



FULL-STACK DEVELOPMENT

First Program Assignment for Technical Review

Congratulations on making it this far! Before starting on-campus classes, the next step is to start studying the ITC online course followed by submitting your first program assignment for technical review. This is the first part of the actual course!

Your first assignment is to build the front-end of your personal portfolio website. We provide you with a starter code base and a list of tasks to complete, each of which require you to add to or modify the starter code. The starter code is located in this folder:

https://drive.google.com/file/d/1im6TXY2rTsBQszxeMGUulAQ1lIPiujj_/view

To prepare for the assignment, please watch this video series: <https://frontendmasters.com/bootcamp/>. The video series teaches the fundamentals for front-end development, particularly HTML, CSS, and JavaScript. **The tasks in the ITC assignment are based on topics covered in the video series.** The video series has several projects. Although we encourage you to actively build the projects in the video series, you do not need to submit them. The only work you need to submit is the ITC project.

For the ITC project you submit to us, please **make sure you understand all of the code** (line-by-line) in the assignment. You will have to explain your code to an ITC staff member over live video in a technical interview. You may want to practice explaining your code out loud before your technical interview.

Please name the folder as follows: **LastName_FirstName_ITCPrecourse** (e.g. Doe_Jane_ITCPrecourse).

Please do not share your code. You need to keep it entirely confidential. Moreover, although you may use the internet to look up information, you are not permitted to work with others; all work must be done independently (remember, you're going to have to explain your work).

If you have questions, please contact Mor Mika the program manager directly at email: mormika@itc.tech

Good luck and have fun!

BEFORE YOU START

- There is useful information spread across the totality of this document. Make sure you read all of it, including the FAQ.
- Make the best out of it.
- The design and implementation of the style in the HTML pages is up to you. You are required to make it look 'designed'. That means, elements should be properly aligned, colors should be consistent across the pages, and needs to be pleasant to the eye. You can look for inspiration online.
- For that reason, we attach an example to provide guidance (see 'Example' folder in the zip provided). It doesn't mean it needs to look exactly that way, rather it is provided 'as is'.

INSTRUCTIONS

Download the starter code, and begin your work using your favourite editor or IDE (throughout the program we will primarily be using [Microsoft Visual Studio Code](#), but you are free to use other tools such as WebStorm or Atom). For a quick walkthrough on installing Visual Studio Code, you can watch the operating system-appropriate "Installing Visual Studio Code" video [here](#).

INDEX.HTML

1. In index.html, change the text of the element with id="my-name" from "Israel Tech Challenge Pre-Course" to your full name. The syntax "my-name" for the id is known as "kebab case". Change it to "myName", which is camelCase and is used for ids. Next, add a subtitle below the title. The text of the subtitle should say "ITC Fullstack Development." For the font family of your project, choose a font from Google fonts (<https://fonts.google.com/>) or attach one of your choice to the styles file using *@font-face* rule. Otherwise, add styling of your choosing to the page. Throughout this assignment, please add your code for the styling to the "style.css" file. Also, throughout this assignment, we expect you to make your styling look **professional** and **eye-catching**.
2. In index.html, remove the "Visit ITC" link. Add a navigation bar at the top of the page. This navigation bar should be fixed to the top, always taking 100% of the width, and no matter if you scroll, it should always be visible. The navigation bar must have three links: "Home", "About", and "Contact". Use an `<a>` tag to link to the pages. Please add the style of your choosing to the navigation bar and links.

3. Add three clickable elements to your page. Place them anywhere. These three clickable elements should be inside an unordered list as list elements. Each one of this list elements should be clickable. Link one to your LinkedIn account. Link another to your Github account. Link the third to your StackOverflow account. If you don't have accounts for each of those, please create them (www.linkedin.com, www.github.com, www.stackoverflow.com). Make sure you have a photo uploaded to your account profile for Github. For style, use icons from Font Awesome (www.fontawesome.com).
4. In portfolio.js, change the value of GITHUB_URL to your Github profile. Then, change the code so that it displays your Github profile picture instead of the ITC Github profile picture. Also, change the text of the element with id="myName" so that it displays your profile name from your Github account instead of hardcoding your name.
5. On your home page, add functionality to the card containing your profile photo and buttons. When the user hovers over the card, have the card respond with some sort of animation. Trigger a second, more elaborate animation, upon clicking the card.

See the image 'index.png' in the 'Examples' folder for reference.

CONTACT.HTML AND ABOUT.HTML

6. In these two files, add the same navbar from your home page to each of the new pages you created. Add some basic HTML to each page so that the about page has a title that says "About" and the contact page has a title of "Contact".
7. In your navbar, connect the navbar links to their respective HTML pages. When you click on the links, your browser should display the corresponding HTML page.

ABOUT.HTML

8. On the "About" page, create a section that includes the following information: describe why you are interested in coding, your educational and career experiences, and your personal interests. Please style and layout the page as you choose.
9. Add another section below what you created in step 6. In this section, write where you were born and the cities in which you have lived. If you have lived in two cities or less, list two other cities that you want to visit. This section should also contain an embedded Google Map with a marker on the city where you were born.
10. Add two buttons: "Previous" and "Next" below the map. Clicking the "Next" button should change the pointer to the next city listed on your about page (i.e., a city you have lived in or

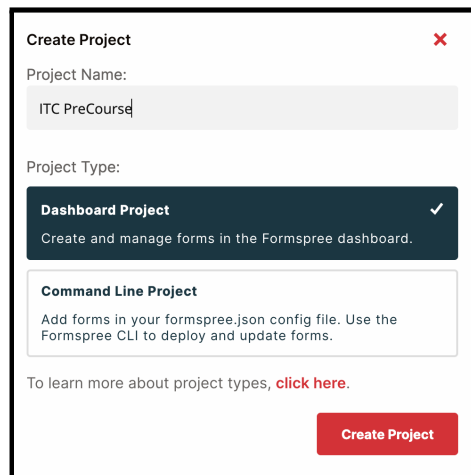
want to visit). When the first of the series of locations is selected, the "Previous" button should disappear, and when the last location is selected, the "Next" button should disappear. Otherwise, both buttons should be visible. We encourage you to implement this using only one iframe.

11. Without deleting the code inside, rename the itcprecourse.js file to "portfolio.js" and make sure your code reflects the change. Have in mind that after changing the file name, all the previous files linked to it should still be linked to it.

See the image 'about.png' in the 'Examples' folder for reference.

CONTACT.HTML

12. Sign in for a free account in Formspree. ([link](#)). Formspree is a free service that allows us to create forms, and receive the information in our email. Create a new project for it named "ITC Precourse". Don't worry, we are attaching snapshots on how to solve this part. From the options, choose the dashboard one:



Create Project [Close]

Project Name:
ITC PreCourse

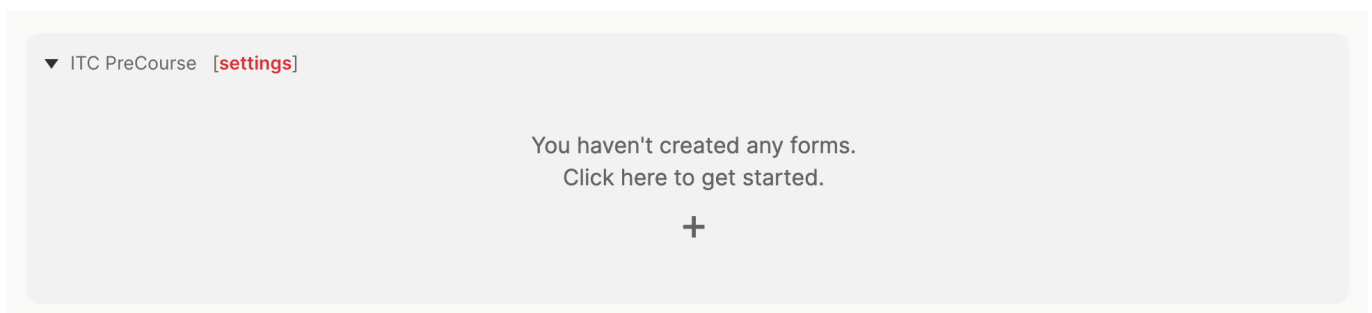
Project Type:

- Dashboard Project** ✓
Create and manage forms in the Formspree dashboard.
- Command Line Project
Add forms in your formspree.json config file. Use the Formspree CLI to deploy and update forms.

To learn more about project types, [click here](#).

Create Project

Add a new form to it by clicking on the plus:



▼ ITC PreCourse [settings]

You haven't created any forms.
Click here to get started.

+

And create the from:

Create Form



Form Name:

contact form

Project:

ITC PreCourse

Send emails to:

@itc.tech

To send to a new email address, please first add it to Linked Emails on the **account** page.

Create Form

Grab your form endpoint:

Your form's endpoint is:

`https://formspree.io/f/xvoda`

Copy

13. After grabbing the endpoint, on the "Contact" page, create a contact form. This form would use the Formspree endpoint to post the information. The form should contain inputs for first name, last name, email, comment. There should be an additional input type where the user can enter additional information in a non text form. All fields except the "last name" field should be required.

Add a submit button to your form. When the submit button is clicked, you should actually submit your form, and get an email with the information of your submitter (provided by Formspree)

14. The submit button should not be functional and should look disabled as long as the required fields are not filled in.

See the image 'contact.png' in the 'Examples' folder for reference.

IN ALL THE PAGES (INDEX, ABOUT, CONTACT)

15. On all of your pages, put a footer at the bottom of the page. The footer should contain the same links as your navbar. It also should contain a sentence saying: "This page was built using: HTML, CSS and JavaScript". The words "HTML", "CSS", "JavaScript" should be stored in an array

and generated on the website rather than be "hard-coded". The word "and" should not be in your array, but it should be in your sentence.

Create the array in a way that if more languages were added to the array, the code wouldn't need any extra modifications.

Your solution needs to work with all of the following arrays interchangeably:

- a. `const codingLanguages = ["HTML"]`
- b. `const codingLanguages = ["HTML", "CSS", "JavaScript"]`
- c. `const codingLanguages = ["HTML", "CSS", "JavaScript", "Python", "Java", "C++"]`

The original array must preserve its original characters without modifications (no adding spaces, commas, or any other character).

The sentence generated should have a space after each comma, except for the last element, which should have " and " before. e.g. `["HTML", "CSS", "JavaScript"] => "This website has been created with: HTML, CSS and JavaScript"`

FREQUENTLY ASKED QUESTIONS (FAQ)

1. How do I fix the layout and styles on my page?

The Frontend Masters videos have great information about HTML and CSS. If the content in those videos do not solve your problems, find examples on W3Schools, CSS Tricks, and StackOverflow. W3Schools and CSS Tricks have great examples, templates, and descriptions, whereas StackOverflow is great for targeting specific problems.

To find the answers you're looking for, try Googling a description of your problem. For instance, "how to create a fixed navbar with items centered HTML CSS" or "how to horizontally align HTML elements using CSS."

Many ways exist for organizing and styling HTML elements. You might need to wrap some of your HTML elements inside of other elements, use certain style properties to direct the browser to override an HTML element's default style properties, or maybe some combination of those and other things. Remember to keep it simple and focus on the things that you know how to do.

2. How do I fix my JavaScript code that's not working?

As you write your JavaScript code, at each step along the way, use `console.log` to better understand what is happening. You can also learn how to use the debugger ([link](#)). For instance, if you're using a variable as a counter each time a button is clicked, console log the variable so that you can see the value change with each button click. Or if you're pulling an HTML element into a JS file, console log it to make sure it's working like you think it should.

3. Do I need to use only one iframe for the Google Maps part of the assignment?

Ideally, yes. You can build the Google Maps and corresponding buttons using only one iframe to cycle through your locations. You can, however, use multiple iframes like one of the examples does from the Frontend Masters videos. We recommend getting your map to work with the previous and next buttons in whichever way you can, and then you can optimize your code after you get it working.

4. Do I need an API key to use Google Maps?

No need for an API key. You can use Google Maps without one. The Frontend Masters videos have a great explanation on how. You also can look at [this link](#) and [this link](#) for more information.

5. How do I fetch information from my Github account instead of hard coding it?

Your JavaScript file should send a request to the Github API. The code you need for this is in the starter code you downloaded from us. You need to change that code to get your Github information. Then, use console log to look at the response from the request to find your Github profile information. Display some of that profile information in the browser instead of whatever is currently there for the profile name and image. The Frontend Masters videos on working with APIs should help.

6. For the footer at the bottom of the page, how do I put the word "and" in the sentence without putting it in the array?

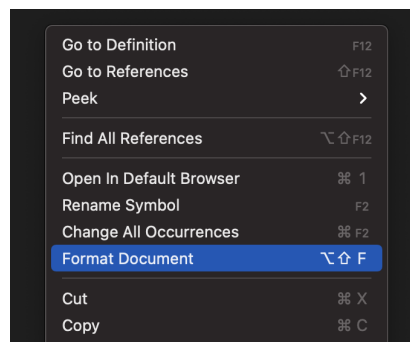
This is part of your challenge! The Frontend Masters videos on arrays, index positions, and loops should be an excellent starting point. Plus, you'll need to use conditional statements, operators, and string concatenation to "build" your sentence.

BEFORE YOU SUBMIT

Before you submit, make sure that you check all the following boxes:

- ☐ You have read the FAQ.
- ☐ Submission is compressed in zip format (not .rar)
- ☐ Your submission file name has the following format: **LastName_FirstName_ITCPrecourse.zip**
- ☐ In your HTML files, you don't have any undesired overflowing content. (What is overflow? [link](#))
- ☐ All your files are properly formatted (indented)

If you don't know how to do it, Visual Studio Code has a tool for it:



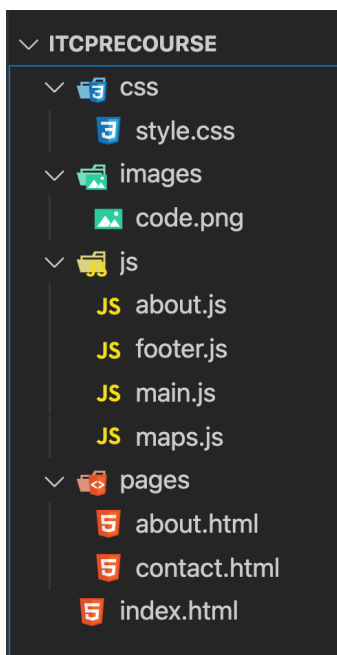
- ☐ All your links are relative rather than absolute: (Want to know more? [link](#))

```
<div class="navlinks">
  <a href="./index.html">Home</a>
  <a href="./about.html">About</a>
  <a href="./contact.html">Contact</a>
</div>
```

- ☐ Your HTML files don't contain errors. You can use [this website](#) to check that your HTML files are valid, as well as any other tool of your convenience.
- ☐ You are able to explain in depth the totality of your code.
- ☐ Your JS files don't create errors in the console:

```
✖ ▶ Uncaught TypeError: Cannot read property 'addEventListener' of null    contact.html:194
    at contact.html:194
```

- ☐ Your files and folders are structured in an organized way, like the following (but not exclusively):



SUBMIT YOUR WORK

Check your work, zip it up (remember the specified format), [submit it and schedule your technical review](#)! Please ensure you are prepared to explain and answer questions on the code submitted.

Link to submit and schedule review:

<https://app.acuityscheduling.com/schedule.php?owner=12116384&appointmentType=1155132>

Looking forward to hearing from you!