

Dipen Parmar

2019

FlexSass Website Template

DEPARTMENT OF COMPUTER SCIENCE, ATMIYA UNIVERSITY
MASTER OF SCIENCE IN INFORMATION TECHNOLOGY

ACKNOWLEDGEMENT

I would like to highlight the fact that this project is the product of many contributors. First, I'd like to thank Professor **Amit Sagpariya** who accepted to work with me on my Project idea, supervising my work.

Second, I would like to thank the stack overflow community and Google for providing tremendous help to the elaboration of this project.

Index

- Project Structure	6
- FlexBox Basics & Terminology	7
- The two axes of flexbox	8
- The Main axes of flexbox	8
- The cross axis of flexbox	9
- Start and end lines (MainAxis)	10
- The flex Container	11
- Changing flex-direction	12
- What is BEM (Blocks, Elements and Modifiers)	13
- Benefits of BEM Methodology	13
- What is Sass	15
- What Can Sass/SCSS Do (Fetures)	16
- What is Node.js?	18
- Why Node.js?	18
- What Can Node.js Do?	19
- What is a Node.js File?	19
- What is NPM?	20
- Why Node.js?	20
- What is a Package?	20
- Use npm to.	21
- Sharing packages and collaborating with others	22
- What is Gulp	27
- The simple gulp API	28
- Installing gulp via npm	29
- Creating a gulpfile	30
- A practical example	31
- What is User Interface Design?	32
- Key Components of UI/UX Design	33
- Pug.js to make your life easier with HTML templates	49
- Pug is available via npm:	49
- Let's see how pug code looks like.	50

ABSTRAT

- ✓ **FlexSass** is a modern Sass WebTemplate, FlexBox is a new layout mode in CSS3, is Based on Latest The Flexible Box Module (CSS3),
- ✓
- ✓ Usually referred to as Flexbox , was designed as a one-dimensional layout model, and as a method that could offer space distribution between items in an interface and powerful alignment capabilities.
- ✓
- ✓ When working with flexbox you need to think in terms of two axes — the main axis and the cross axis. The main axis is defined by the flex-direction property, and the cross axis runs perpendicular to it. Everything we do with flexbox refers back to these axes, so it is worth understanding how they work from the outset.

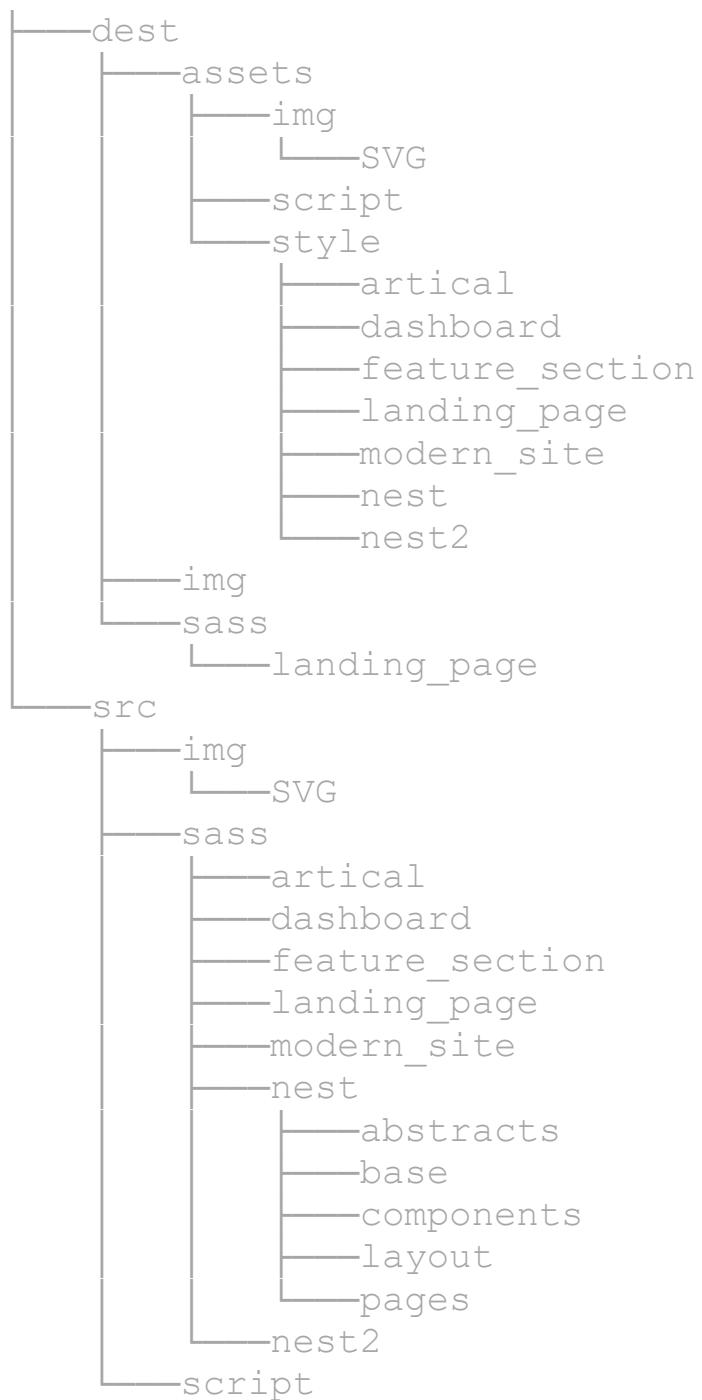
PROJECT PROFILE

- Developed By : Dipen Parmar
- Project on : General Website Template
- Sem-Year : 2stSem(2019)
- Guidance By : Mr. Amit Sagpariya
- Web Browser : Google Chrome
- Operating System : Windows 10
- Editor : Vscode, Sublime
- Submitted To : Atmiya University



Getting Started

- Project Structure



Css FlexBox

- FlexBox Basics & Terminology

The Flexible Box Module, usually referred to as flexbox, was designed as a one-dimensional layout model, and as a method that could offer space distribution between items in an interface and powerful alignment capabilities. This article gives an outline of the main features of flexbox, which we will be exploring in more detail in the rest of these guides.

When we describe flexbox as being one dimensional we are describing the fact that flexbox deals with layout in one dimension at a time — either as a row or as a column. This can be contrasted with the two-dimensional model of [CSS Grid Layout](#), which controls columns and rows together.

- The two axes of flexbox

When working with flexbox you need to think in terms of two axes — the main axis and the cross axis. The main axis is defined by the flex-direction property, and the cross axis runs perpendicular to it. Everything we do with flexbox refers back to these axes, so it is worth understanding how they work from the outset

- The Main axes of flexbox

The main axis is defined by flex-direction, which has four possible values:

- row
- row-reverse
- column
- column-reverse

Should you choose `row` or `row-reverse`, your main axis will run along the row in the **inline direction**



Choose `column` or `column-reverse` and your main axis will run from the top of the page to the bottom — in the **block direction**

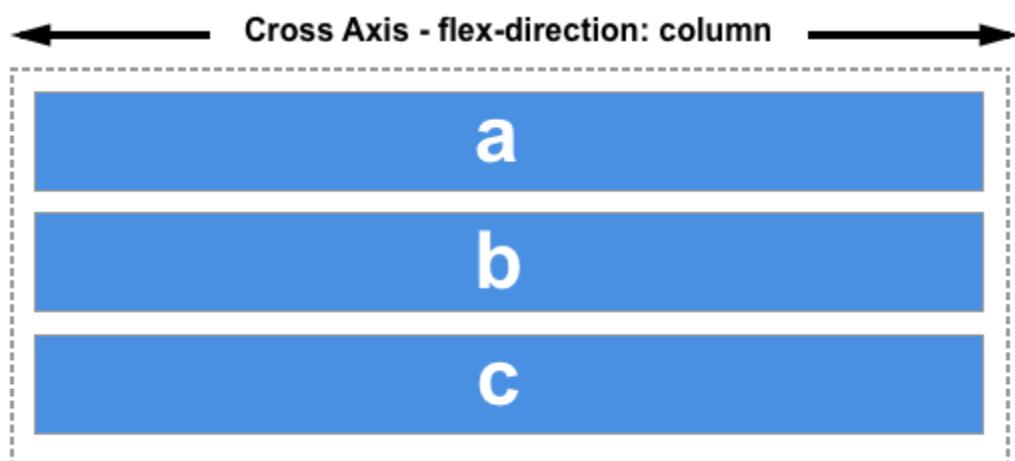


- The cross axis of flexbox

The cross axis runs perpendicular to the main axis, therefore if your `flex-direction` (main axis) is set to `row` OR `row-reverse` the cross axis runs down the columns.



If your main axis is `column` OR `column-reverse` then the cross axis runs along the rows.



Understanding which axis is which is important when we start to look at aligning and justifying flex items; flexbox features properties that align and justify content along one axis or the other.

- Start and end lines (MainAxis)

Another vital area of understanding is how flexbox makes no assumption about the writing mode of the document. In the past, CSS was heavily weighted towards horizontal and left-to-right writing modes. Modern layout methods encompass the range of writing modes and so we no longer assume that a line of text will start at the top left of a document and run towards the right hand side, with new lines appearing one under the other.

You can read more about the relationship between flexbox and the Writing Modes specification in a later article, however the following description should help explain why we do not talk about left and right and top and bottom when we describe the direction that our flex items flow in.

If the `flex-direction` is `row` and I am working in English, then the start edge of the main axis will be on the left, the end edge on the right.



If I were to work in Arabic, then the start edge of my main axis would be on the right and the end edge on the left.



In both cases the start edge of the cross axis is at the top of the flex container and the end edge at the bottom, as both languages have a horizontal writing mode.

After a while, thinking about start and end rather than left and right becomes natural, and will be useful to you when dealing with other layout methods such as CSS Grid Layout which follow the same patterns.

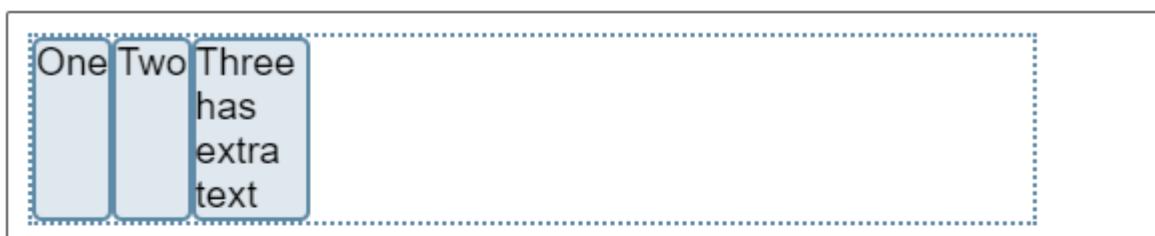
- The flex Container

An area of a document laid out using flexbox is called a **flex container**. To create a flex container, we set the value of the area's container's display property to `flex` or `inline-flex`. As soon as we do this the direct children of that container become **flex items**. As with all properties in CSS, some initial values are defined, so when creating a flex container all of the contained flex items will behave in the following way.

- Items display in a row (the `flex-direction` property's default is `row`).
- The items start from the start edge of the main axis.
- The items do not stretch on the main dimension, but can shrink.
- The items will stretch to fill the size of the cross axis.
- The `flex-basis` property is set to `auto`.
- The `flex-wrap` property is set to `nowrap`.

The result of this is that your items will all line up in a row, using the size of the content as their size in the main axis. If there are more items than can fit in the container, they will not wrap but will instead overflow. If some items are taller than others, all items will stretch along the cross axis to fill its full size.

You can see in the live example below how this looks. Try editing the items or adding additional items in order to test the initial behavior of flexbox.



```
.box {  
    display: flex;  
}
```

```
<div class="box">  
    <div>One</div>  
    <div>Two</div>  
    <div>Three  
        <br>has  
        <br>extra  
        <br>text  
    </div>  
</div>
```

- Changing flex-direction

Adding the `flex-direction` property to the flex container allows us to change the direction in which our flex items display. Setting `flex-direction: row-reverse` will keep the items displaying along the row, however the start and end lines are switched.

If we change `flex-direction` to `column` the main axis switches and our items now display in a column. Set `column-reverse` and the start and end lines are again switched.

The live example below has `flex-direction` set to `row-reverse`. Try the other values — `row`, `column` and `column-reverse` — to see what happens to the content.

```
.box {
  display: flex;
  flex-direction: row-reverse;
}

<div class="box">
  <div>One</div>
  <div>Two</div>
  <div>Three</div>
</div>
```

BEM

- What is BEM (Blocks, Elements and Modifiers)

CSS is a language that is easy to learn but very hard to maintain. As the project grows larger, without proper structure, maintaining CSS is unbearable. People have come up with different types of solutions to this such as OOCSS, SMACSS, and BEM. Currently, BEM is the most widely used, it's unique naming method makes CSS to maintain. Without further a due, let's start learning BEM.

BEM is a naming styling that is created by Yandex (think of them as Russia's Google). The problem BEM is trying to solve is the naming problem and structure that CSS often run into. BEM also provides a better structure for your CSS code and scalable CSS.

- Benefits of BEM Methodology

- **Modularity**
 - Block styles are never dependent on other elements on a page, so you will never experience problems from cascading.
 - You also get the ability to transfer blocks from your finished projects to new ones.
- **Reusability**
 - Composing independent blocks in different ways, and reusing them intelligently, reduces the amount of CSS code that you will have to maintain.
 - With a set of style guidelines in place, you can build a library of blocks, making your CSS super effective.
- **Structure**
 - BEM methodology gives your CSS code a solid structure that remains simple and easy to understand.

Block

Standalone entity that is meaningful on its own.

Examples

```
header , container , menu , checkbox , input
```

Element

A part of a block that has no standalone meaning and is semantically tied to its block.

Examples

```
menu item , list item , checkbox caption , header  
title
```

Modifier

A flag on a block or element. Use them to change appearance or behavior.

Examples

```
disabled , highlighted , checked , fixed , size  
big , color yellow
```



SASS

- What is Sass

Sass (Syntactically Awesome Style Sheets) is an extension of CSS that enables you to use things like variables, nested rules, inline imports and more. It also helps to keep things organised and allows you to create style sheets faster.

Sass is compatible with all versions of CSS. The only requirement for using it is that you must have Ruby installed. Users are also asked to follow the Sass

- SCSS (Sassy CSS): Uses the .scss file extension and is fully compliant with CSS syntax
- Indented (simply called 'Sass'): Uses .sass file extension and indentation rather than brackets; it is not fully compliant with CSS syntax, but it's quicker to write



Awesome Sass

- What Can Sass/SCSS Do (Features)

1. Nested Rules — Nest your CSS properties within multiple sets of {} brackets. This makes your CSS code a bit more clean-looking and more intuitive.

2. Variables — Standard CSS has variable definitions. So what's the deal? You can do a lot more with Sass variables: iterate them via a for-loop and generate property values dynamically. You can embed them into CSS property names themselves. It's useful for property-name-N { ... } definitions.

3. Better Operators. You can add, subtract, multiple and divide CSS values. Sure the original CSS implements this via calc() but In Sass you don't have to use calc() and implementation is slightly more intuitive.

4. Functions — Sass lets you create CSS definitions as reusable functions.

5. Trigonometry — Among many of its basic features (+, -, *, /) SCSS allows you to write your own functions. You can write your own sine and cosine (trigonometry) functions entirely using just the Sass/SCSS syntax just like you would in other languages such as JavaScript.

Some trigonometry knowledge will be required. But basically, think of sine and cosine as mathematical values that help us calculate the motion of circular progress bars or create animated wave effects for example.

6. for-loops, while-loops, if-else statements. You can write CSS using familiar code-flow and control statements similar to another languages. But don't be fooled, Sass still results in standard CSS in the end. It only controls how property and values were generated. It's not a real-time language. Only a pre-processor.

7. Mixins. Create a set of CSS properties once and reuse them or “mix” together with any new definitions. In practice, you can use mixins to create separate themes for the same layout, for example.

Node (Backend)

You will need a working version of Node and NPM to get started.



NodeJs

- What is Node.js?

- Node.js is an open source server environment
- Node.js is free
- Node.js runs on various platforms (Windows, Linux, Mac OS X, etc.)
- Node.js uses JavaScript on the server

- Why Node.js?

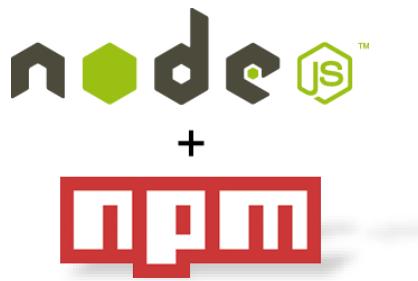
- Node.js uses asynchronous programming!
- Here is how PHP or ASP handles a file request:
 - Sends the task to the computer's file system.
 - Waits while the file system opens and reads the file.
 - Returns the content to the client.
 - Ready to handle the next request.
- Here is how Node.js handles a file request:
 - Sends the task to the computer's file system.
 - Ready to handle the next request.
 - When the file system has opened and read the file, the server returns the content to the client.

- What Can Node.js Do?

- Node.js can generate dynamic page content
- Node.js can create, read, write, delete, and close files on the server
- Node.js can collect form data
- Node.js can add, delete, modify data in your database

- What is a Node.js File?

- Node.js files contain tasks that will be executed on certain events
- A typical event is someone trying to access a port on the server
- Node.js files must be initiated on the server before having any effect
- Node.js files have extension ".js"



Node Package Manager

- What is NPM?

- Node.js is an open source server environment
- Node.js is free
- Node.js runs on various platforms (Windows, Linux, Mac OS X, etc.)
- Node.js uses JavaScript on the server

- Why Node.js?

- NPM is a package manager for Node.js packages, or modules if you like.
- www.npmjs.com hosts thousands of free packages to download and use.
- The NPM program is installed on your computer when you install Node.js

- What is a Package?

- A package in Node.js contains all the files you need for a module.
- Modules are JavaScript libraries you can include in your project.

- Use npm to..

- Adapt packages of code for your apps, or incorporate packages as they are.
- Download standalone tools you can use right away.
- Run packages without downloading using [npx](#).
- Share code with any npm user, anywhere.
- Restrict code to specific developers.
- Create Orgs (organizations) to coordinate package maintenance, coding, and developers.
- Form virtual teams by using Orgs.
- Manage multiple versions of code and code dependencies.
- Update applications easily when underlying code is updated.
- Discover multiple ways to solve the same puzzle.
- Find other developers who are working on similar problems and projects.

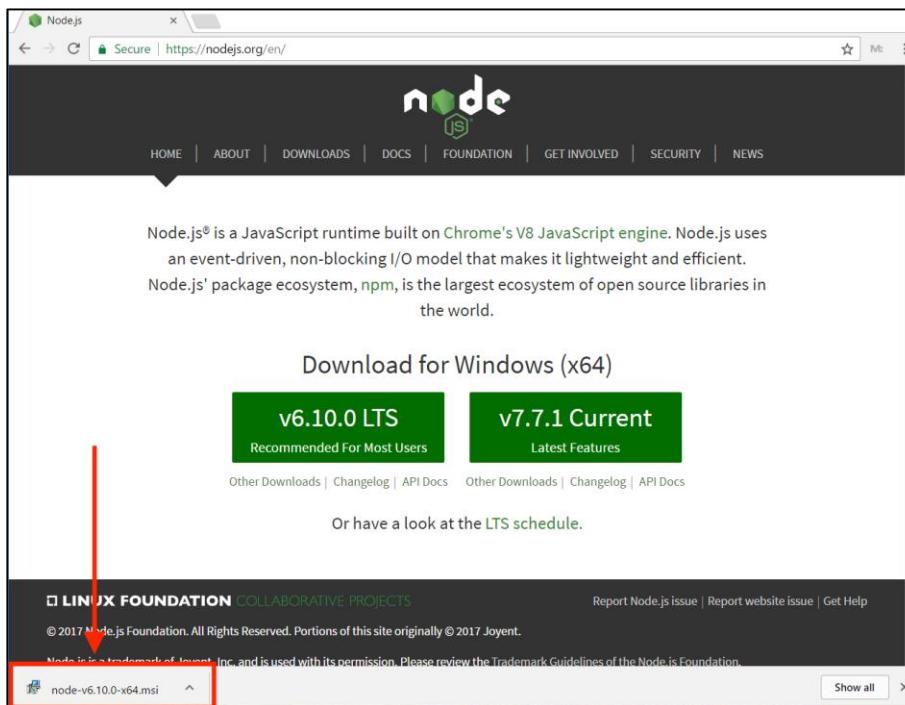
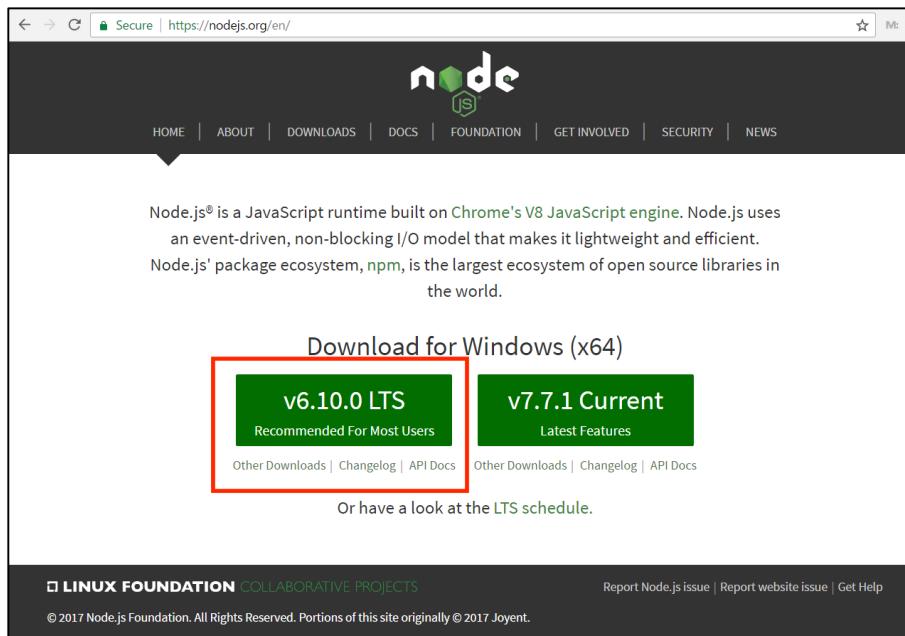
-

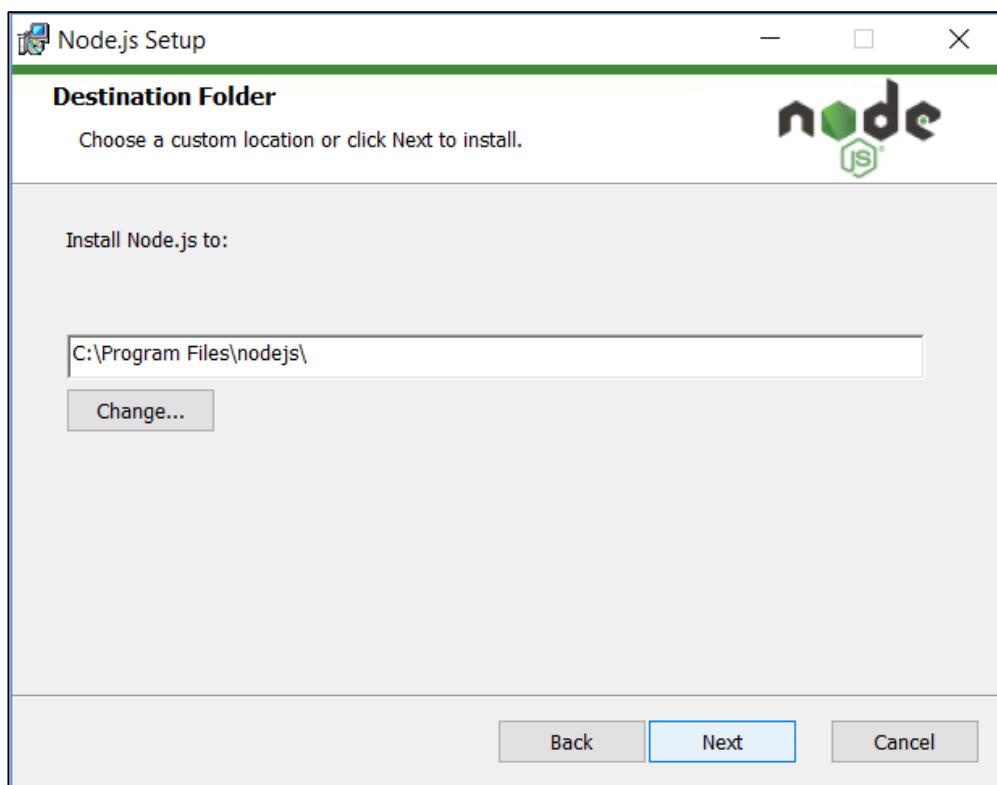
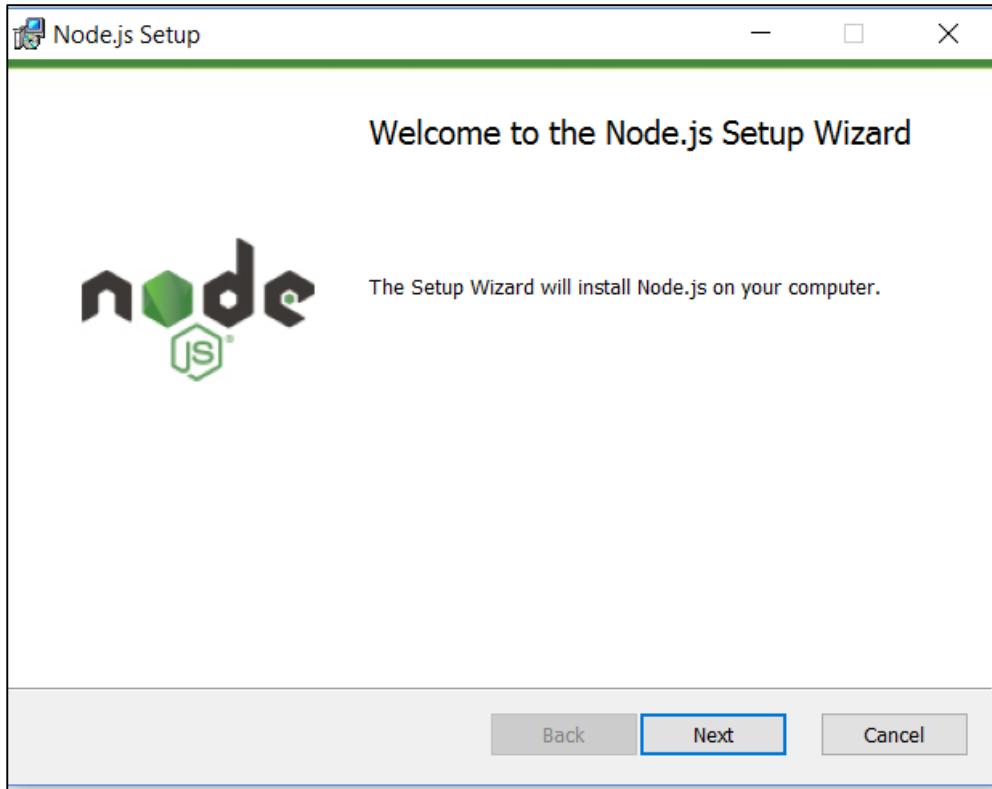
- Sharing packages and collaborating with others

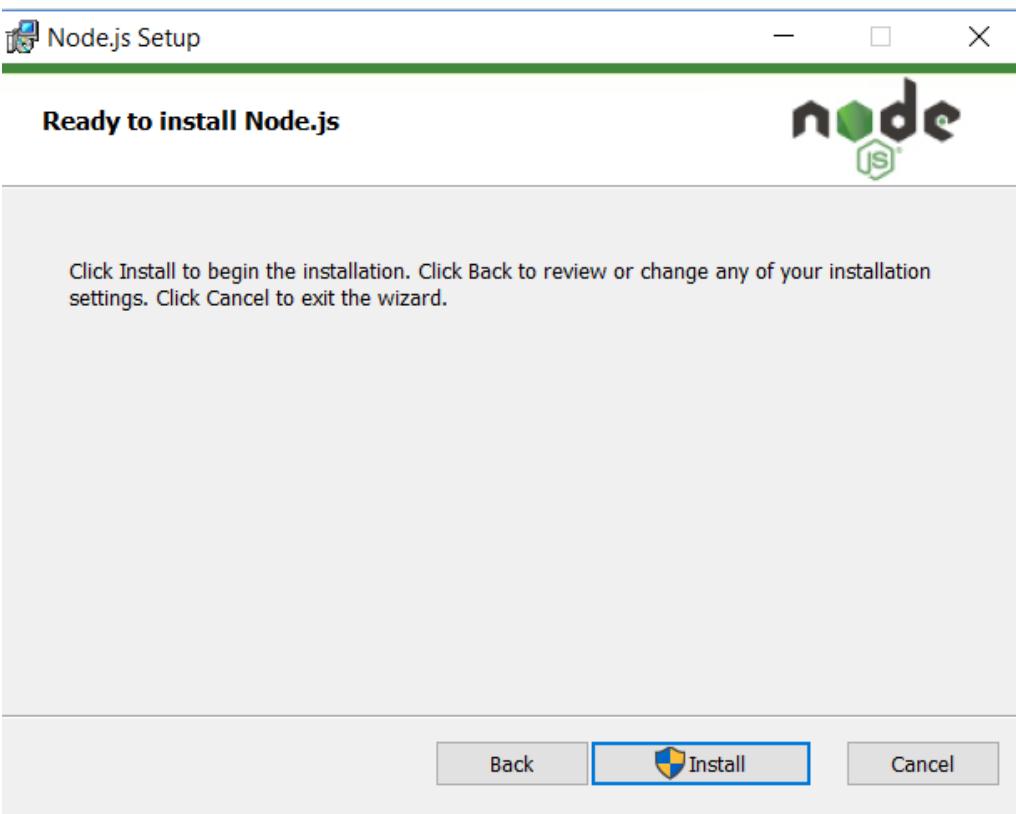
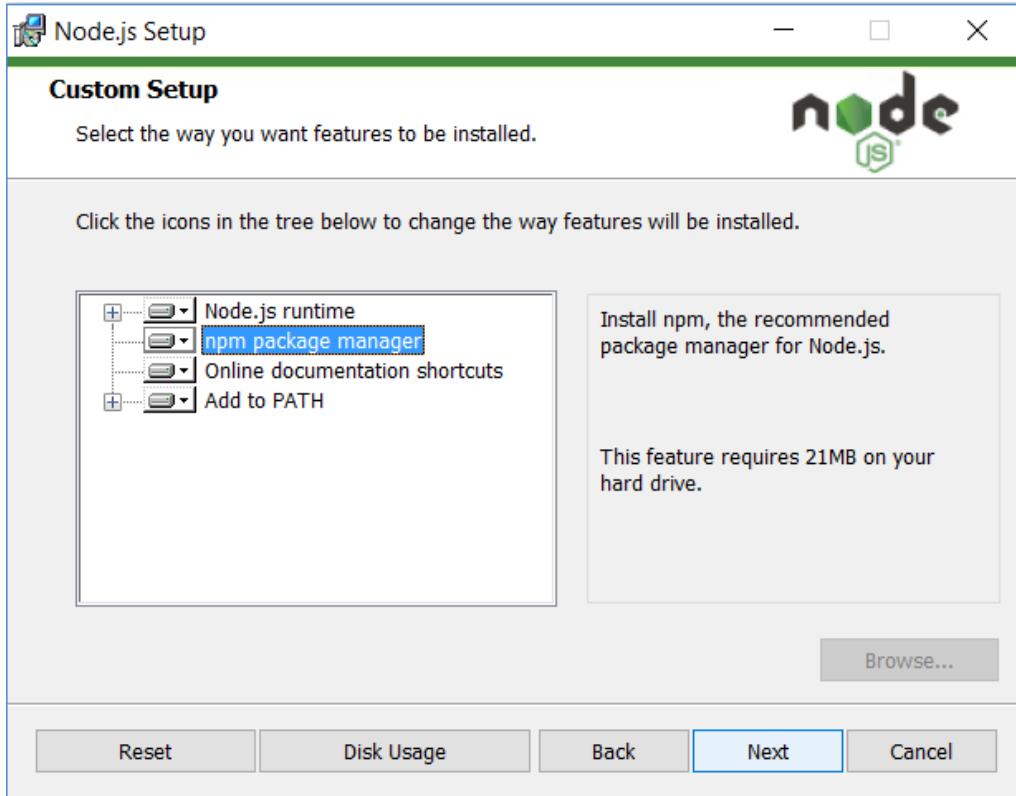
If you choose to share your packages publicly, there is no cost. To use and share private packages, you need to upgrade your account. To share with others, create organizations, called **npm Orgs**, and invite others to work with you, privately (for a fee) or publicly (for free). Or you can sign up for a private instance of npm for your company, called **npm Enterprise**, so you can develop packages internally that are not shared publicly.

Install Node.js & NPM on Windows

Go to the [official Node.js site](https://nodejs.org/en/) which will offer you two versions of Node to download. Click on the Installer that says “Recommended for Most Users”.

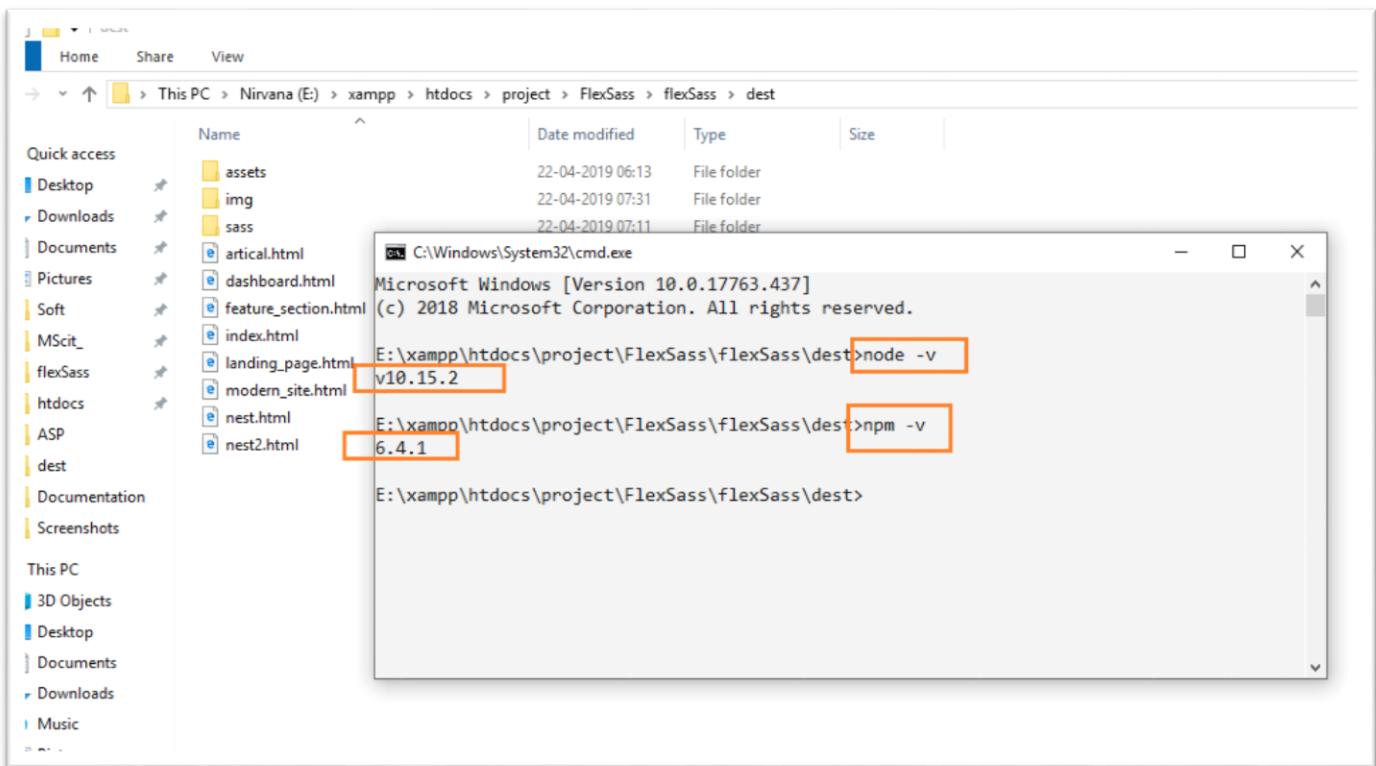






In The Command Prompt. To confirm Node is installed type `node -v` which will print the current version number. To

confirm NPM is installed type npm -v which will print its current version number.





Gulp

- What is Gulp

- Gulp is a JavaScript task runner that lets you automate tasks such as...
- Bundling and minifying libraries and stylesheets.
- Refreshing your browser when you save a file.
- Quickly running unit tests
- Running code analysis
- Less/Sass to CSS compilation
- Copying modified files to an output directory
- -----
- Stream-based build system
- NodeJs
- Code (JavaScript) over configuration
- Asynchronous
- Simple API
- Plugin based
- Easy

- The simple gulp API

Using gulp is super simple because you don't have to figure out how a complex API works in order to be productive with it. There are only 4 API's in gulp!

API	Purpose
gulp.task	Define a task
gulp.src	Read files in
gulp.dest	Write files out
gulp.watch	Watch files for changes

- Installing gulp via npm

The npm package manager comes installed with Node.js. While node.js isn't a requirement to use gulp, it does make demonstrating it a lot easier. I will be installing gulp from npm locally into my project. Make sure that you're in your project's root folder before running the command, otherwise your node modules will be downloaded into the wrong folder

```
cd myproject  
npm install --save-dev gulp
```

This will install the gulp node module locally to the project (as opposed to globally). The `--save-dev` argument lets npm know to update its package.json file with a new devDependencies record. devDependencies will need to be resolved at development time, whereas dependencies will need to be resolved at run time. Because gulp is a tool to aid us in development, it needs to be resolved at development time.

- Creating a gulpfile

A gulpfile is a file that will act as a manifest to define our tasks. Tasks that we want to execute will be found within this file. Whenever we run the command `gulp hello-world` from the command line, we are telling gulp that we want to run the hello-world task within gulpfile.js. After creating `gulpfile.js` within the root of your project, add a basic tasks.

```
var gulp = require('gulp');

gulp.task('hello-world', function(){
    console.log('hello world');
});
```

`require` is a function implemented by node (which is an implementation of the CommonJS spec) that will add references to node modules that we have installed. Once we make a reference to the gulp module, we can use it to create a task. Here, our task simply writes to the console window, but you could have it do any number of automated tasks.

- A practical example

Create a gulp task (or set of gulp tasks) for cleaning directories and compiling less to CSS. Such a set of tasks may look like the following. This set of tasks also has its own set of dependencies, which would require using `npm install` to get installed locally

```
function clean(path, done) {
  log('Cleaning: ' + plugins.util.colors.blue(path));
  // utilizes del node module
  del(path, cb);
}

gulp.task('clean-styles', function(cb) {
  var files = './styles/*.css'; // glob pattern to any existing css
  clean(files, cb);

  // cb is called because clean uses del,
  // which does not use gulp streams.
});

// 2nd param signifies task dependencies
// in this case, clean-styles will run first.
gulp.task('styles', ['clean-styles'], function() {
  log('Compiling Less --> CSS');
  return gulp
    .src('./src/styles/*.less')
    .pipe(plugins.less()) // utilizes gulp-less to compile less -> css

    // utilizes gulp-autoprefixer to add
    // browser specific css prefixes
    .pipe(plugins.autoprefixer())

    // copies the generated css to a destination folder.
    .pipe(gulp.dest('./styles/*.css'));
});
```

Running the following in the command line starts the watch:

```
$ gulp styles
```

Hopefully it's obvious why taking the time to create gulp tasks for automating repetitive development tasks can increase productivity.

USER INTERFACE

- What is User Interface Design?

User Interface (UI) Design is the link between users and your website. It includes the basic design elements that need to be present in order for someone to navigate your site and make decisions. It is the ever-evolving relationship between a person and the system that they are using. It includes the way that your website interacts with users, the overall design and how information is presented.

There are many different ways that you can look at user interface, but the basics always include the communication from a product to the user and vice versa1.

UI Design is all about structure, user manipulation and communication. This is one of the reasons it's so important that you pay close attention to it. It is the basic building blocks of how your website is set up and functions when visited by your target audience. If it doesn't go smoothly, problems tend to follow.

Some of the elements involved in User Interface Design include the input controls, navigational components, informational components and containers4. The devil is in the details, as they say. Buttons, lists, toggles, icons, breadcrumbs, tags and more also play a huge role. It is the way that you design your site from the back-end so that your audience can have a seamless, and enjoyable, front-end experience.

There are 3 issues / parts of a user interface design, these being

- how it looks,
- how it works,
- how easy it is to use.

- Key Components of UI/UX Design



There are some core components of UX Design, which are as follows:

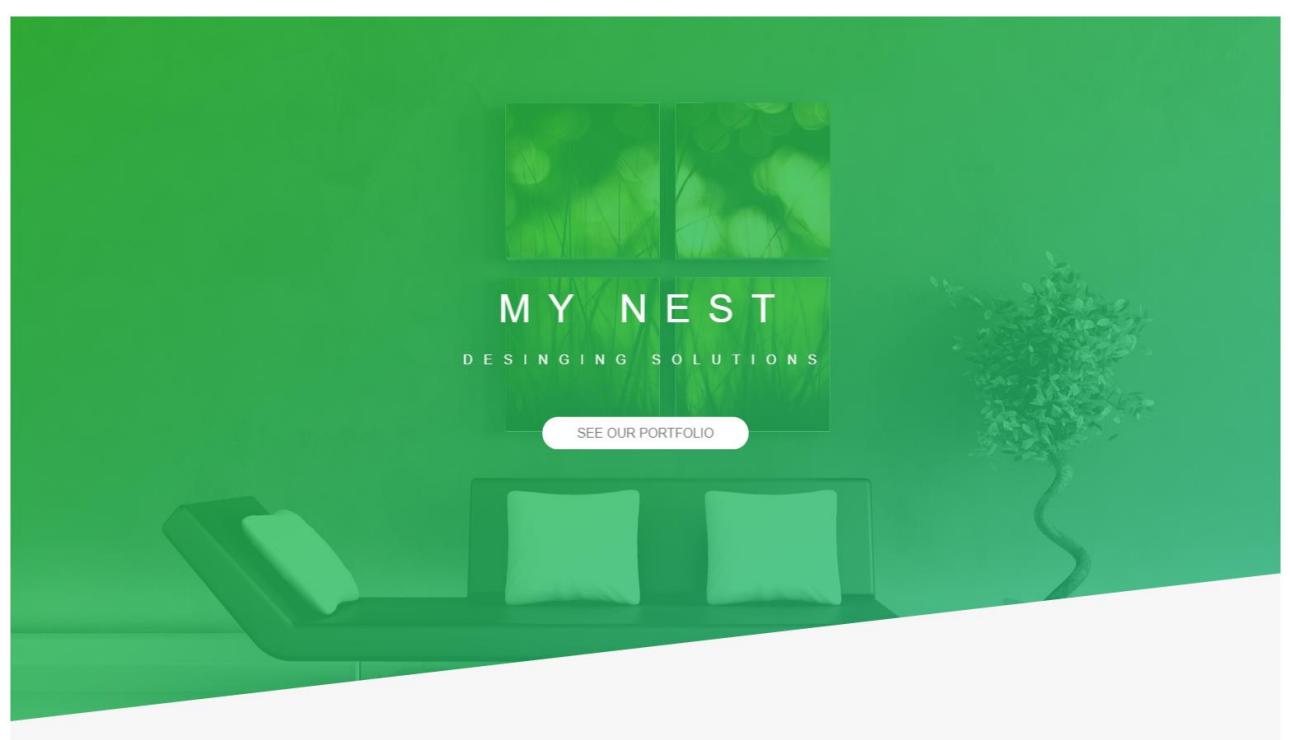
- Information Architecture
- Interaction Design
- Usability
- Wire framing
- Visual Design

WireFrame

WirFrames



CSS | OutPut



PAGE TWO HEADING H2

SECONDARY HEADING

Lore ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec erat

SECONDARY HEADING

Lore ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus,

DESIGN TO YOUR STYLE WITH MY NEST

WHO WE ARE

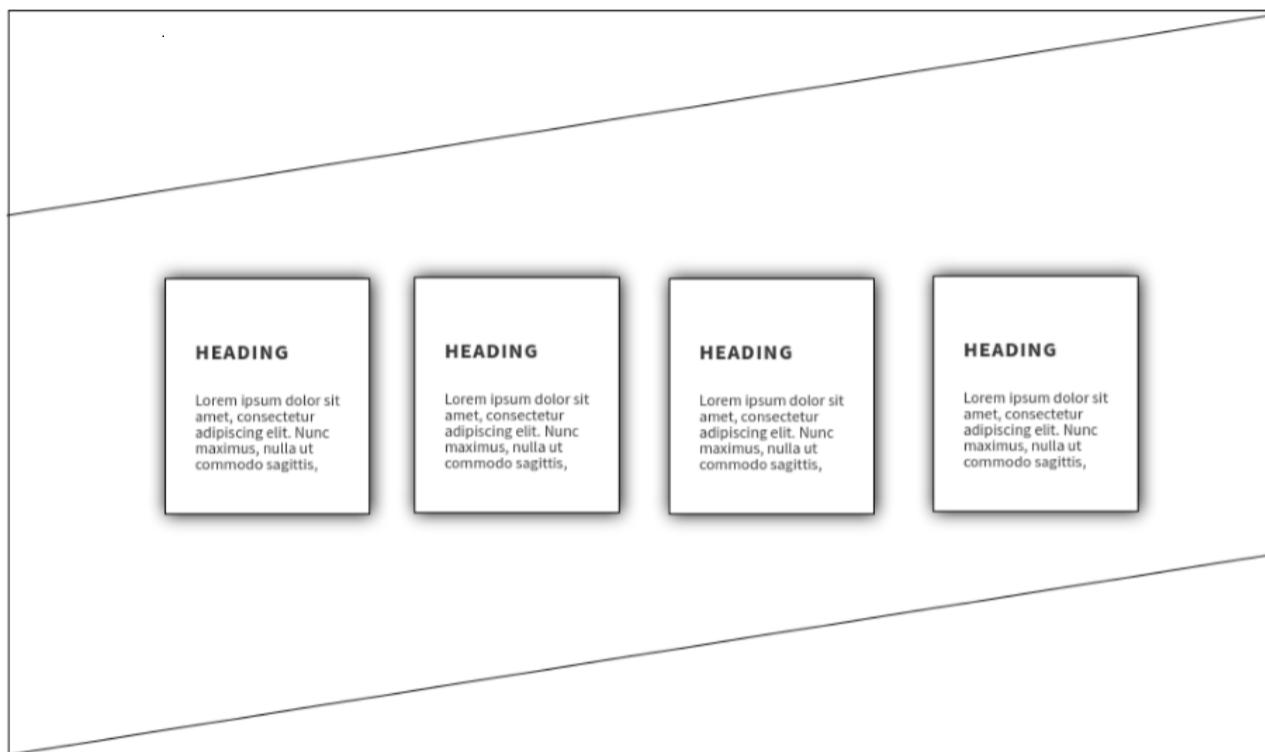
Our aim is to lend a helping hand to all our customers in achieving their design goals and guiding them effortlessly through to the final, successful conclusion. please visit "What We Do" page to see the wide range or services we offer..

WHAT IS IT

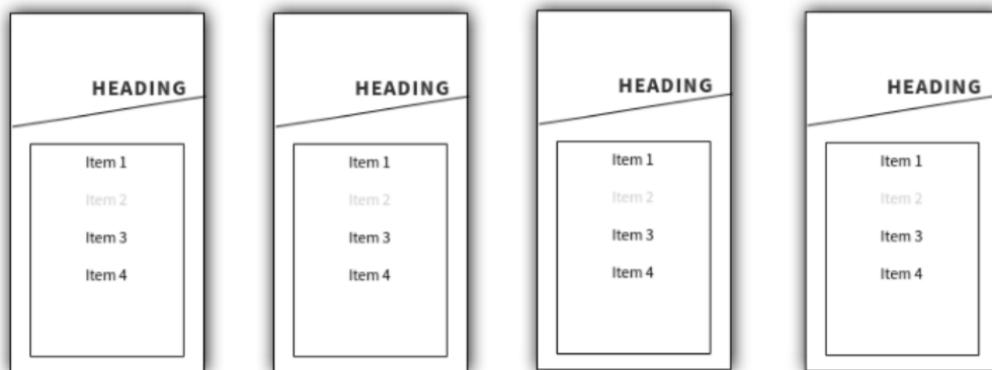
Lore ipsum dolor sit amet consectetur adipiscing elit. Deleniti omnis dolore quasi? Sapiente ratione in.

[More about us →](#)





CARDS GRID PAGE TWO HEADING



Button Click

OUR BEST PLAN AND PRICES (MY NEST)



Color
Furniture
heading-primary
1GB Storage
500%GB HDD



Color
Furniture
heading-primary
1GB Storage
500%GB HDD



Color
Furniture
heading-primary
1GB Storage
500%GB HDD



Color
Furniture
heading-primary
1GB Storage
500%GB HDD

SEE OUR ALL PLANS

PAGE THREE WEBSITE OWNER_

BIG USER
PICTURE



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec et commodo sagittis, sapien dui



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec et commodo sagittis, sapien dui



REVIEWS AND CUSTOMERS FEEDBACK

Dipen Parmar



HELLO, I AM DIPEN PARMAR

Our aim is to lend a helping hand to all our customers in achieving their design goalsto the final, successful conclusion. please visit "What We Do" page to see the wide range or services we offer. design goalsto the final, successful conclusion. please visit "What We Do" page to see the wide



HELLO, I AM DIPEN PARMAR

Our aim is to lend a helping hand to all our customers in achieving their design goalsto the final, successful conclusion. please visit "What We Do" page to see the wide range or services we offer. design goalsto the final, successful conclusion. please visit "What We Do" page to see the wide on. please visit We Do" page to see the wide

FORM ATMIAY BOX LOREM

Button Click

BOOK YOUR SWEET HOME WITH US!

Full name

Email Address

Submit





[Link 1](#) | [Link 2](#) | [Link 3](#) | [Link 4](#)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec erat. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mat



[TERMS](#) [LINK1](#) [LINK2](#) [D_CSS](#) [D_SWIFT](#) [D_PHP](#)

Hello this is **DIPEN PARMAR** From any wareLorem ipsum
[FACEBOOK](#) sit amet consectetur adipisciing elit. Magnam
[GOOGLE+](#) labore provident reprehenderit, at, sapiente ipsum
debitis earum officia illum ab, fugit veritatis consequatur.
Necessitatibus consectetur porro sequi autem.

ATMIYA UNIVERSITY
MSC- IT
Lorem Ipsum hello World

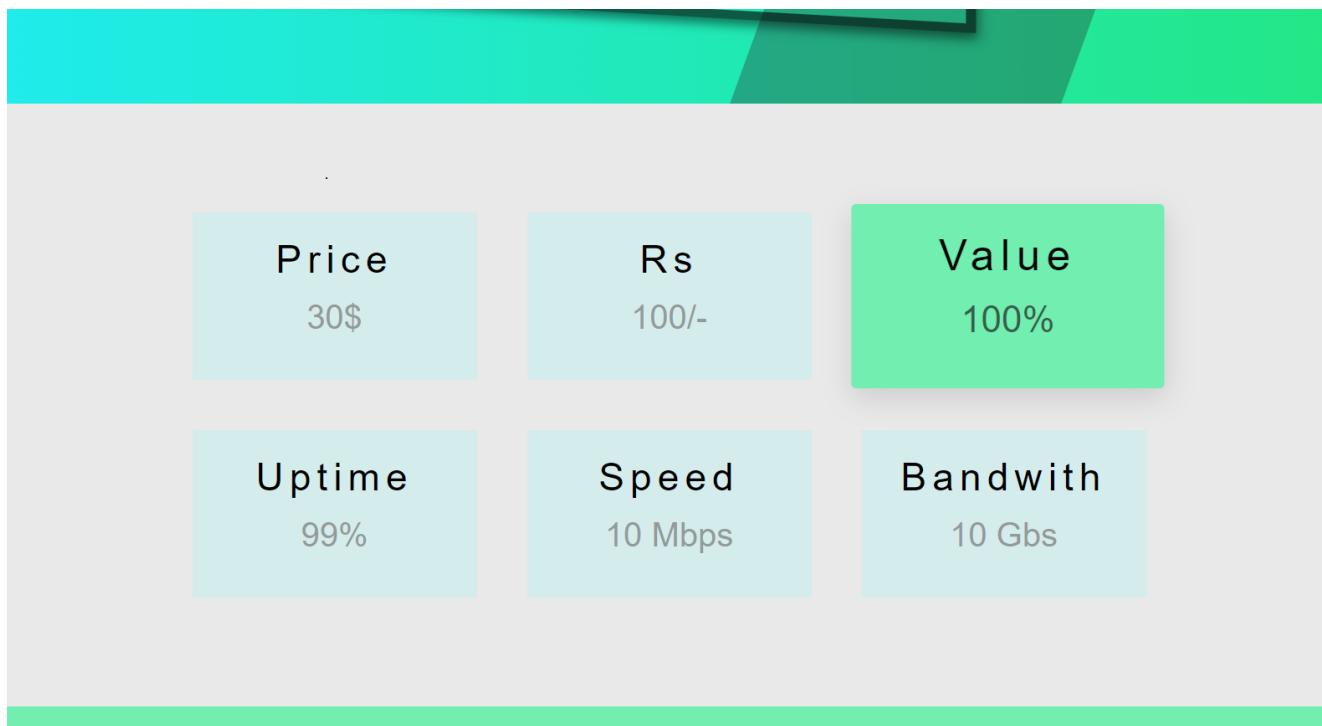
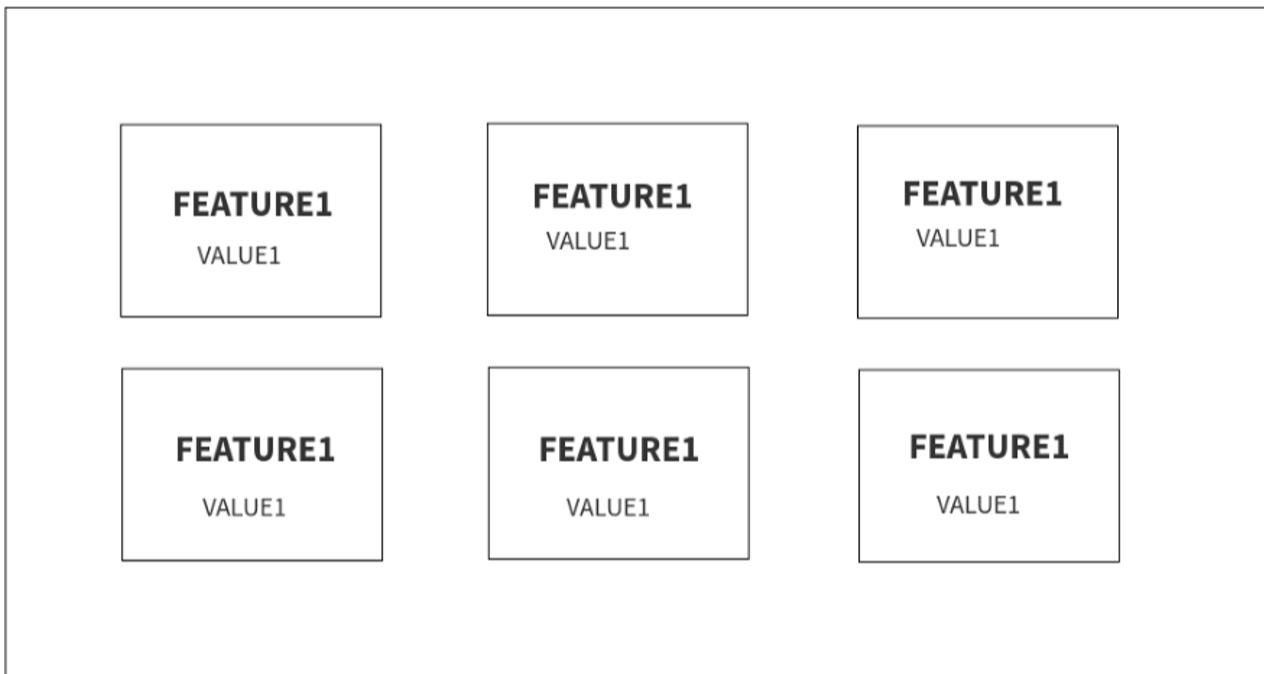
HEADING

LINK1
LINK1
LINK1

ATMIYA UNIVERSITY
MSC
- DEPARTMENT OF COMPUTER SCIENCE

HEADING

CLICK_MSCIT
ABOUT
CONTACT
DOWNLOAD
FAQ



ATMIAY BOX

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec erat

FOOTER

Atmiya BOX

Lorem dolor consectetur adipisicing elit. Culpa ipsa nisi, nesciunt consequatur sequi nihil ex nobis!

Dolore eos totam, molestiae ipsa modi libero nihil! Qui culpa animi necessitatibus dolore!

Design & Developed By Dipen Parmar

LOGO

 USER NAME —

- Item 1
- Item 2
- Item 3
- Item 4

DASHBOARD PAGE SECONDRY HEADING

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec erat. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec erat. Aliquam egestas, velit at condimentum placerat, sem sapien laoreet mauris, dictum porttitor lacus est nec enim. Vivamus feugiat elit lorem, eu porttitor ante ultrices id.

YOUR LOGO
Search Item

MY FEV
 LINUX
 DASHBOARD
 TWITTER
 INSTAGRAM
 LOG
 PIA CHART
 CHROME
 SHUTDOWN

THIS IS WHAT WE DO!

Lorem ipsum dolor sit amet **This is what i love** consectetur adipisicing elit. Dolor non distinctio natus perspiciatis, eius esse accusantium debitis omnis amet, at sed quae odio? Cupiditate provident unde reprehenderit.
Dipends THemes Lorem ipsum donderit.

Lorem ipsum dolor sit amet consectetur adipisicing elit. Dolor non distinctio natus perspiciatis, eius esse accusantium debitis omnis amet, at sed blanditiis laborum minima quae odio? Cupiditate provident unde reprehenderit.

Project Features

- > Hello World.
- > You're very beautiful
- > Hello World program
- > You look as pretty as always

Testimonials

WELCOME TO ATMIYA UNIVERSITY

SubHeading

**FEATURE1**

Lore ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut co

**FEATURE1**

Lore ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut co

SubHeading 2

**FEATURE1**

Lore ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut co

**FEATURE1**

Lore ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut co

SubHeading 3

**FEATURE1**

Lore ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut co

**FEATURE1**

Lore ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut co

**FEATURE1**

Lore ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut co

Welcome to Atmiya University

Get Started

**Let's Getting Started**

Get introduced to the basic concepts behind HTML & CSS, the box model, the webflow Designer UI and more

**Let's Getting Started**

Get introducemodel, Lorem ipsum Designer Ui Designer UI and more

Design

**Let's Getting Started**

Get introducemodel, Lorem ipsum Designer Ui Designer UI and more

**Let's Getting Started**

Get introduced to the basic concepts behind HTML & CSS, the box model, the webflow Designer UI and more

**Let's Getting Started**

Get introduced to the basic concepts behind HTML & CSS, the box model, the webflow Designer UI and more

Set up

**Let's Getting Started**

Get introduced to the basic concepts behind HTML & CSS, the box model, the webflow Designer UI and more

**Let's Getting Started**

Get introducemodel, Lorem ipsum Designer Ui Designer UI and more

LOGO

 USER NAME

- Item 1
- Item 2
- Item 3
- Item 4

DASHBOARD PAGE SECONDRY HEADING

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc maximus, nulla ut commodo sagittis, sapien
 dui mattis dui, non pulvinar lorem felis nec erat. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc
 maximus, nulla ut commodo sagittis, sapien dui mattis dui, non pulvinar lorem felis nec erat. Aliquam
 egestas, velit at condimentum placerat, sem sapien laoreet mauris, dictum porttitor lacus est nec enim.
 Vivamus feugiat elit lorem, eu porttitor ante ultrices id.

Press **F11** to exit full screen

ATMIYA

COURSES

- i click1
- i click1
- i click1
- i click1

COURSES

- i click1
- i click1

COURSES

- i click1

Dipen Parmar

This is What we do

This is What we do

This is What we do

TOP TEN OF THE 2019 & 2020

Lorem ipsum dolor sit amet consectetur adipiscing elit. Eos dolorem rem maiores, impedit perferendis voluptatem esse dignissimos reiciendis ullam
 voluptatibus velit, officiis quam! Laborer tenetur, aporiam facilis quisquam qui maiores.
 ProductContainer

Button

Press F11 to exit full screen

MIAN TITLE |

THIS IS SECOND TITLE OF PAGE |

[LOGIN HERE](#)

OUR ONE OF THE BEST PRICES

Front dev

Front dev

Front dev

Front dev

CONTANT FORM

Hello Word -

Email -

Hello Word -

Hello Word -

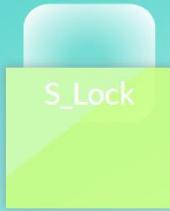
SUBMIT



Community Forum
Lorem ipsum dolor, sit amet consectetur adipisicing elit. Nulla, facilis.



Community Forum
Lorem ipsum dolor, sit amet consectetur adipisicing elit. Nulla, facilis.



SSL Certificates

PUG

- Pug.js to make your life easier with HTML templates

Pug.js is a HTML templating engine, which means you can write much simpler Pug code, which Pug compiler will compile into HTML code, that browser can understand. But why not write HTML code in the first place?

Pug has powerful features like conditions, loops, includes, mixins using which we can render HTML code based on user input or reference data. Pug also support JavaScript natively, hence using JavaScript expressions, we can format HTML code.

Pug.js, as from its name is a JavaScript library. Hence, we can run it inside browser or on node.js. Here, I am using node.js to demonstrate few examples.

This Gulp plugin enables you to compile your Pug templates into HTML or JS, with support for template locals, custom Pug filters, AMD wrapping, and others. Here is a simple example using gulp-pug:

- Pug is available via npm:

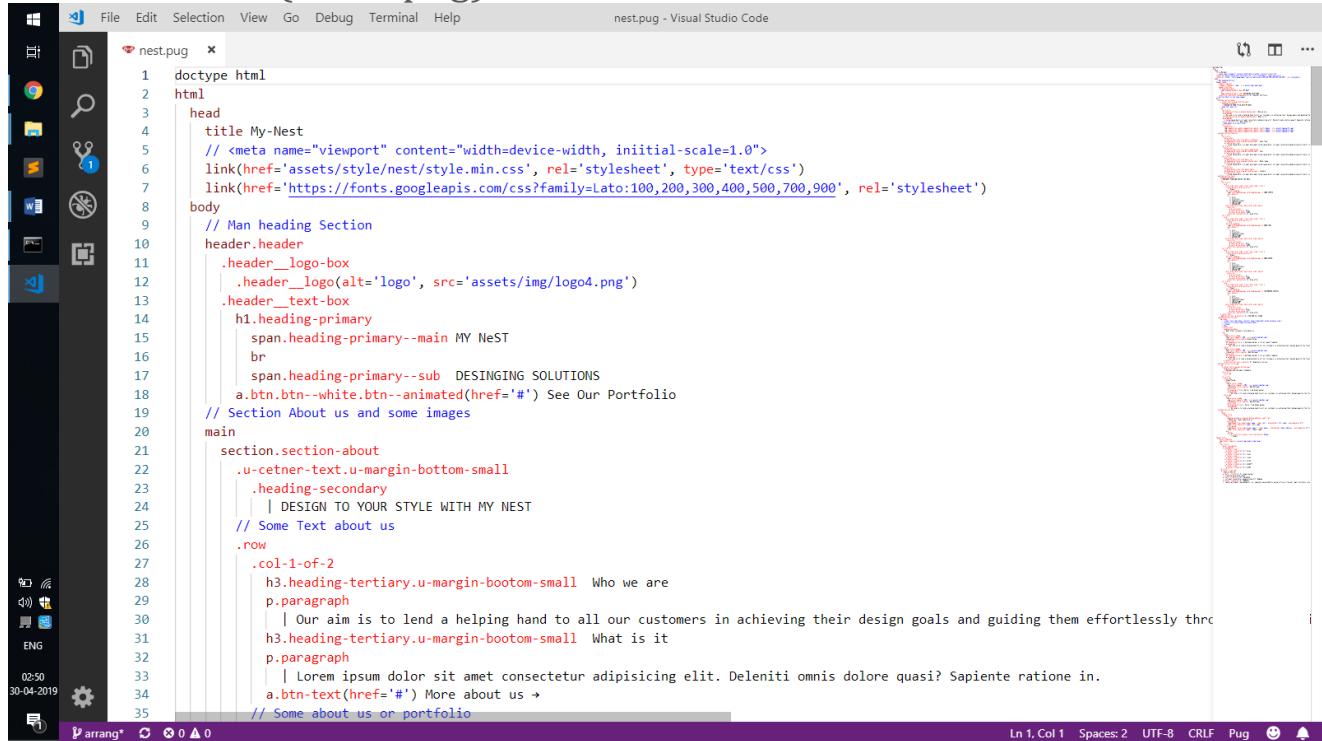
```
$ npm install pug
```

```
var pug = require('gulp-pug');

gulp.task('views', function buildHTML() {
  return gulp.src('views/*.pug')
    .pipe(pug({
      // Your options in here.
    }))
});
```

- Let's see how pug code looks like.

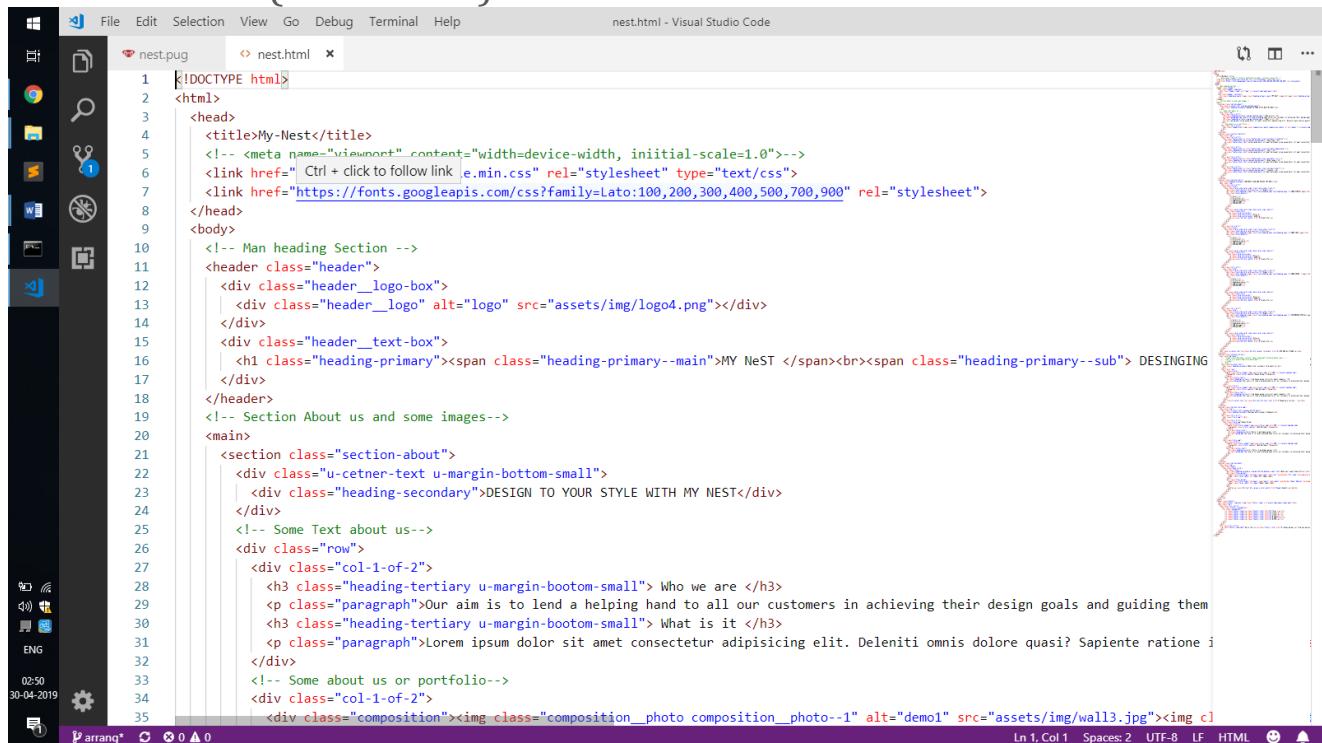
PUG Structure (index.pug)



The screenshot shows the Visual Studio Code interface with the file 'nest.pug' open. The code is a PUG template for a website called 'My-Nest'. It includes a header section with a logo and text, followed by a main section about the company's design philosophy and services. The code uses indentation for structural elements and classes for styling.

```
1 doctype html
2 html
3   head
4     title My-Nest
5     // <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     link(href='assets/style/nest/style.min.css', rel='stylesheet', type='text/css')
7     link(href='https://fonts.googleapis.com/css?family=Lato:100,200,300,400,500,700,900', rel='stylesheet')
8   body
9     // Main heading Section
10    header.header
11      .header__logo-box
12        | .header__logo(alt='logo', src='assets/img/logo4.png')
13        .header__text-box
14          | h1.heading-primary
15            | span.heading-primary--main MY NeST
16            | br
17            | span.heading-primary--sub DESINGING SOLUTIONS
18            | a.btn.btn--white.btn--animated(href='#') See Our Portfolio
19     // Section About us and some images
20     main
21       section.section-about
22         .u-cetner-text.u-margin-bottom-small
23           .heading-secondary
24             | DESIGN TO YOUR STYLE WITH MY NEST
25           // Some Text about us
26           .row
27             .col-1-of-2
28               h3.heading-tertiary.u-margin-bootom-small Who we are
29               p.paragraph
30                 | Our aim is to lend a helping hand to all our customers in achieving their design goals and guiding them effortlessly thro
31               h3.heading-tertiary.u-margin-bootom-small What is it
32               p.paragraph
33                 | Lorem ipsum dolor sit amet consectetur adipisicing elit. Deleniti omnis dolore quasi? Sapiente ratione in.
34               a.btn-text(href='#') More about us →
35     // Some about us or portfolio
```

OutPut HTML (index.html)



The screenshot shows the Visual Studio Code interface with the file 'nest.html' open. This is the output of the PUG template shown above, rendered as standard HTML. The structure remains the same, but the dynamic content like URLs and class names is now static HTML. The code is identical to the PUG version above.

```
1 !DOCTYPE html
2 <html>
3   <head>
4     <title>My-Nest</title>
5     <!-- <meta name="viewport" content="width=device-width, initial-scale=1.0">-->
6     <link href="Ctrl + click to follow link .min.css" rel="stylesheet" type="text/css">
7     <link href="https://fonts.googleapis.com/css?family=Lato:100,200,300,400,500,700,900" rel="stylesheet">
8   </head>
9   <body>
10    <!-- Main heading Section -->
11    <header class="header">
12      <div class="header__logo-box">
13        | <div class="header__logo" alt="logo" src="assets/img/logo4.png"></div>
14      </div>
15      <div class="header__text-box">
16        | <h1 class="heading-primary"><span class="heading-primary--main">MY NeST </span><br><span class="heading-primary--sub"> DESINGING
17      </div>
18    </header>
19    <!-- Section About us and some images-->
20    <main>
21      <section class="section-about">
22        <div class="u-cetner-text u-margin-bottom-small">
23          | <div class="heading-secondary">DESIGN TO YOUR STYLE WITH MY NEST</div>
24        </div>
25        <!-- Some Text about us-->
26        <div class="row">
27          <div class="col-1-of-2">
28            <h3 class="heading-tertiary u-margin-bootom-small"> Who we are </h3>
29            <p class="paragraph">Our aim is to lend a helping hand to all our customers in achieving their design goals and guiding them
30            <h3 class="heading-tertiary u-margin-bootom-small"> What is it </h3>
31            <p class="paragraph">Lorem ipsum dolor sit amet consectetur adipisicing elit. Deleniti omnis dolore quasi? Sapiente ratione in.
32          </div>
33          <!-- Some about us or portfolio-->
34          <div class="col-1-of-2">
35            <div class="composition__photo composition__photo--1" alt="demo1" src="assets/img/wall3.jpg"><img alt="wall3.jpg" data-bbox="106 538 916 890"/>
```