OnBoarding Plan Documentation

Hello new star!

We'd like to welcome you to the SITech fabulous team on behalf of all the staff. Each of us will play a role to ensure your successful integration into the department.

Your new team anticipates taking you out to launch to get to know you and to make sure that you meet everyone whom you'll be working with .

And for the required skills, this document will get you the required links to learn. Let's get you started!

What is **SITech?**

At SiTech, we are obsessed with helping companies transform into globally competitive companies by taking the pain of building technologically great products that help them excel with their mission. Transforming companies into new-age tech companies, is at its heart,

To whom who always has thirst for success in his journey

putting the enterprise in a startup mentality that can help them be nimble, fast, iterative, and globally competitive. Check out our Case Studies. Be sure to go through our various Perspectives on various technology related issues.

As a full stack developer, you'll work with both the front end and the back end of a website.

Front End:

The front end of a website is the part a user sees and directly interacts with. As a full stack developer in SITech, you'll be working with HTML, CSS, JavaScript, JQuery, and Bootstrap.

Back End:

The back end is much more behind-the-scenes for specific projects. As a full stack developer in SITech, you'll be working mainly with Python language and Django web application framework.

Why not start getting you all set? Let's go!

Guide: we'll start with the front end programming languages then the back end then a little project to test your awesome skills.

HTML:

If you are a beginner we recommend you this Code Academy course
https://www.codecademy.com/learn/learn-html

If you're already familiar with HTML then go to
https://www.w3schools.com/html/default.asp

➔ subjects :

- HTML Basic
- HTML Elements
- HTML Attributes
- HTML Headings
- HTML Paragraphs
- HTML Styles
- HTML Formatting
- HTML Quotations
- HTML Comments
- HTML Colors
- HTML CSS
- HTML Links
- HTML Images
- HTML Tables
- HTML Lists
- HTML Blocks
- HTML Classes
- HTML Id

Note: Always keep in my mind to keep the clode clean and to understand how the flow works for future purposes.
"you don't need to memorize a one google search away code"

-Ahmad Bazadough

To whom who always has thirst for success in his journey

CSS:

If you are a beginner we recommend you this Code Academy course
https://www.codecademy.com/learn/learn-css
Note: no need to go deeper in the grid lessons. Just go through the first lesson.

If you're already familiar with CSS then go to
https://www.w3schools.com/css/

➔ subjects :

- Syntax CSS
- How To
- CSS Colors
- CSS Backgrounds
- CSS Borders
- CSS Margins
- CSS Padding
- CSS Height/Width
- CSS Box Model
- CSS Outline
- CSS Text
- CSS Fonts
- CSS Icons
- CSS Links
- CSS Lists
- CSS Tables
- CSS Display
- CSS Max-width
- CSS Position
- CSS Overflow
- CSS Float
- CSS Inline-block

- CSS Align
- CSS Combinators

JavaScript:

If you are a beginner we recommend you this Code Academy course
https://www.codecademy.com/learn/introduction-to-javascript
Note: no need to go deeper in the requests lesson.

If you're already familiar with JavaScript then go to
https://www.w3schools.com/js/

➔ subjects :

- JS Syntax
- JS Comments
- JS Variables
- JS Operators
- JS Arithmetic
- JS Assignment
- JS Functions
- JS Objects
- JS Events
- JS Arrays
- JS Array Methods
- JS Array Sort
- JS Array Iteration
- JS Conditions
- JS Switch
- JS Loop For
- JS Loop While
- JS Break
- JS Type Conversion

- JS Forms

All good for now?
If you have any question regarding anything above please reach us or ask any member, anyone would gladly help you out. -if they ask for money report them to the HR they're used to this. *hopes not*

❖ Project time.

Front End Project:

#partOne

Requirements:

- PyCharm: the Python IDE (preferable)

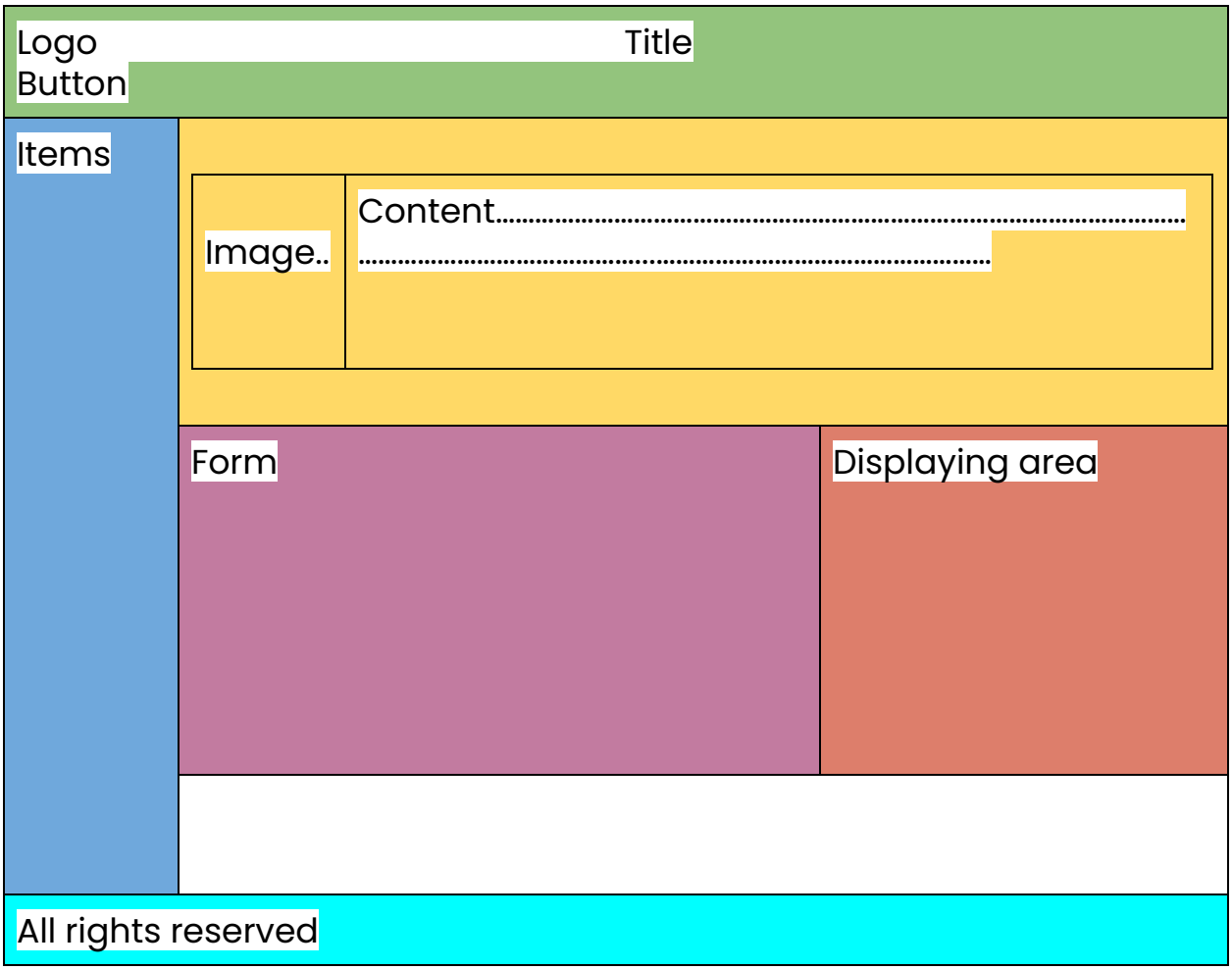
Introduction: In this section, we'll cover how to take the skills you've learned on links above and use them to create a basic web page built entirely from the tools you have on your computer. The project will focus on using HTML and CSS. Exciting right?

Note: The good news is *drum roll please * you can choose the subject content, all we care about is the output structure.

Project output:

- The green box is the header of the page. Consist of three things.
- The blue box is the sidebar of the page. It contains of items(be creative)
- The orange box is the page main content

- The purple box is a space for the form (will be explained later).
- The watermelon color box is a space for displaying data when pressing on submit button in the previous form.
- The light blue is the footer of the page.



The form will consist of 4 fields:

1. Text (enter text)
2. Color (choose a color for the text)
3. Size (text size)

To whom who always has thirst for success in his journey

4. repeat(how many times to repeat the text)

   And a submit button

Note: when clicking on the submit button the data will be displayed on the displaying area

Let's get you started!

STEP 1: CREATE A FOLDER STRUCTURE FOR YOUR WEB PAGE

Let's create a folder structure to support your web page. A well-designed folder structure will help you quickly navigate to the HTML or CSS files that contain your code.

First, open Pycharm. Next, create a folder (also known as a directory) called **ExampleProject**. This folder will contain all of the files for your HTML and CSS project.

Open the **ExampleProject** folder. Inside, create the following items:

1. A new file called **index.html** (use your preferred text editor)
2. A new folder called **resources**

The **index.html** file will contain the HTML code for your web page, while the **resources** folder will contain all of the necessary resources needed by the HTML files (CSS files, images, etc.).

Next, open the newly created **resources** folder. Inside this folder, create the following:

1. An additional folder named **CSS**

The **CSS** folder will contain the CSS files needed to style your web page.

Finally, open the **CSS** folder you just created. Inside this folder, create the following:

1. A new file named **index.css** (use your preferred text editor)

The **index.css** file will contain all of the CSS styling rules that will be applied to your web page.

Note: This overall folder structure will help support your workflow as you add files or resources.

At a high-level, here's what it should look like:

-ExampleProject(directory)

Index.html

-Resources(directory)

-Css (directory)

Index.css

STEP 2: ADD CONTENT TO YOUR WEB PAGE

Great! With your folder structure, HTML, and CSS files all in the right place, you can add content to the web page.

First, open the **index.html** file in your preferred text editor. Next, add the required HTML code.

STEP 3: LINK YOUR HTML FILE AND CSS FILE

As it turns out, the HTML content you added will not be styled by the CSS rules unless **index.html** and **index.css** are linked together. In the `<head>`section, link the stylesheet to the HTML file.

```
<link href="./resources/css/index.css" type="text/css" rel="stylesheet">
```

You might be wondering why the `href` attribute is set to `./resources/css/index.css`. This is because you must specify exactly where the **index.css** file lives within your folder(s) *relative* to where **index.html** lives (otherwise, the two files won't link).

Note: again, make sure to save your changes!

STEP 4: DO YOUR PROJECT

To whom who always has thirst for success in his journey

Start creating the above project, don't hesitate to ask any question. we're here to support our new star. At this point, feel free to make changes to your HTML or CSS code. Keep in mind that in order to view any *new* changes you make, you'll have to refresh your browser.

Okay, now you finished. not literally, huh! we mean the project.

REVIEW TIME!

Congrats! In this short time, you learned how to take what you've been learning on Codecademy/W3schools and apply it using the tools you have available on your own personal computer. You successfully learned how to:

1. Create a folder structure to support both your workflow and your web page
2. Add HTML content and CSS styling to respective files
3. Link the HTML and CSS files together
4. View your web page in a browser (and refresh the browser to view new changes)

In general, the four points above are a strong starting point for your own developer workflow. As you learn more, you may modify the workflow to fit your specific needs. At this point, feel free to

modify the content of the web page or create an entirely new project. Happy coding!

#partTwo

Congrats again for the awesome website! How we know it's awesome, of course, it’s awesome. You’re one of us now.

Project output:

Make the form available and the displaying area works correctly.

STEP 1: ADD ON THE FOLDER STRUCTURE FOR YOUR WEB PAGE

Open the previous **resources** folder. Inside this folder, create the following:

1. An additional folder named **js**

The **js** folder will contain the JS files needed to make your web page actually work.

Finally, open the **js** folder you just created. Inside this folder, create the following:

1. A new file named **index.js** (use your preferred text editor)

The **index.js** file will contain all of the JS code that will be applied to your web page.

At a high-level, here's what it should look like:

-ExampleProject(directory)

Index.html

-Resources(directory)

-Css (directory)

Index.css

-JS(directory)

Index.js

STEP 2: LINK YOUR HTML FILE AND JS FILE

In the <head>section, link the JS file to the HTML file.

```
<link href="./resources/js/index.js" type="text/javascript" >
```

This is because you must specify exactly where the **index.js** file lives within your folder(s) *relative* to where **index.html** lives (otherwise, the two files won't link).
Note: again, make sure to save your changes!

-Great work -

STEP 3: ADD FUNCTIONALITY TO YOUR WEB PAGE

Firstly, open the **index.js** file in your preferred text editor. Next, add the required JS code.

REVIEW TIME!

Congrats! In this short time, you learned how to take what you've been learning on Codecademy/W3schools and apply it using the tools you have

1. Add JS to respective files
2. Link the HTML and CSS and JS files together
3. View your web page in a browser (and refresh the browser to view new changes)

You have now a fully functional website! Why not make it more professional?

Let's continue learning new skills!

Guide: Let's improve our journey in Frontend learning to develop your website and beautify it.

Bootstrap:

Bootstrap is is the most popular CSS Framework for developing responsive and mobile-first websites.

We recommend you to go to their official website https://getbootstrap.com/docs/4.1/getting-started/introduction/

Note: don't forget to read about linking bootstrap with your project.

➔ subjects :

- Layout

- Content
- Components (v.imp)
- Utilities

jQuery:

The purpose of jQuery is to make it much easier to use JavaScript on your website.

We recommend you this Code Academy course https://www.codecademy.com/learn/learn-jquery

Note: don't forget to read about linking jQuery with your project.

To whom who always has thirst for success in his journey

❖ Project time.

Front End Project:

#partOne

Project output: in the previous project replace all the JavaScript text with the required jQuery text.

#partTwo

Project output: in the previous project replace all the CSS text with the required Bootstrap components.

P.S: When you're done with Bootstrap you no longer need your .css file.

To whom who always has thirst for success in his journey

REVIEW TIME!

Congrats! Now you have a well professional website. You amazingly learned how to:

Make a webpage **easy to use, responsive and browser compatible.**

Congratulations fresh star!

You've gained a really good knowledge. And you did an outstanding job.

SITech is so proud of you for making every effort to achieve that job. We're sure it was worth it, knowing that your achievements will make such a positive difference in your onboarding path. Well done!

You worked hard and proved to yourself and everyone what you are capable of.

-SITech Team

To whom who always has thirst for success in his journey

Keep concentrating. We'll begin with the back end programming language.

Ladies and gentlemen we represent to you Python Programming language!

Python:

We recommend you this Code Academy course: https://www.codecademy.com/learn/learn-python.

With this course, you will start to get into the dark side. Once started, you will start coding for real, getting to know the verge of the Python language and its syntax.

We also recommend always keep an eye on the Python documentation for any reference linkage in the link below: https://docs.python.org/3/tutorial/index.html

Django Framework:

We recommend you this link tutorial:
https://tutorial.djangogirls.org/en/

This link is provided by Django Girls volunteers that offer free workshops in many cities of the world.It is aimed at complete beginners, teaching about HTML, CSS, Python and Django.
Please follow with their tutorial .

When you complete Django Girls tutorial.Please show it to your team leader your output.

❖ Project time.

Project output: create a new django file and create a todolist project.the project will list the user's tasks and enable him to change task status (done/not done). Provide editing and deleting for each task.Good luck!

Model name: task

Model fields:

- Title (string)
- Creator (user)
- Done (boolean)
- Description (string)

To whom who always has thirst for success in his journey

Before you get into more progress stuff, let us make sure you have an understanding of the following subjects.

On the upcoming missions you're going to use REST framework and API's

So what are these things and what do they do ?

To get the hook of these stuff you have to understand it's basics and what is it built on

For instance, we will be getting into Http requests and how to send and receive data smoothly.

First, let us get to know what a HTTP request means and what's its methods?

Check out this documentation to get the basics understanding of these request and what each of the does :

http://berb.github.io/diploma-thesis/original/021_www.html

Great, now let's read more about REST API's and get to know it's concept before we go deep into it's documentation using Django

https://spring.io/understanding/REST
https://searchmicroservices.techtarget.com/definition/RESTful-API

To whom who always has thirst for success in his journey

After you're done, all the instructions will be taken directly from your beloved instructor.

Well, Till here our work is done !

We wish you all the best of luck in your journey

And always keep this in mind:
CARPE DIEM
Which means
Seize the day.

Yours Truly,
Your Fellows

To whom who always has thirst for success in his journey