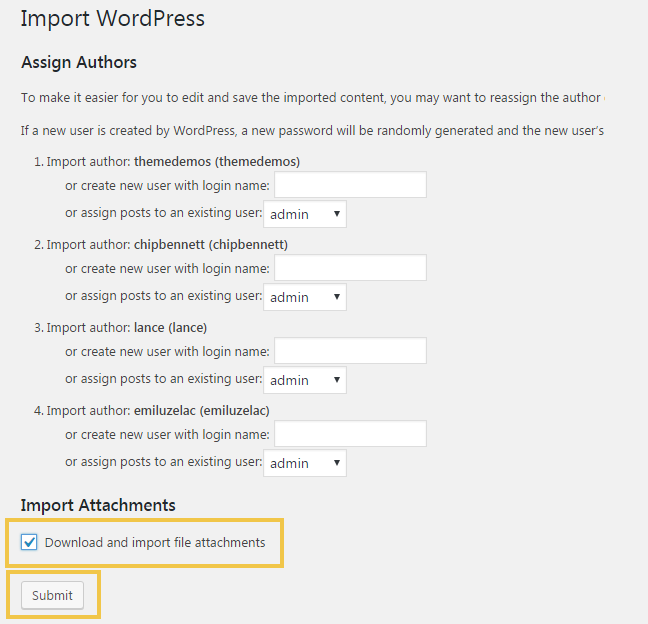


**Import** the sample content:



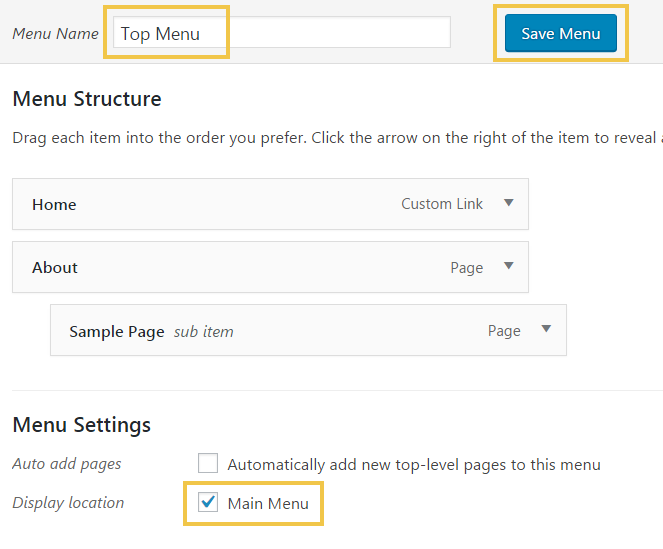
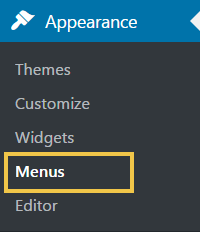
Be sure to import the content **with attachments**. You may create **new users** or assign **existing users** as content authors:



Be patient. It usually takes 20-50 seconds with no progress indication.

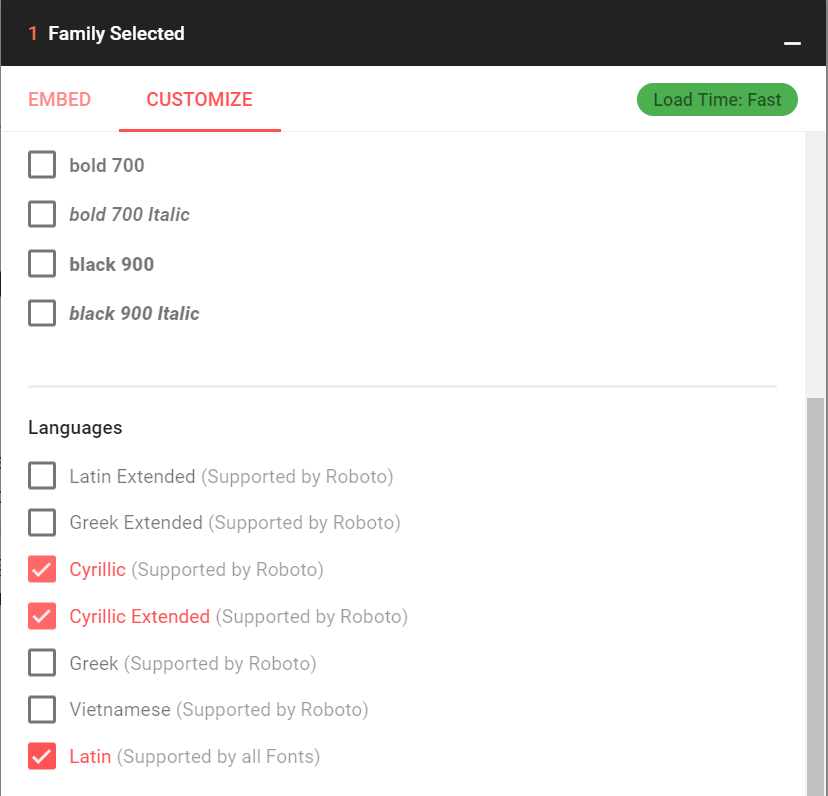
## Create the Top Menu

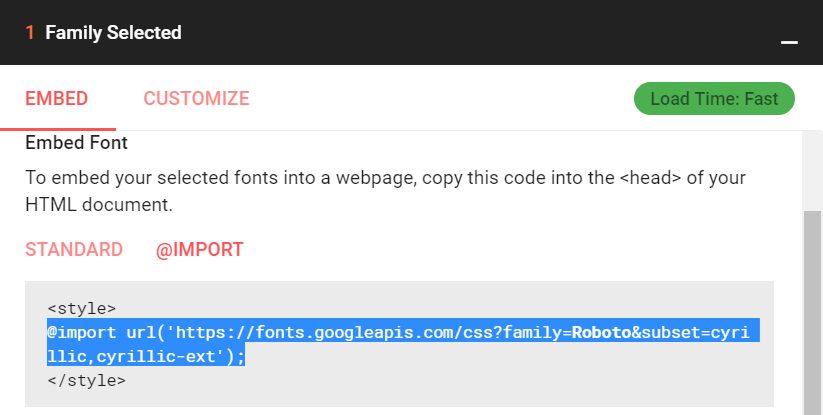
Create the **top menu** and attach it to the "**Main Menu**" location:



## Change the Theme Font

1. Open style.css.
2. Import the "**Roboto**" font from <https://fonts.google.com/specimen/Roboto>. If you plan to use **Cyrillic texts** in your site, ensure you have included the Cyrillic languages:

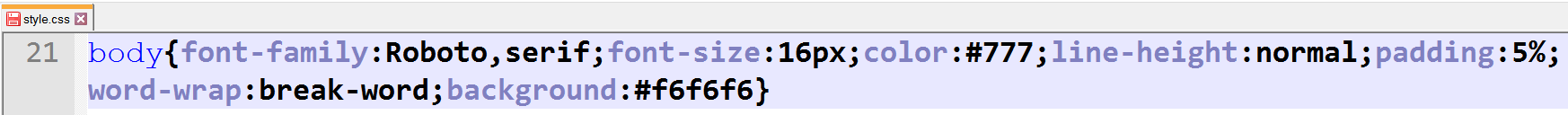




Import the front from your CSS:



Assign the “Roboto” font at all places where “georgia” is used:



1. Change the **colors** for the "top-menu".

You may use **different** colors and styling. For example, put the code below at the end of your CSS:

**#menu, #menu ul li:hover ul li a {**

border: 0;

background: #251121;

padding: 1%;

**}**

**#menu #menu-short li a {**

color: #ccc;

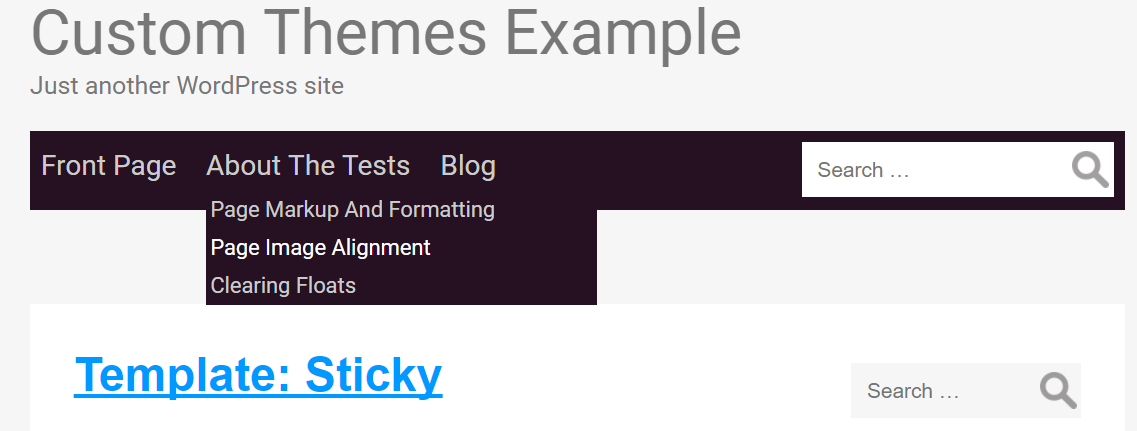
**}**

**#menu #menu-short li a:hover {**

color: #fff;

**}**

Test the results in the browser. To avoid the browser caching, use a force-refresh using **[Ctrl+F5]**:



1. Change main background colors and body padding in the CSS:

#container {

background: #e6e6e5;

margin: 0 auto 20px;}

#wrapper {

max-width: none;

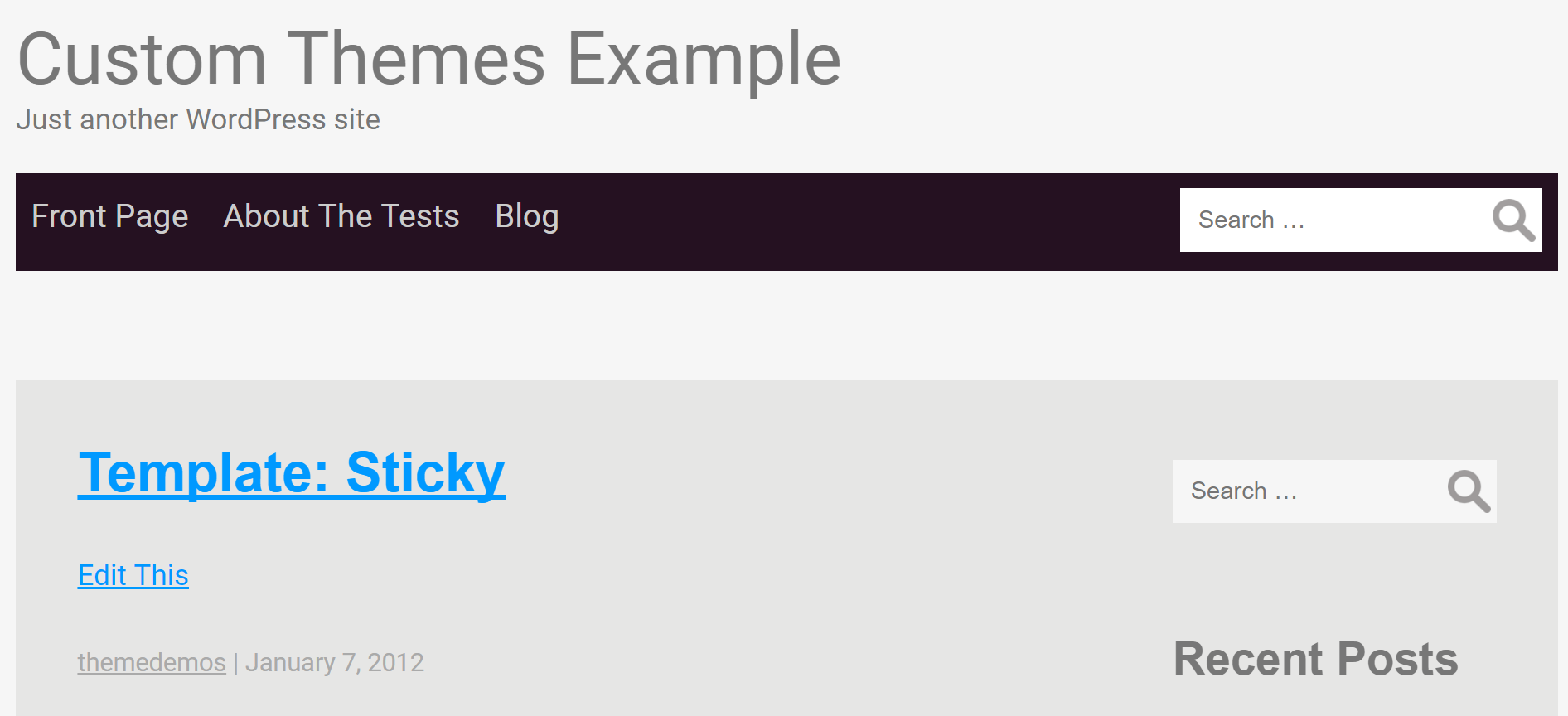
}

body {

padding: 1%;

}

Test again the result in your Web browser:



## Creating Page Templates

Page templates provide **different layouts** for your pages. For example, at your home page you may want to display content in 2 columns, while in the “news” section, you may want to have a news article with a sidebar on the right. Layouts in WordPress themes are created with **page templates**. Let’s create a few page templates.

## "No-Sidebar" Page Template: PHP Code

1. Copy page.php file into new file: page-no-sidebar.php.
2. Open page-no-sidebar.php file and add the following code:

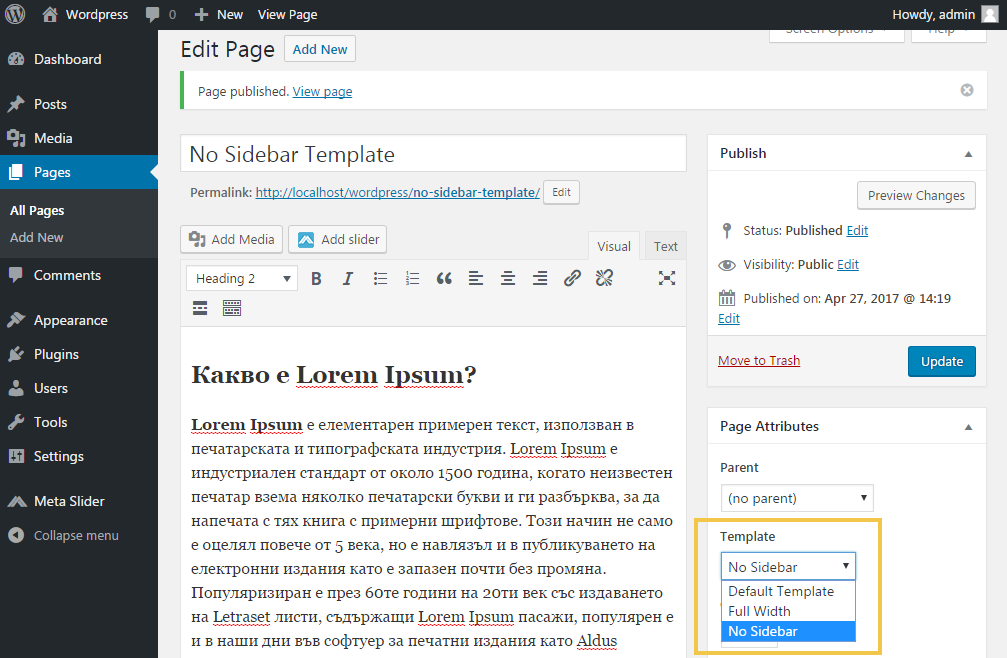
**<?php** /\* Template Name: No Sidebar \*/ **?>**

The above line tells WordPress that this is a page template, so it displays its name in the list of templates in the page editor.

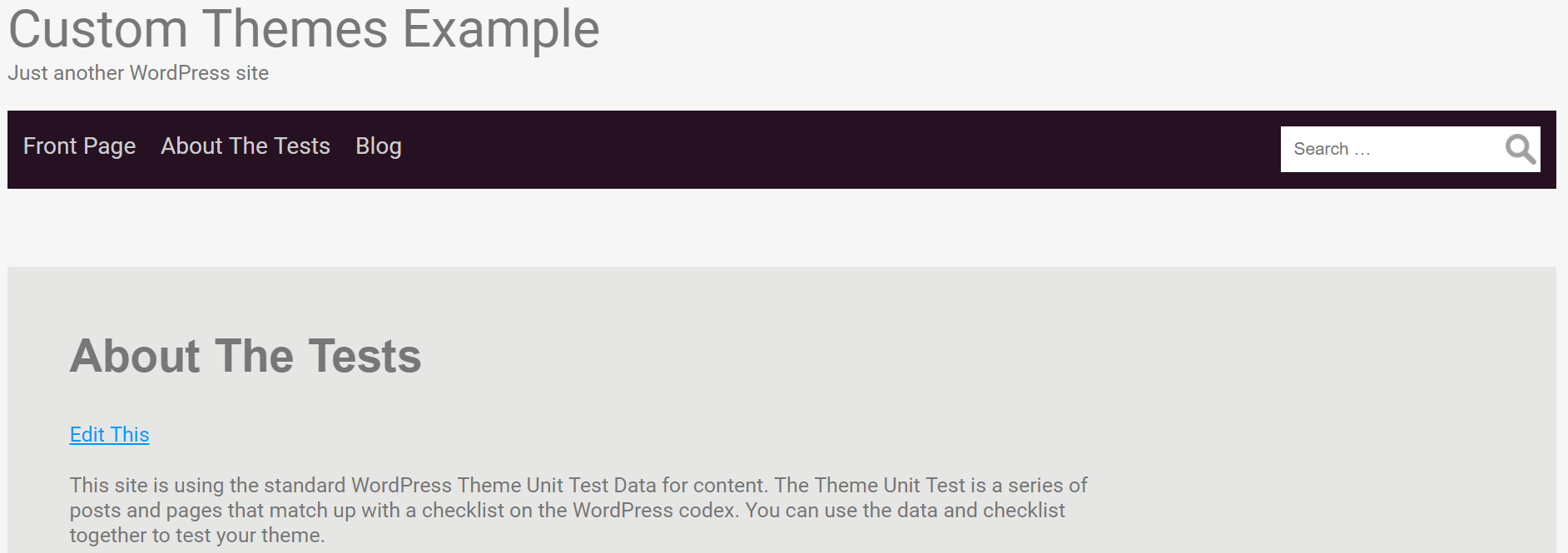
1. Remove the sidebar from the template:

~~<?php> get\_sidebar() ?>~~

1. Now create or edit a page and assign the new template “**No Sidebar**”. Test the page in your browser.



Test your new page template in your Web browser:



Looks good: the sidebar has gone. But its space still stays empty on the right side. Let’s fix this.

## "No-Sidebar" Page Template: CSS Code

Add some CSS for this page template. WordPress assigns the name of the page as CSS class in the page body, so styling is easy. In this case, the class name is “page-template-page-no-sidebar”:

.page-template-page-no-sidebar #content {

width: 100%;

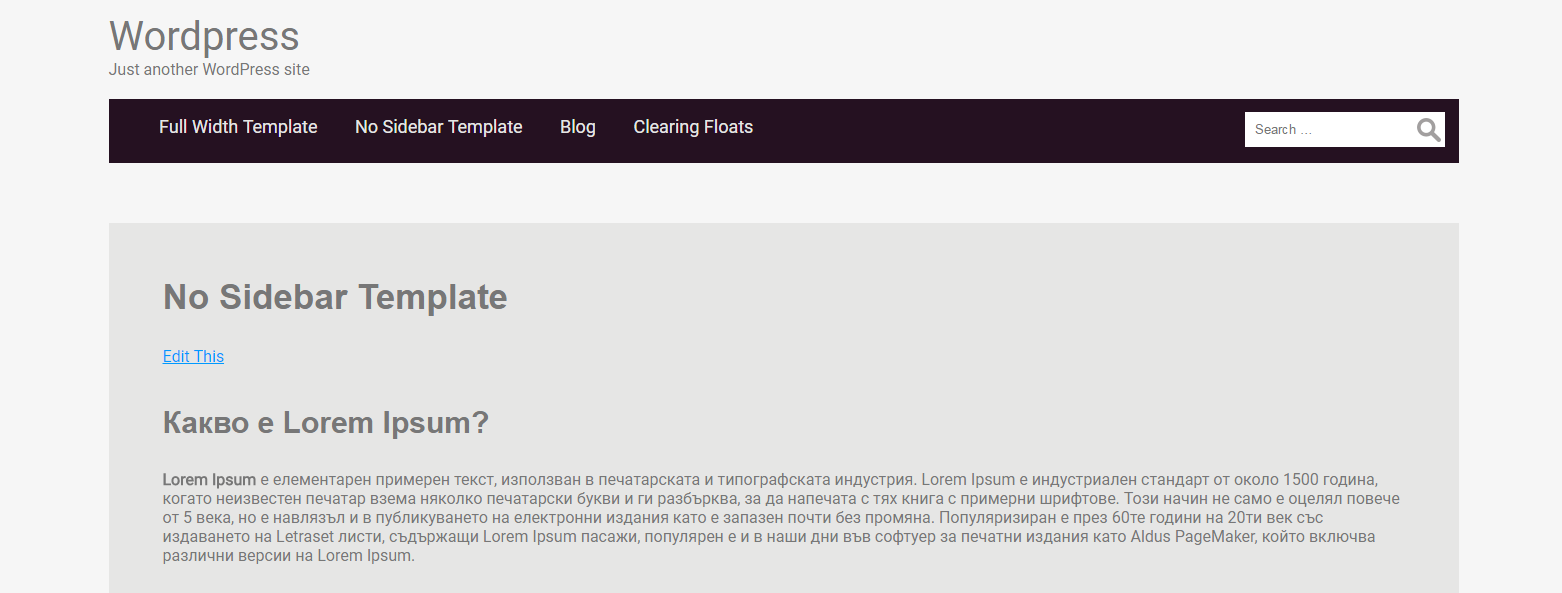
}

#menu ul, #menu ul li {

margin-left: 1.4%;

}

Test again in browser:



## "Full-Width" Page Template: PHP Code

Let’s create a “full width” page template with no spacing around the page. This may be good for the site front page, for example.

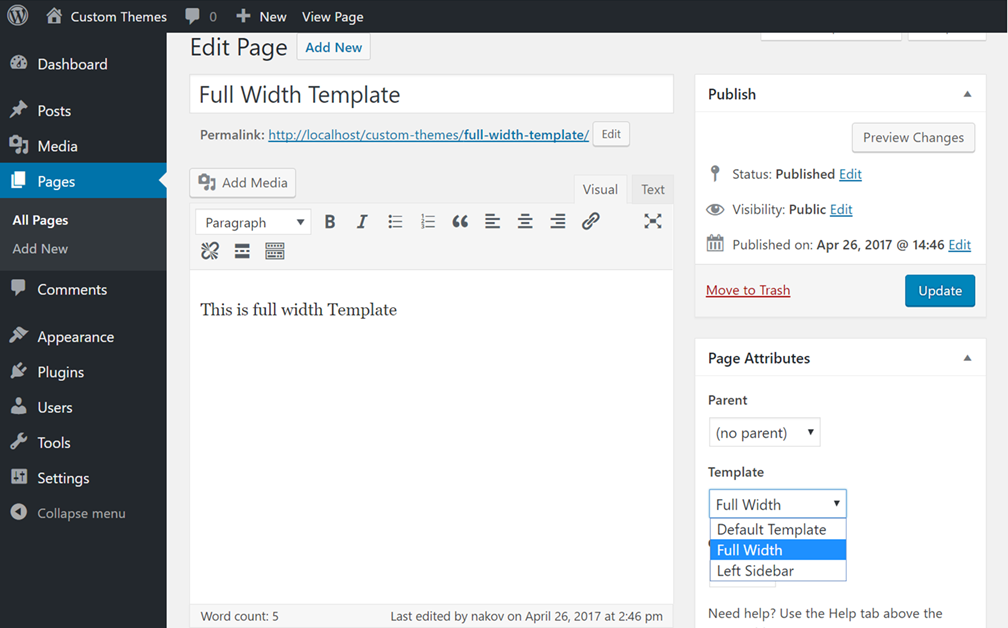
1. Copy page.php file into new file: page-full-width.php.
2. Open page-full-width.php file and add the following code:

<?php /\* Template Name: Full Width \*/ ?>

1. Remove the sidebar code:

<?php> **get\_sidebar()** ?>

1. Assign the new template for some page:



## "Full-Width" Page Template: CSS Code

Now style the new page template in the site CSS. Recall that WordPress automatically assigns the page template as class name in the page body. In our case the class name is “page-template-page-full-width”:

.page-template-page-full-width #content { max-width: 100%; }

.page-template-page-full-width #wrapper { max-width: none; }

body.page-template-page-full-width { padding: 0; }

.page-template-page-full-width #header { margin-bottom: 0; }

.page-template-page-full-width #branding {

padding-left: 3.5%;

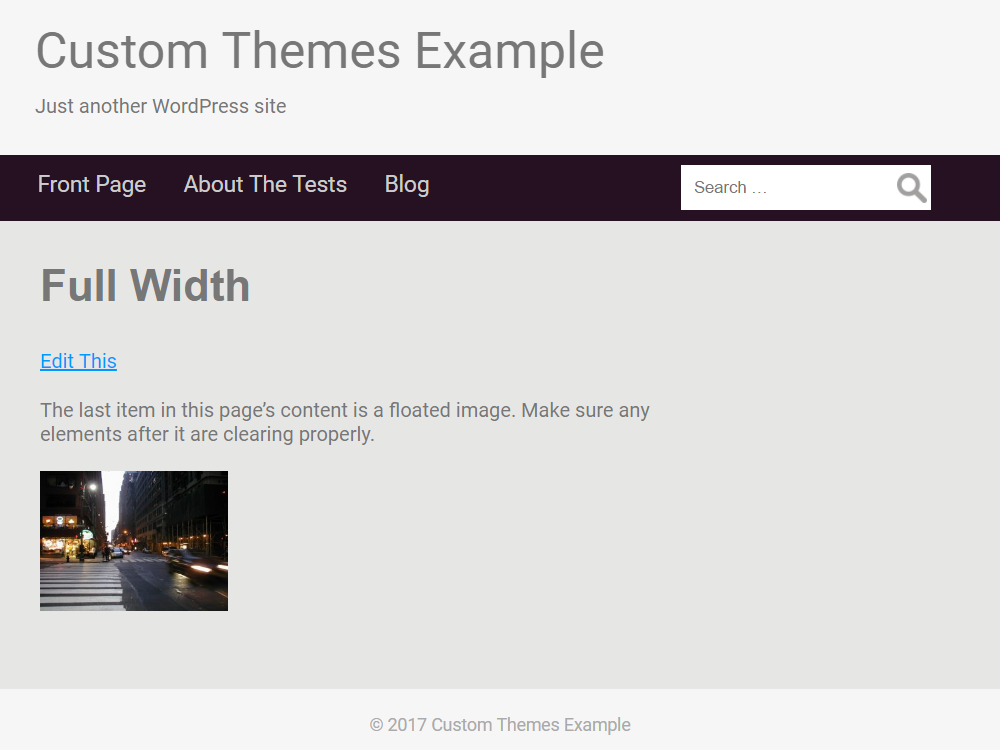
padding-top: 20px;

padding-bottom: 10px; }

.page-template-page-full-width #site-title { padding-bottom: 10px; }

.page-template-page-full-width #search { margin-right: 6%; }

The new page template should look like this:



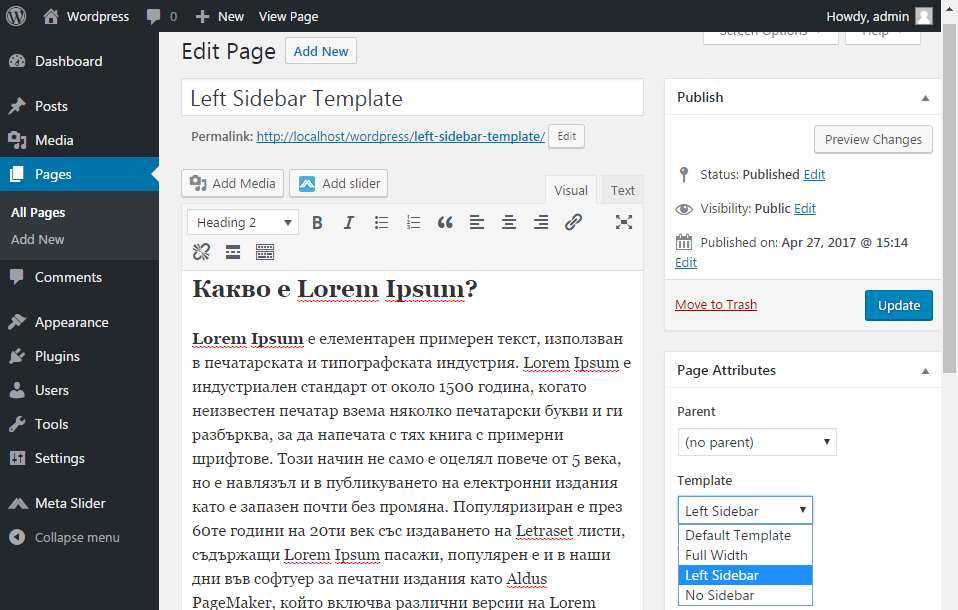
## "Left-Sidebar" Page Template: PHP Code

Let’s create a new **page template** that shows the **sidebar on the left** and the **page content on the right**. It will be very similar to the original page template from the file “page.php”.

1. Copy page.php file into new file: page-left-sidebar.php.
2. Open page-left-sidebar.php file and add the following code:

<?php /\* Template Name: Left Sidebar \*/ ?>

1. Assign the new template for some page:

****

## "Left-Sidebar" Page Template: CSS Code

Add some CSS code for your new page template:

.page-template-page-left-sidebar #sidebar {

width: 25%;

padding: 3% 3%;

float: right;

}

.page-template-page-left-sidebar #content {

float: right;

}

The new page template should look like this:



## Creating Custom Widget Area

**Custom widget areas** allow site users to define custom content in specific location of the page, e.g. in the sidebars, in the footer or somewhere else. Widget areas give flexibility: instead of hard-coded content in the theme, it allows widgets to be added / deleted later.

## Create Widget Area for Slider

First, let’s create a **widget area to hold a slider** at the top of the page.

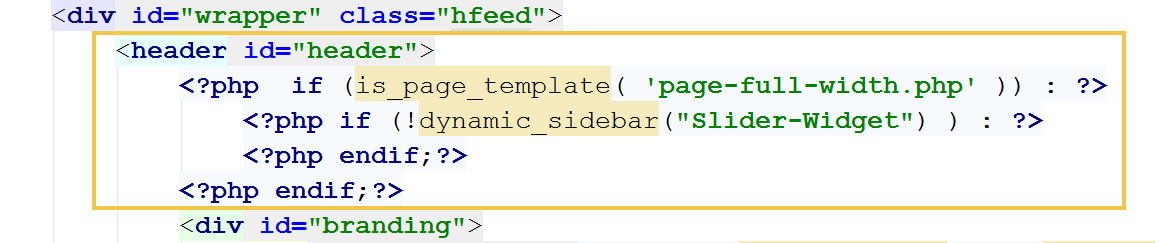
First, define a new widget area using a piece of **PHP code** in the "functions.php" file:

function slider\_widget() {  
 register\_sidebar(array(  
 'name' => 'Slider-Widget',  
 'id' => 'widget-slider',  
 'before\_widget' => '<div class="slider-widget">',  
 'after\_widget' => '</div>',  
 'before\_title' => '<h3 class="slider-widget-title">',  
 'after\_title' => '</h3>',  
 ));}  
add\_action('widgets\_init', 'slider\_widget');

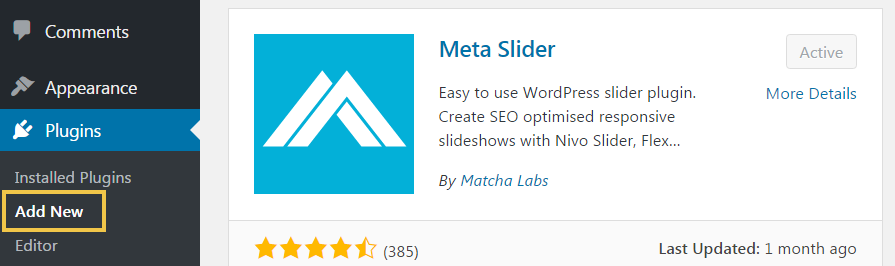
Now open header.php and paste the following code in the “wrapper” div:

<?php if (is\_page\_template( 'page-full-width.php' )) : ?>  
 <?php if (!dynamic\_sidebar("Slider-Widget") ) : ?>  
 <?php endif;?>  
<?php endif;?>

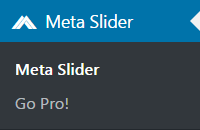
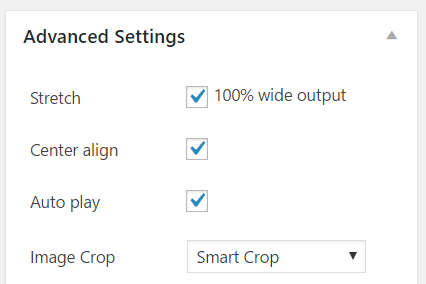
See the screenshot below to get a better idea:

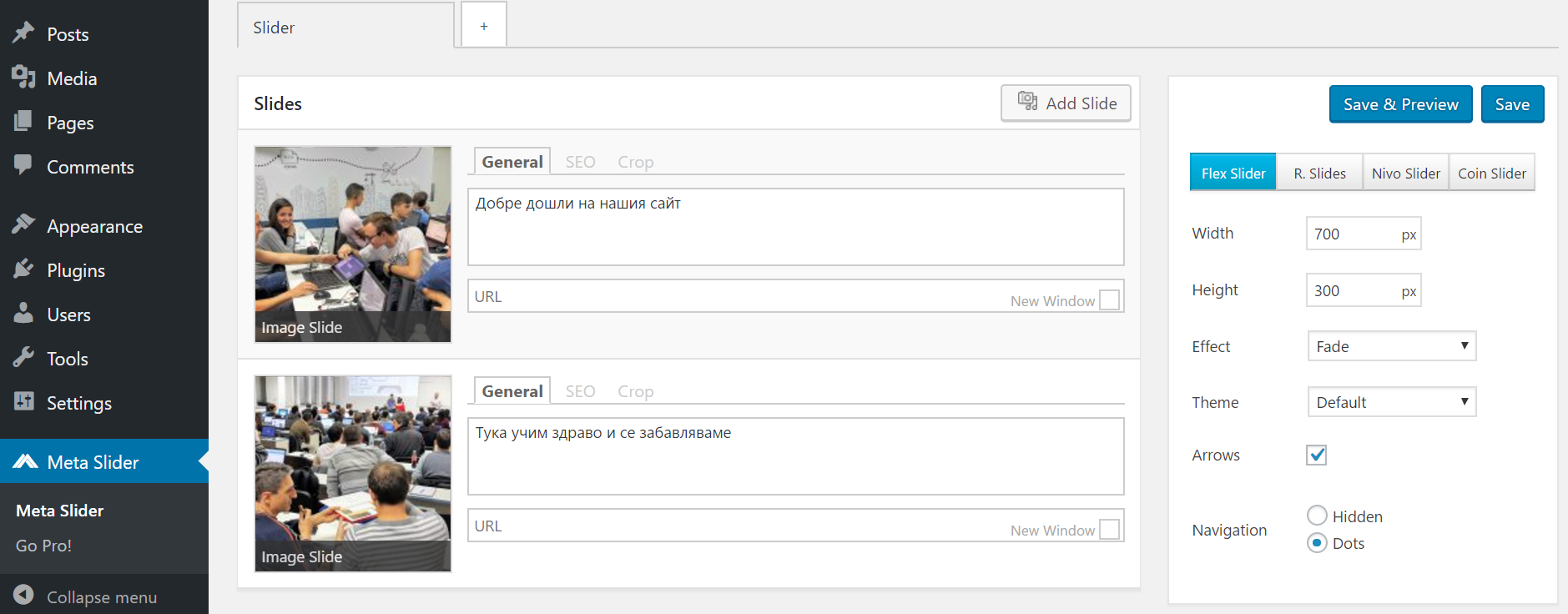


Now install the "**Meta Slider"** plugin:

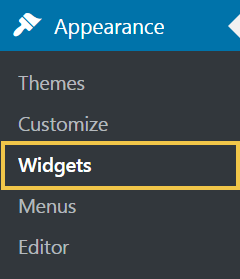


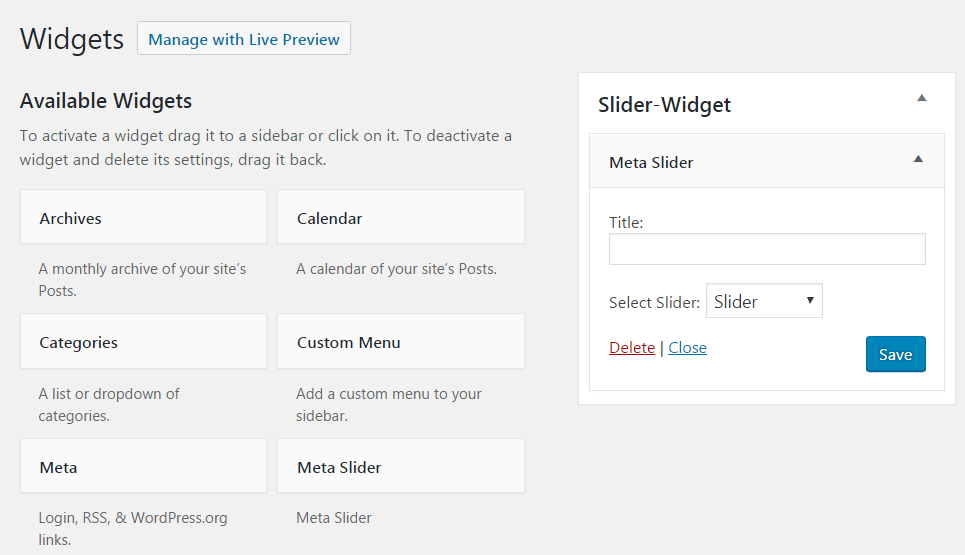
Create a **slider**, holding a few slides (upload some photos and write some texts):

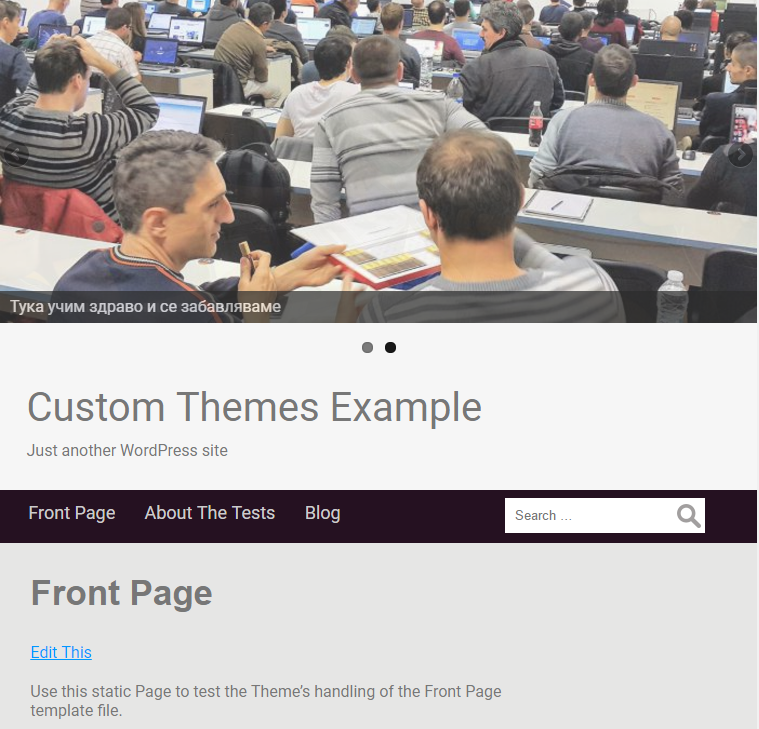


Now insert the meta slider widget in the “**Slider-Widget**” custom widget area:





Open your **full-width page**. It should hold the slider at the top, before the site title.



## Create Widget Areas for the Footer

Now let’s create **3 new custom widget areas** to hold the site footer content. Let’s call them “**Left Footer**”, “**Middle Footer**” and “**Right Footer**”. Add the following PHP code in the “functions.php” file:

function left\_footer\_widget() {

register\_sidebar(array(

'name' => 'Left Footer',

'id' => 'widget-footer-area-left',

'before\_widget' => '<div class="left-footer">',

'after\_widget' => '</div>',

'before\_title' => '<h3 class="footer-widget-title">',

'after\_title' => '</h3>',

));}

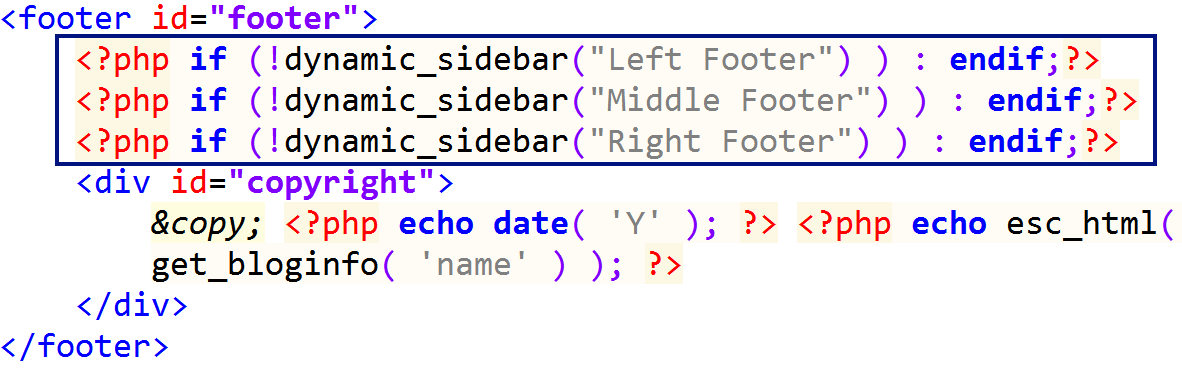
add\_action('widgets\_init', 'left\_footer\_widget');

function middle\_footer\_widget() {  
 register\_sidebar(array(  
 'name' => 'Middle Footer',  
 'id' => 'widget-footer-area-middle',  
 'before\_widget' => '<div class="middle-footer">',  
 'after\_widget' => '</div>',  
 'before\_title' => '<h3 class="footer-widget-title">',  
 'after\_title' => '</h3>',  
 ));  
}  
add\_action('widgets\_init', 'middle\_footer\_widget');  
  
function right\_footer\_widget() {  
 register\_sidebar(array(  
 'name' => 'Right Footer',  
 'id' => 'widget-footer-area-right',  
 'before\_widget' => '<div class="right-footer">',  
 'after\_widget' => '</div>',  
 'before\_title' => '<h3 class="footer-widget-title">',  
 'after\_title' => '</h3>',  
 ));  
}  
add\_action('widgets\_init', 'right\_footer\_widget');

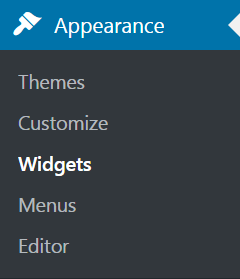
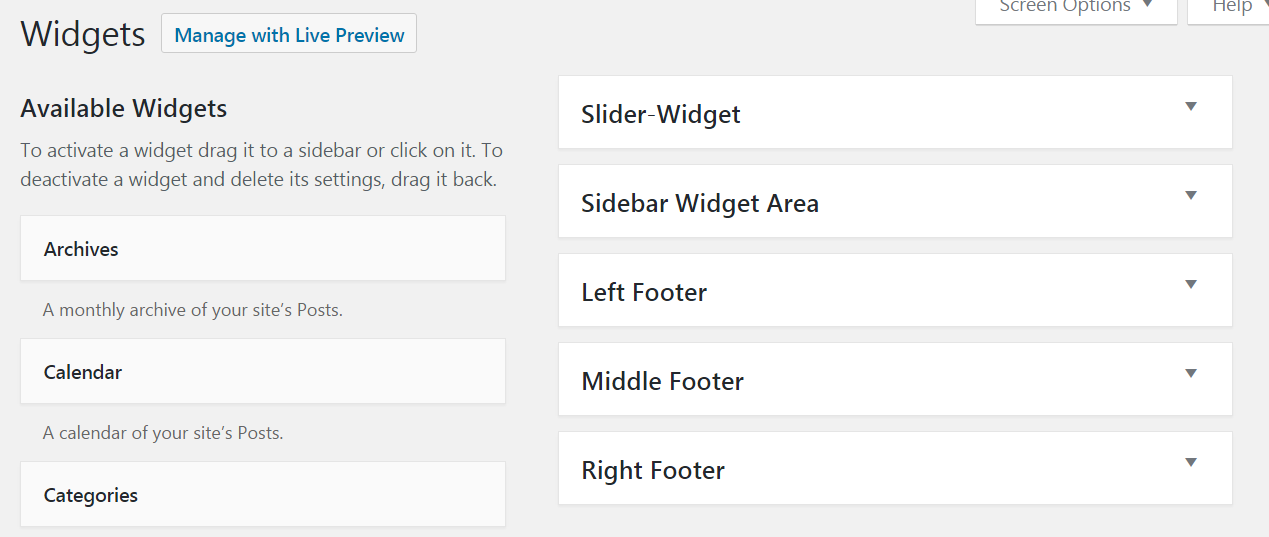
Now open footer.phpand add this code to display the footer widget areas’ content:

<?php if (!dynamic\_sidebar("Left Footer") ) : endif;?>  
<?php if (!dynamic\_sidebar("Middle Footer") ) : endif;?>  
<?php if (!dynamic\_sidebar("Right Footer") ) : endif;?>

See the code screenshot below to get a better idea:



Now you have 3 new widgets for the site the footer. Insert some content to be displayed in them:

Finally, add some CSS styles to display the footer widgets in 3 columns, as expected by design. Add this CSS code to style.css:

.left-footer,

.middle-footer,

.right-footer {

width: 20%;

margin-left: 12%;

display: inline-block;

vertical-align: top;

color: #E6E6E5;

}

.footer-widget-title {

font-size: 1.4em;

padding-top: 0;

margin-bottom: 20px;

}

#copyright {

padding-top: 20px;

padding-bottom: 20px;

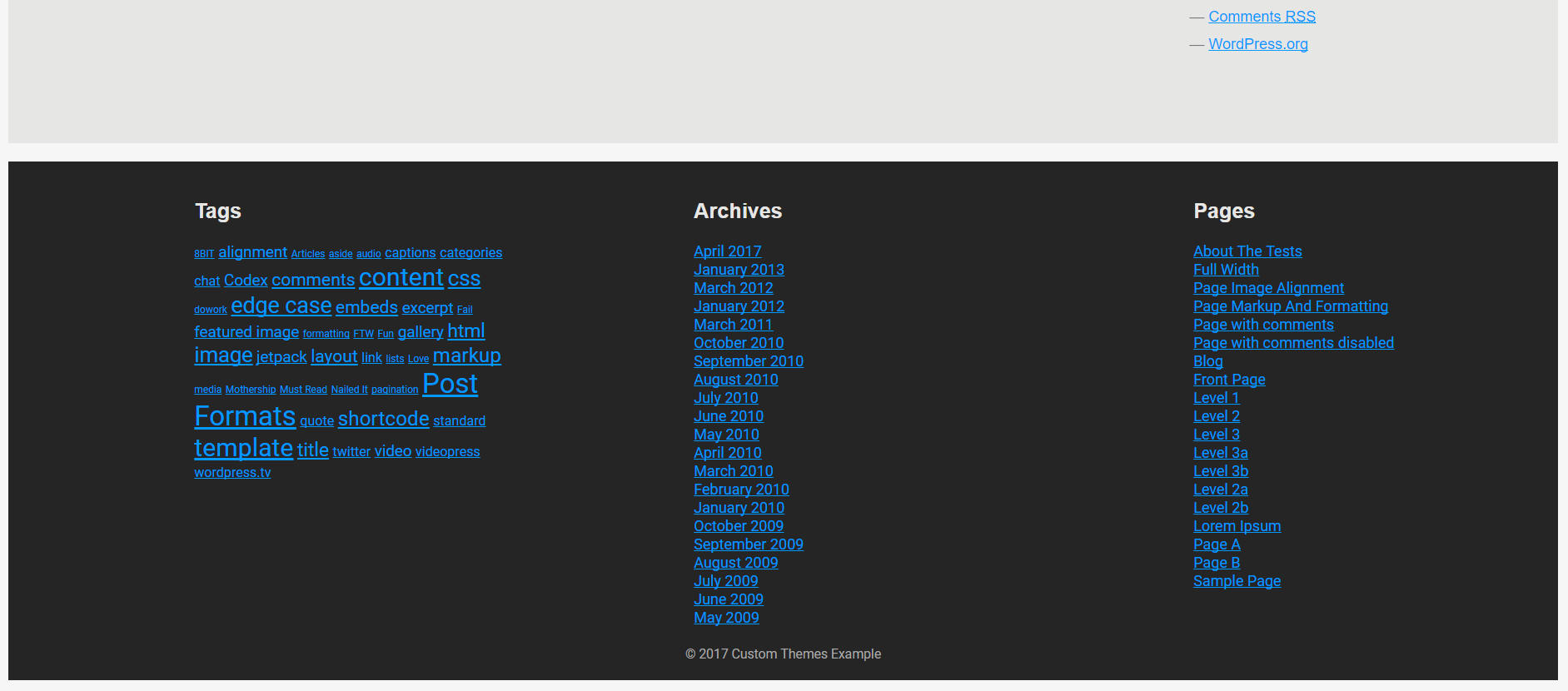
}

#footer {

background: #252525;

}

The footer should now look like this:



# What to Submit?

To ensure you have completed this homework, submit it at the course web site.

Create a **text file** or **Word document** or **PDF** (e.g. trifon-wordpress-homework.docx) holding your assets.

* Put in the file **screenshots** of all your site pages.
* Insert the screenshots in your homework document or upload them somewhere and provide links, e.g. <http://imgur.com/a/Z3SbB>.

Submit your text file (or document) as **homework** at the course Web site.