

|  |
| --- |
| Site project  2019 |
|  |
| JULY 20  CSS  Authored by: Danailov |



# Site Project with Jekyll & gulp

|  |
| --- |
| Static site Exam Project.  1. Requirements   # Project Assignment for the [CSS Advanced Course @ SoftUni](https://softuni.bg/trainings/2427/css-advanced-july-2019)  \*\*Design\*\* and \*\*implement\*\* a website about a topic of \*\*YOUR CHOICE\*\* using mainly \*\*HTML & CSS\*\*. The site should follow several requirements which are stated below. Any additional technologies and techniques that you use will be treated as \*\*bonuses\*\*. Do \*\*NOT SACRIFICE REQUIRED\*\* functionality for \*\*UNREQUIRED\*\* functionality. Implement first what is required and only then, think of additional things that can be added.  ## \*\*1. Requirements\*\*  \* \*\*Use HTML & CSS\*\* – the major part of your work should be HTML & CSS - \*\*written by you\*\*      \* You \*\*must\*\* use the \*\*CSS preprocessor - SASS *\_(Gulp)\_*\*\*      \* You \*\*must\*\* use the \*\*CSS Grids\*\*      \* You \*\*may\*\* use JavaScript and JS libraries like Slick and Sweet Alert to simplify your work      \* You \*\*may\*\* use PHP or other server-side scripting but not as primary technology (e.g. you may use PHP for including HTML fragments to avoid repeating code)  \* \*\*Work Alone\*\*      \* Use \*\*GitHub\*\* as project collaboration platform      \* You should have more than \*\*15 commits\*\* in \*\*5 different days\*\*  \* \*\*Create original content\*\* – your content should be written / created by you, not copy-pasted      \* You may take and adopt ideas from other sites, but avoid direct copy / paste of their content  \* \*\*Web design\*\*      \* Using your own web design (creative art work) is \*\*allowed\*\* but \*\*not required\*\*      \* You may use free images, icons and other site elements from Internet      \* You may use portions of other site's design, but don't copy / paste them entirely  \* \*\*Publish your project live in Internet\*\*      \* Your website should be \*\*hosted\*\* in [GitHub Pages](https://pages.github.com/)      \* You may share your project to get external feedback  \* \*\*Valid and high-quality HTML and CSS\*\*      \* Follow the \*\*best practices\*\* for high-quality HTML and CSS: good formatting, good code structure, consistent naming, semantic HTML, correct usage of classes, etc.  \* \*\*Responsive design\*\* – your site should open correctly and be functional at any modern device      \* Standard \*\*desktop\*\* Web browser (computer / laptop)      \* \*\*Tablets\*\* (Android, iPad, Windows, Linux)      \* \*\*Smartphones\*\* (Android, iPhone, Windows Phone)      \* Typical screen widths to support: 1600px, 1024px, 960px, 768px, 640px, 480px, 320px.  \* \*\*Usability and UX\*\*      \* Your site should be easy-to-use, with intuitive UI, with good usability      \* You may follow some usability guide / checklist [(search in Google)](https://www.google.com/search?q=web+site+usability+checklist)  \* \*\*Target all major modern browsers\*\*      \* Use \*\*HTML5 and CSS3\*\*, not HTML 4.01 or CSS 2.1      \* Ensure your site works correctly in the latest \*\*HTML5-compatible browsers\*\*: Chrome, Firefox, Edge, Opera, Safari (latest versions, desktop and mobile versions)      \* You do \*\*NOT\*\* need to support old browsers like IE9  ## \*\*2. Forbidden Techniques and Tools\*\*  \* Using HTML and CSS \*\*site templates\*\* from Internet is \*\*forbidden\*\*  \* Using \*\*CMS systems\*\* (like WordPress, Drupal and Joomla) is \*\*forbidden\*\*  \* Using \*\*site authoring tools\*\* and \*\*site generators\*\* (like Adobe Edge, Microsoft Expression Design, Adobe Dreamweaver, etc.) is \*\*forbidden\*\*  \* The use of \*\*client-side frameworks\*\* is forbidden      \* Your content should be in HTML files, not in a database or in the cloud  ## \*\*3. Work Alone\*\*  \* Your HTML and CSS should be written by yourself, not copied from somewhere      \* Use \*\*GitHub\*\* as your source-code repository and project management tool      \* Your \*\*source code\*\* should be published as \*\*open-source project\*\* in Internet  ## \*\*4. Deliverables\*\*  \* \*\*Source Code\*\* - Submit the \*\*URL\*\* of your project \*\*source code in GitHub\*\* as deliverable      \* Put the following assets in the project repository:          \* The complete \*\*source code\*\* of your project (HTML, CSS, images, scripts and other files)          \* Any other project assets (optionally): documentation, design, tests, etc  \* \*\*Published Live Project in Internet\*\*      \* Put a link to your \*\*live project\*\* in the project description at GitHub  \* \*\*Commit Logs\*\*      \* Each team member should have \*\*at least 15 commits\*\* (changes) in the project repository      \* Please \*\*commit every day\*\* during the project development to demonstrate your work progress      \* More commits (especially in more than the last 1-2 days) are better during the project assessment  ## \*\*5. Project Defense\*\*  Each project should be presented in front of a commission of trainers.  Each person will have only \*\*~10 minutes\*\* for the following:  \*   Demonstrate the website \*\*content and functionality\*\* (very shortly)  \*   Show the \*\*source code\*\* and explain briefly how it works  \*\*Hints\*\* for better presentation:  \*   Be \*\*well prepared\*\* for presenting maximum of your work for minimum time  \*   \*\*Open all project assets\*\* beforehand to save time: open your site in the browser, open your project repository page to show the commit logs, etc.  \*   Please be \*\*strict in timing!\*\* You might get less score if your time is not spent the right way  ## \*\*6. Assessment Criteria (Score)\*\*  \*   \*\*Site\*\* – \*\*0-10\*\* score      - Structure [more than 2 pages]      - Technical implementation      - Layout      - Design      - Usability  \*   \*\*Content\*\* (quality of content) – \*\*0-3\*\* score  \*   \*\*Responsive design\*\* (site behaves correctly in all devices, browsers and platforms) – \*\*0-5\*\* score  \* Using \*\*CSS\*\* transitions and animations – \*\*0-8\*\* score  \* Using \*\*Sass\*\* variables and mixins – \*\*0-8\*\* score  \* Using \*\*CSS Grids\*\* - \*\*0-8\*\* score  \*   \*\*GitHub\*\* - more than 15 contributions in 5 different days – \*\*0-6\*\* score  \* \*\*Bonus\*\* – \*\*0-5\*\* score |
|  |
| 1. Project sequence    1. Tool installation       1. Install **Full** Ruby development environment. - <https://rubyinstaller.org/downloads/>       2. Install Jekyll (under Windows   -from the command promt type: gem install jekyll bundler  For Jekyll installation.   * + 1. Install NodeJs - <https://nodejs.org/en/>     2. Install gulp from the command promt - npm install --global gulp-cli     3. Create folder for the project     4. From the command promt navigate to it and create new Jekyll Project * jekyll new myblog * Create pakage.json in the project directory - npm init * This will guide you through giving your project a name, version, description, etc. * Install gulp package in your project directory - npm install --save-dev gulp * Check the gulp version - gulp –version * Create gulpfile.js in the root of project folder with the following content:   "use strict";  let gulp = require("gulp"),      autoprefixer = require("gulp-autoprefixer"),      exec = require("gulp-exec"),      browserSync = require('browser-sync').create(),      sass = require('gulp-sass'),      cp = require("child\_process");  gulp.task("css", function() {      return gulp.src( '\_assets/scss/\*\*/\*.scss' )          .pipe( sass().on('error', sass.logError) )          .pipe( autoprefixer() )          .pipe( gulp.dest( './docs/css/' ) )          .pipe( browserSync.stream({ match: '\*\*/\*.css' }) )      ;  });  // Jekyll  gulp.task("jekylldev", function() {      return cp.spawn("bundle", ["exec", "jekyll", "build"], { stdio: "inherit", shell: true });  });  // Jekyll  gulp.task("jekyllprod", function() {      return cp.spawn("bundle", ["exec", "jekyll", "build --baseurl /ProjectSiteName"], { stdio: "inherit", shell: true });  });  gulp.task("watch", function() {      browserSync.init({          server: {  baseDir: "./docs/"          }      });      gulp.watch( '\_assets/scss/\*\*/\*.scss', gulp.series('css') );      gulp.watch(          [              "./\*.html",              "./\_includes/\*.html",              "./\_layouts/\*.html",              "./\_posts/\*\*/\*.\*"          ]      ).on('change', gulp.series('jekylldev', 'css') );      gulp.watch( 'docs/\*\*/\*.html' ).on('change', browserSync.reload );      gulp.watch( 'docs/\*\*/\*.js' ).on('change', browserSync.reload );  });  gulp.task("deploy", gulp.series('jekyllprod', 'css'));  gulp.task("default", gulp.series('jekylldev', 'css', 'watch')); |
| 1. *Site content*    1. *Site file and folder organization.* |

3.2 Some layout properties

Layouts are placed in folder \_Layouts

-main.html – layout for pages

-gallery.html – layout for galleries

For the gallery layout exist four different type of pictures

* Portrait
* Landscape
* Square
* Bigsquare
  1. All CSS files are transformed to SCSS and included in site.scss.

-every scss file correspond to corresponding html file.

(styles for about.html are in \_about.scss).

In files \_animations and \_transitions are used transition effects.

In file \_variables are places some common variables used in scss files.