**Corporate Website Documentation**

# Technology used

* **Base Technology** – HTML5, CSS3, Javascript (ES5)
* **CSS Preprocessor** – SASS (.scss files)
* **Javascript framework**

1. jQuery (version 3.2.1)
2. VueJS (non cli version ; version 2.x)

* **Javascript libraries**

1. Jquery.dlmenu.js (for responsive menu; for more info on this, find jquery.dlmenu.js under “[Folder Structure and Files > Assets Folder > JS Folder > Folders > Vendor > Common](#_Common)” section)
2. Owl-carousel.min.js (for carousel; for more info on this, find owl-carousel.js under “[Folder Structure and Files > Assets Folder > JS Folder > Folders > Vendor](#_Vendor)” section)
3. GSAP (mainly for animation; for more info on this, go to “[Folder Structure and Files > Assets Folder > JS Folder > Folders > Vendor > GSAP](#_GSAP)” section)

* **Development and build technologies**

1. Npm scripts
2. Node JS
3. Selenium webdriver

For more info on this, go to “[Development & Build](#_Development_&_Build)” section.

# Browser Compatibility

This website is compatible in all browsers and IE 11 and above.

Actually entire website is supported in IE10 but because of onetrust cookie consent script, which disables jquery code in IE 10.

In non-supported browsers, website show browser compatibility message.

The code for browser compatibility is written in assets/js/compatibility-detector.js

# Folder Structure and Files

Each and everything resides in “en” folder, which has following folders.

Each folder has web.config file which can be upload to live server (to upload files on demo server, go to “[Upload](#_Upload)” section).

## Assets Folder

All the assets like css, fonts, images, js and json files are resides in this folder. Following are the files and folders

### Files

* **All-countries-data.json** – this file has data about countries displayed in country dropdown. (for more info about this data, go to “[Content/Data Changes > Country Dropdown Data](#_Country_Dropdown_Data)” section)

### CSS Folder

* **dlmenu.css** – for responsive menu and child menu e.g. Blog Page sub section, News sub section (use dlmenu.min.css for linking, link is given in style.scss)
* **map.css** – for map used in footprint page (for more info about footprint map, go to “Footprint Map” section)
* **owl-carousel.min.css** – for carousel (used in tabs in individuals home page, each blog article page, each government solution page and each executive board member page)
* **style.scss** – all website styles are created in this file (for more info about styles go to “Style” section)

### Fonts Folder

Font file format used are .eot, .svg, .ttf, .woff. Each folder has font-face.css which contains @font-face declaration (use font-face.min.css for linking, link is given in style.scss).

* **Icons Folder** – All the icons given in website uses this font. (for more info about icons go to “Icons” section)
* **Inter-UI Folder** – This font is used for textual content in entire website. It has 2 font “InterUI-Regular” and “InterUI-Bold” for regular and bold font respectively.

### Images Folder

It has all the image files even .svg image files are here. Following are the files and folders

#### Files

* **all-browser-logos.png** – this image is used in showing browser compatibility message (for more info about browser compatibility, go to “[Browser Compatibility](#_Browser_Compatibility)” section).
* **all-flags.png** – this image is used to show the flag of countries (used in dropdown; for more info about flags, go to “[Flags](#_Flags)” section).
* **bg-map.png** – this image is used for showing map in background (used in governments home page and in milestones page)
* **vfs-global-logo.svg** – this image is used to show logo of VFS Global (given in Header of all pages)
* **vfs-global.logo.png** – this image is used to show logo in browser compatibility message, since IE8 doesn’t support SVG.
* **vfs-200-million-logo.svg** – this image is used to show 200 million applications celebration logo

#### Folders

* **About** – All the images of “About” section from both Individuals and Governments section are here.
  + **Executive-Board** – This folder is inside About folder, since Executive Board is in About section of Governments. All the image of executive board members are here. All the files stored here must be of resolution 450 \* 450 and of format .jpg.
* **Banners** – All the blue colored banner images are here.
* **Blog** – All the images used in “Blog” section under Individuals section are here.
  + **Sub Section Folder** – Sub sections of blog are “Visa Application Guide”, “About”, “Featured”, “People & Places” and their respective folders are “visa-application-guide”, “about”, “featured”, “people-places”.

Images used in content of each article page of each section are stored here and main article image is stored resolution wise.

* + - **Resolution Folder** – Each and every sub section of blog has 3 resolution folders i.e. “extra-large”, “large”, “medium”. These are folder where main image of article is stored.

1. **Extra-large** – All the images stored here are used for top banner in article page. Image resolution must be 1500 \* 640
2. **Large** – All the images stored here are used for showing in article in Individuals home page.
3. **Medium** – All the images stored here are used in article box (e.g. see any sub section page like blog-visa-application-guide.html; to see html page in browser, go to “[Run](#_Run)” section).

* **Certification** – certification images used in footer are stored here.
* **Footprint** – Images required for footprint page are stored here.
* **Government-services** – Images shown in “Government Services” tab under Individuals home page are stored here. All images must be of maximum width 58 and maximum height 38.
* **Icons** – favicon (shortcut icon) and icons used in plain textual content are stored here.
* **News** – All the images used in “News” section under Governments section are here.
  + **CSR** – All the images used in “Social Responsibility” under “News” section are stored here.
  + **Events** - All the images used in “Events” under “News” section are stored here.

Image resolution in both “CSR” and “Events” must be of 440 width and height must be no more than 330.

* **Solutions** – All the images of “Governments > Solutions” are stored here. All the files stored here must be of resolution 1500 \* 640 and of format .jpg.

(Note: Service Icon of each Governments Solutions are not an images those are font icons, to know more about icons go to “Icons” section).

* **Tourism-services** - Images shown in “Tourism Services” tab under Individuals home page are stored here. All images must be of maximum width 58 and maximum height 38.

### Js Folder

All the java script files are here. Following are the files and folders. Logic of each and every code is written in comment in respective js files. (to do any changes in any js file and view in browser, go to “[Run, Development, Build & Upload](#_Run,_Development,_Build)” section).

To know more about structure about website from javascript perspective, go to “Javascript Structure”

#### Files

* **Compatibility-detector.js** – code for browser compatibility is written here. It is linked in each and every html file. (linked compatibility-detector.min.js)
* **Config.js** – configuration of website and even some data that are to be shown in multiple pages to be taken from here (for more info read comments in config.js file)
* **Main.js** – initializes vfs modules (written in vfs-global.js) and vue application (written in vue-app.js).

Loading of One trust cookie consent js file is wriiten here. (for more info read comments in main.js file)

* **Vfs-global.js** – all the modules required throughout the website are written here. Modules are exported in window object that means you can see list of modules and sub modules in console by typing vfs. (for more info read comments in vfs-global.js file)
* **Vue-app.js** – all the code related to Vue is written here. (for more info read comments in vue-app.js file)
* **Map.js** – js code for footprint map is written here. (linked map.min.js)
* **Script.min.js** – this file is linked in all pages which has code of all 4 files like vfs-global.js, vue-app.js, config.js and main.js

#### Folders

##### **Vendor**

All the 3rd party plugins and libraries are here. Following are the files and folders.

* + **Files**
  + **owl.carousel.js** – for carousel (used in tabs in individuals home page, each blog article page, each government solution page and each executive board member page).

(linked owl-carousel.min.js)

Important Note: Don’t replace it with original Owl Carousel library from plugin’s website since we have changed code of it to solve error for autoWidth enabled carousel used in tab buttons.

* + **gsap.min.js** – it is auto generated from “GSAP” folder (for more info about how it is generated automatically, go to “Run, Development, Build & Upload” section)
  + **common.min.js** – it is auto generated from “common” folder (for more info about how it is generated automatically, go to “Run, Development, Build & Upload” section)
  + **Folders**

###### **Common**

All the libraries here are linked in all html files, and common.min.js is getting generated automatically (for more info about how it is generated automatically, go to “Run, Development, Build & Upload” section). Following are the files and folders.

* + - **Files**
  + **jquery-3.2.1.min.js** – jQuery plugin
  + **vue.min.js** – VueJS plugin, this is automatically replaced during development. (for more info about why and how it is getting replaced, go to “Run, Development, Build & Upload” section)
  + **modernizr.custom.js** – this plugin is used for responsive menu
    - **Folders**
* **Jquery-plugins** – all the jquery dependent plugins are stored here. Following are the files and folders.
  + - * **Files**
  + **Jquery.dlmenu.js** – this plugin is used for responsive menu and child menu e.g. Blog Page sub section, News sub section
* **Vue-resource** – VueJS file both minified and unminified are here.

When website is in development mode unminified file is used and in live website minified version is used. (for more info about why and how it is getting replaced, go to “Run, Development, Build & Upload” section)

###### **GSAP**

GSAP is a js tool for high performance animation (for more info about GSAP visit <https://greensock.com/gsap>). Following are the files and folders.

* + - **Files**
  + **TweenMax.min.js** – main library for animation (used in 200 million logo animation and milestones, for more info go to vue-app.js million-200-logo and milestones component)
    - **Folders**
  + **Utilities** – all the GSAP utility tools are here. Following are the files and folders.
    - * **Files**
    - **Draggable.min.js** – this library is used to drag element horizontally, vertically or rotationally. It is used in milestones page.

## General Folder

This folder has all html files which are common to both “Individuals” and “Governments” sections.

This folder has sitemap.xml file which has urls of all html files resides in this folder except footprint-map.html file since it’s an iframe which is called inside footprint.html page. Sitemap.xml is being automatically generated (for more info about how it is generated automatically, go to “Run, Development, Build & Upload” section)

Also it has robots.txt file which has link to sitemap.xml file.

## Individuals Folder

This folder has all html files which are only for “Individuals” section.

This folder has sitemap.xml file which has urls of all html files resides in this folder. Sitemap.xml is being automatically generated (for more info about how it is generated automatically, go to “Run, Development, Build & Upload” section)

Also it has robots.txt file which has link to sitemap.xml file.

## Governments Folder

This folder has all html files which are only for “Individuals” section.

This folder has sitemap.xml file which has urls of all html files resides in this folder. Sitemap.xml is being automatically generated (for more info about how it is generated automatically, go to “Run, Development, Build & Upload” section)

Also it has robots.txt file which has link to sitemap.xml file.

## PDF Folder

This has all the pdf files used in website. Following are the files and folders

### CSR Folder

This has all the pdf files that will be used in CSR (Corporate Social Responsibility) page.

### Media-releases Folder

This has all the pdf files that will be used in Media releases page. This has year wise folder which has pdf files, which will be linked to content of that year in media releases section.

# Content/Data Changes

Note: Simple way to update content/data would be find related text in entire folder; then open found file and update.

Data other than listed here are given in config.js and vue-app.js. If you want to know about that data which is not listed here, find keyword of that data in config.js and vue-app.js and read comments.

Below are given the list of content/data section with files and location.

## News Sub Section Content

News sub section content is given in config.js (just find all\_content\_details in config.js and then find news subsection name)

1. media\_releases – it has link to pdf files which is stored in PDF/media-releases folder. To know more about folder and file structure, go to [Media-releases Folder](#_Media-releases_Folder).
2. news\_coverage – it has external links
3. events – it has images which is stored in assets/images/news/events. To know more about images, find **Events** in [Images Folder > Folders](#_Folders) section.
4. Awards – it is plain content without any image.
5. Csr – it has images which is stored in assets/images/news/csr. To know more about images, find **CSR** in [Images Folder > Folders](#_Folders) section.

To know detail about each sub section content, open config.js and read comments.

To know about how to run and update changes in web browser, go to “[Run, Development, Build & Upload](#_Run,_Development,_Build)” section.

## Navigation Menu Content

Navigation menu content is given in config.js (just find navigation\_page\_details in config.js and then find the respective text to update)

To know detail about navigation content, open config.js and read comments.  
To know about how to run and update changes in web browser, go to “[Run, Development, Build & Upload](#_Run,_Development,_Build)” section.

Below are some content which are given in navigation\_page\_details array.

### Executive board members

Html file name of each board member must be given as board-member-(fullname).html e.g. board-member-zubin-jal-karkaria.html

List of executive board members are given here. Just find executive\_board in navigation\_page\_details.  
For more info about board member Box content, open config.js and read comments.   
Detailed html content of each and every board members are given in respective html page itself.

Board member images are stored in assets/images/about/executive-board. To know about images of board members, find **Executive-Board** folder in [Images Folder > Folders](#_Folders) section.

### Governments solutions

Html file name of each government solution must be given as solutions-(solution-name).html e.g. solutions-lidpro.html

List of governments solutions are given here. Just find solutions\_services in navigation\_page\_details.  
For more info about government solution Box content, open config.js and read comments.  
Detailed html content of each and every Government solutions are given in respective html page itself.

Banner Images of Government solutions are stored in assets/images/solutions. To know about images of government solutions, find **Solutions** folder in [Images Folder > Folders](#_Folders) section.

To know more about solution icons go to “Icons” section.

### Blogs

Html file name of each blog sub section must be given as blog-(sub-section-name).html e.g. blog-featured.html, blog-visa-application-guide.html, etc.

Html file name of each article (post) must be given as article-(article-name).html e.g. article-simple-visa-tips-for-a-great-start.html, article-setting-up-your-finances-abroad.html, etc.

List of articles (posts) are given here. Just find blog in navigation\_page\_details.  
For more info about article Box content, open config.js and read comments.  
Detailed html content of each and every articles are given in respective html page itself.

Images of Article are stored in assets/images/blog. To know about images of blog article, find **Blog** folder in [Images Folder > Folders](#_Folders) section.

## Certification and Accreditation Content

This content is given in vue-app.js (just find certification\_content in vue-app.js and then find the respective text to update)

To know detail about this content, open vue-app.js and read comments.  
To know about how to run and update changes in web browser, go to “[Run, Development, Build & Upload](#_Run,_Development,_Build)” section.

## Individuals Home Page Tabs Heading and Related 3 Steps Content

Individuals Home Page Tabs heading is given in vue-app.js (just find service-tabs-3-steps-container-with-content in vue-app.js and then find the respective heading under template to update)

3 Steps Content is given in vue-app.js (just find service-tabs-3-steps-container-with-content details in vue-app.js and then find the all\_step\_content under data)

To know detail about these contents, open vue-app.js and read comments.  
To know about how to run and update changes in web browser, go to “[Run, Development, Build & Upload](#_Run,_Development,_Build)” section.

## Government Services in Individuals Home Page

This content is given in vue-app.js (just find government-services component definition in vue-app.js and then find the respective content under data function)

Government services images are stored in assets/images/government-services/. To know about images of Governments services, find **Government-services** folder in [Images Folder > Folders](#_Folders) section.

To know detail about government services content, open vue-app.js and read comments.  
To know about how to run and update changes in web browser, go to “[Run, Development, Build & Upload](#_Run,_Development,_Build)” section.

## Tourism Services in Individuals Home Page

This content is given in vue-app.js (just find tourism-services component definition in vue-app.js and then find the respective content under data function)

Tourism services images are stored in assets/images/tourism-services/. To know about images of Tourism services, find **Tourism-services** folder in [Images Folder > Folders](#_Folders) section.

To know detail about tourism services content, open vue-app.js and read comments.  
To know about how to run and update changes in web browser, go to “[Run, Development, Build & Upload](#_Run,_Development,_Build)” section.

## VDash Widget Top line in Individuals Home Page

This content is given in vue-app.js (just find vdash-widget component definition in vue-app.js and then find the respective content under data function)

To know detail about this content, open vue-app.js and read comments.  
To know about how to run and update changes in web browser, go to “[Run, Development, Build & Upload](#_Run,_Development,_Build)” section.

## Company Profile Content

This content is displayed in below two places.

1. Individuals > About
2. Governments > About > About VFS Global > Profile

This content is given in vue-app.js (just find company-profile-content in vue-app.js and then find the content under template)

In this content we have linked stats data. (For more info about stats data, go to “[Content/Data Changes > Stats Data](#_Stats_Data)” section)

To know detail about company profile content, open vue-app.js and read comments.  
To know about how to run and update changes in web browser, go to “[Run, Development, Build & Upload](#_Run,_Development,_Build)” section.

## Contact Media Queries Content

This content is displayed in below two places.

1. Individuals > Contact
2. Governments > Get in Touch

This content is given in vue-app.js (just find contact-media-queries in vue-app.js and then find the content under data function)

Note: We have obfuscated email link. (For more info about email obfuscation, go to “[Content/Data Changes > Email Obfuscation](#_Email_Obfuscation)” section)

To know detail about company profile content, open vue-app.js and read comments.  
To know about how to run and update changes in web browser, go to “[Run, Development, Build & Upload](#_Run,_Development,_Build)” section.

## Stats Data

Below are the list of stats which are used in website.

These are the list of stats that are given manually in stats object in vue-app.js file (just find stats in vue-app.js file)

1. **Number of Application Centers** - Key: application\_centres
2. **Number of Countries of Operation** – Key: operation\_countries
3. **Number of Continents** – Key: continents
4. **Number of Client Governments** – Key: client\_governments
5. **Number of Employees** – Key: employees
6. **Number of Biometric Enrolments (in millions)** – Key: biometric\_enrolments\_in\_millions
7. **Number of Applications** – Key: applications
8. **Date of Stats** – Key: stats\_date

These are the list of computed stats that are computed programmatically using above data. Which are also given in vue-app.js file (just find applications\_str in vue-app.js file)

1. **Number of Applications in comma separated format** – Key: applications\_str (computed from stats.applications)
2. **Number of Applications (in millions)** – Key: applications\_in\_millions (computed from stats.applications)
3. **Date of Stats by adding ordinal indicator** – Key: stats\_ordinal\_indicator\_date (computed from stats.stats\_date) e.g. if stats\_date = 31 July 2019 then computed ordinal indicator date will be 31st July 2019.

We also show stats data in Individuals > About in circles.  
To change text or icon there, just go to vue-app.js file and find all\_stats\_details array. (For more info on this, go to vue-app.js and read comments)

Pl. note these data can be linked in html page inside master-page tag and also in Vue component template using vue interpolation or v-dynamic-html (For more info about v-dynamic-html, go to “Javascript Structure”).

E.g. Total number of applications are over {{applications\_in\_millions}} and {{stats.client\_governments}} client governments.

These stats data are shown in many places like Government Home Page, Individuals/Government > About Us Page, Milestones page etc.

To know about how to run and update changes in web browser, go to “[Run, Development, Build & Upload](#_Run,_Development,_Build)” section.

## Country Dropdown Data

Country dropdown data is used in visa/permits and passport services in individuals home page.  
It is given in all-countries-data.json file.

Below are data that are given in all-countries-data.json

### Visa/Permits and Passport Services

Key of these data are “visa\_and\_permits” and “passport\_services”.

This data is given in below format.

Note: Resident country name in an array must be unique. And also visiting country in that resident country must be unique.

[{

“name”: “resident country name”,

“visiting”: [{

“name”: “visiting country name”,

“url”: “website url”,

“contact\_url”: “url of contact page” // this will be shown in individuals contact page

}, {

…..

},

…..]

},

{

…..

},

…..

]

### Geographic Details

Key of this data is “geographic\_details”.

This contains data about country flag and country code of each and every resident and visiting countries. See below for description of each

1. **Flag** – it stores the position of flag of current country in all-flags.png sprite. Format of position is [x, y] (for more info about flags, go to “[Flags](#_Flags)” section).
2. **Country Code** – it stores the iso-2-letter code of current country. It will be used in preselecting the user’s country based on the ip.

Note: Don’t add same geographic details for different country name. If you want to, then add entry in geographic\_duplicates.

### Geographic Duplicates

Key of this data is “geographic\_duplicates”.

This contains the entry of those country names which have duplicate flag and country code.

e.g. “Finland Residence Permit” has same geographic details as “Finland”;   
“Saudi Arabia & GCC Residents” has same geographic details as “Saudi Arabia, Kingdom of” (Kingdom of is given after country name because in footprint map we want to show “Saudi Arabia, Kingdom of” in “S” letter rather than “K” letter; for more info about footprint map, go to “Footprint Map” section).

### Country Aliases

Key of this data is “country\_aliases”.

This contains the data about alias names of countries.   
E.g. “United Arab Emirates” – alias name is “UAE”  
“Côte d’Ivoire (Ivory Coast)” – alias name is “Cote d'Ivoire” (character “ô” and “’” is changed to “o” and “'” respectively)

This data is used while user searches the dropdown.  
e.g. If user has typed “UAE” he must see “United Arab Emirates” option in dropdown.

## Email Obfuscation

We have obfuscated email id’s in entire website wherever used. Email obfuscation is required. Search google for why it is required. We have tool to obfuscate email ID. Visit below link

<http://210.210.25.188/vfsdemowebsites/New_Template/Encoder-script/>

# Run, Development, Build & Upload

## Run

To run this website in browser, we need localhost server.

Install “200 ok” chrome app which is a tool that will create localhost server. After installing run the app and choose “en” folder.

## Development & Build

This website uses “npm scripts” and custom node script for development and build purpose.

Important Note: First you need to install below npm packages to make “npm scripts” workable.

Make sure you run below command as it is without editing. Since some packages we require to install globally

1. npm install --global cpy-cli
2. npm install --global uglify-js
3. npm install --global rimraf
4. npm install --global npm-run-all
5. npm install --global ncat
6. npm install --global chokidar
7. npm install --save-dev cli-progress
8. npm install --save-dev pretty
9. npm install --save-dev selenium-webdriver

Also, after installing all the above packages we need to download selenium driver for chrome <http://chromedriver.chromium.org/downloads>.

To allow selenium webdriver workable from nodejs, you need to add folder path of driver in PATH variable (For windows, Right Click Computer > Properties > Advanced System Settings > Advanced Tab > Environment Variables > Under System Variables section > In Variable column find Path).

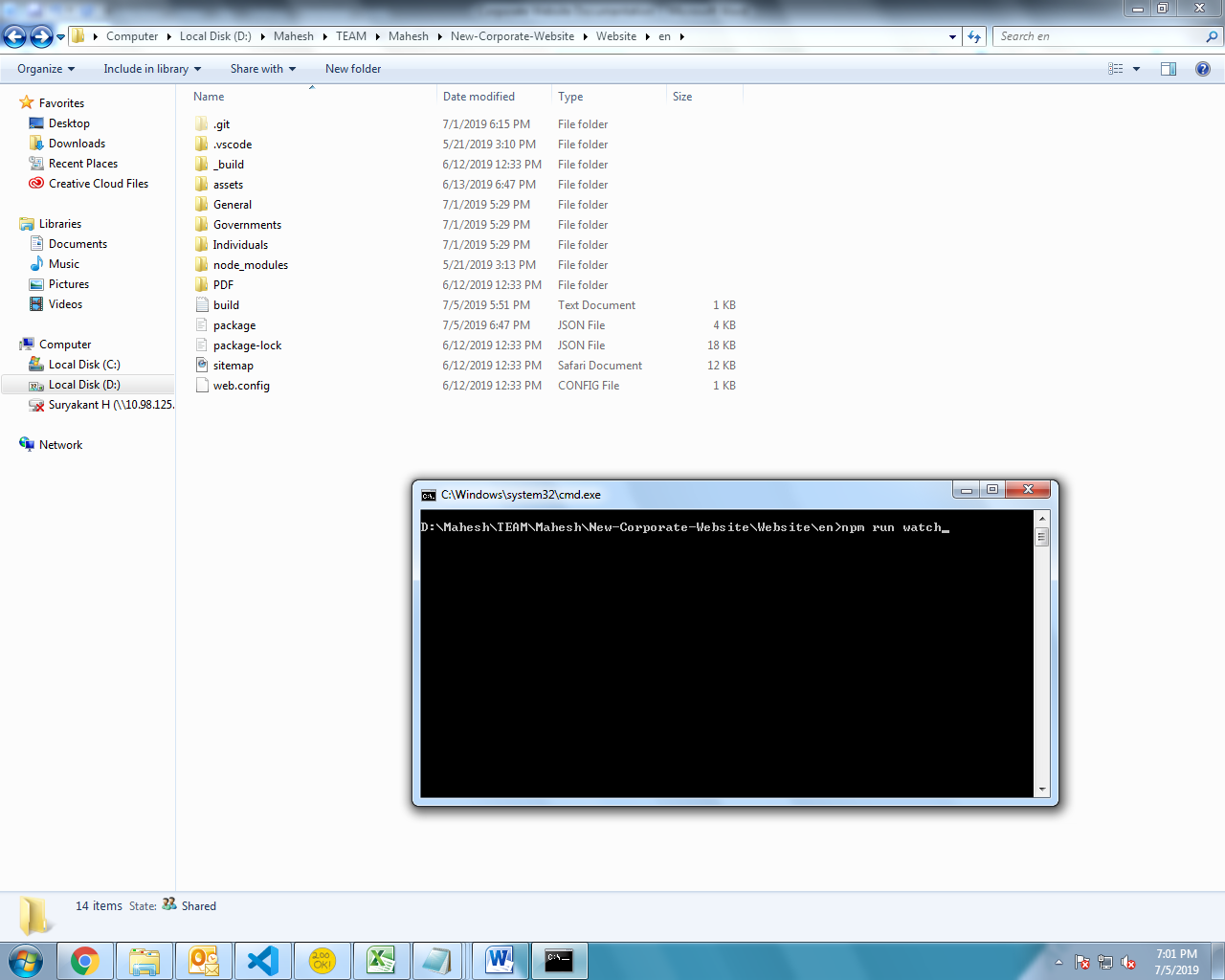
We need Selenium Webdriver for build purpose. For more info on why we need selenium webdriver, go to “Build” section.

### Development

#### JS / JSON

If you are changing in one of the below files, then you need to run “npm run watch” in command window in “en” folder

1. compatibility-detector.js
2. vfs-global.js
3. config.js
4. all-countries-data.json
5. vue-app.js
6. main.js
7. map.js



It will start website for development purpose. When you edit in any of the above listed files, after avgerage 4-5 seconds you can refresh the browser to see changes. (For more info on how to run website in browser, go to “[Run](#_Run)” section)

**What does “npm run watch” do?**

When you run “watch” command, it copies above unminified js files to minified js file (which is linked to all the html files) for faster development purpose.

After copying all unminified files to their appropriate location, it starts running chokidar, which watches for the file changes and on change of any file it again copies unminified files.

For detailed info, go to package.json and under that understand the commands given in “scripts” object.

#### Style.scss

In all html files style.min.css is linked so we need conversion tool, if you are using visual studio code editor then you can install “Easy Sass” extension. This extension convert .scss files to .css and .min.css in same folder path on save of .scss file. (For more info about this extension, visit <https://marketplace.visualstudio.com/items?itemName=spook.easysass>)

(for more info about styles go to “Style” section)

#### Other files

If you are changing in html files then you don’t need to run any command or don’t need any tools.

For below files, you need to do minification manually.

##### **font-face.css**

Both the font-face file of “InterUI” and “Icons” font are linked in style.scss (but font-face.min.css is linked instead of font-face.css)

So an easy way is,  
During development purpose, in style.scss change link from font-face.min.css to font-face.css.  
After Development completes, minify font-face.css online and change link in style.scss back to font-face.min.css

##### **owl.carousel.js**

This file is not linked in any html pages instead owl.carousel.min.js file is linked.

So during development, you need to change owl carousel link from .min.js to .js in one of the html file.

After development completes minify js online and change html file owl carousel link back to the .min.js

##### **jquery.dlmenu.js**

This file is not linked directly, instead all html file has link to vendor/common.min.js, which has minified code of third party plugins that are required in all html pages.

So after changes in jquery.dlmenu.js file run “npm run build” command. (for more info on build, go to “Build” section)

##### **dlmenu.css**

It is linked in style.scss (but dlmenu.min.css is linked instead of dlmenu.css)

So an easy way is,  
During development purpose, in style.scss change link from dlmenu.min.css to dlmenu.css.  
After Development completes, minify dlmenu.css online and change link in style.scss back to dlmenu.min.css

### Build

If you do changes in any js file or if you have added new html pages, and want to upload, then you need to run build command in command window.

There are 2 build commands “build” and “build-all” depending on what changes you did.

#### npm run build-all

This command is a complete build command, no matter whatever changes you have did in website this command will make website ready to upload. (for more info on how to upload, go to “Upload” section)

**What does “npm run build-all” do?**

Coming soon…

#### npm run build

This command does the short job of what “build-all” command does.

Coming soon…

## Upload

Only the following folders are used on server side

1. assets
2. Individuals
3. Governments
4. General
5. PDF

Don’t upload any other folders on server side.

Important Note: If you upload any of the above entire folder on demo/testing server, then delete **web.config** files from all the 5 folders from server, since web.config files are written for live server only.

**What to upload and when?**

1. If you have done edits in any javascript file then upload entire JS folder.
2. If you have run “npm run build-all” command then upload all the three folder (Individuals, Governments, General), since it modifies **sitemap.xml** file. (For more info on when to run build-all command, go to “[Build > npm run build-all](#_npm_run_build-all)” section)
3. If you done edits in style.scss then upload additional style.css and style.min.css files.

# Flags

Flags of all countries that are used in country dropdown is the sprite image. The path of that sprite image is assets/images/all-flags.png.

**How to add new flag?**

First create flag of that country of size 28 \* 20 and add that flag image to sprite image after last flag of last row (if there is no space after that last flag then increase space below i.e. canvas size; and then add flag image).

Since it is sprite image, we need to store position of newly added flag. Position of each flag is stored inside geographic\_details in assets/all-countries-data.json file.

The format of geographic\_details is given below.

{

“Country Name”: {

“flag”: [x, y],

“country\_code”: “…”

},

…

}

In “flag”: [x, y]

x is x-coordinate, y is y-coordinate and both is number which starts from 0

e.g. First flag is of Algeria so position is [0, 0]

Second flag in first row is of Australia so position is [1, 0]

India flag position is [9, 1]

To know more about geographic\_details or country\_code, go to “[Country Dropdown Data > Geographic Details](#_Geographic_Details)” section.

# Icons

This website uses fonts for monocolor icons like font-awesome do.

The font files and their font face declaration file is stored in assets/fonts/icons folder.

The icons are displayed by giving class name (class name can be given to any tag but generally ‘i’ tag).  
We also have ‘icon’ vue component that are used in some other components like icon-heading-wrapper and icon-details (For more info on vue components go to “Javascript Structure” section or open vue-app.js file and read comments)

The List of icon class names are listed in font-face.css file in icons folder under fonts folder.

There is convention of governments solution icon i.e. use icon-solution-(solution-name).  
E.g. icon-solution-fast-pass, icon-solution-internet-kiosk, etc.

**How to add/change icon?**

First create svg file of an icon.