Project - Simple Interest Calculator
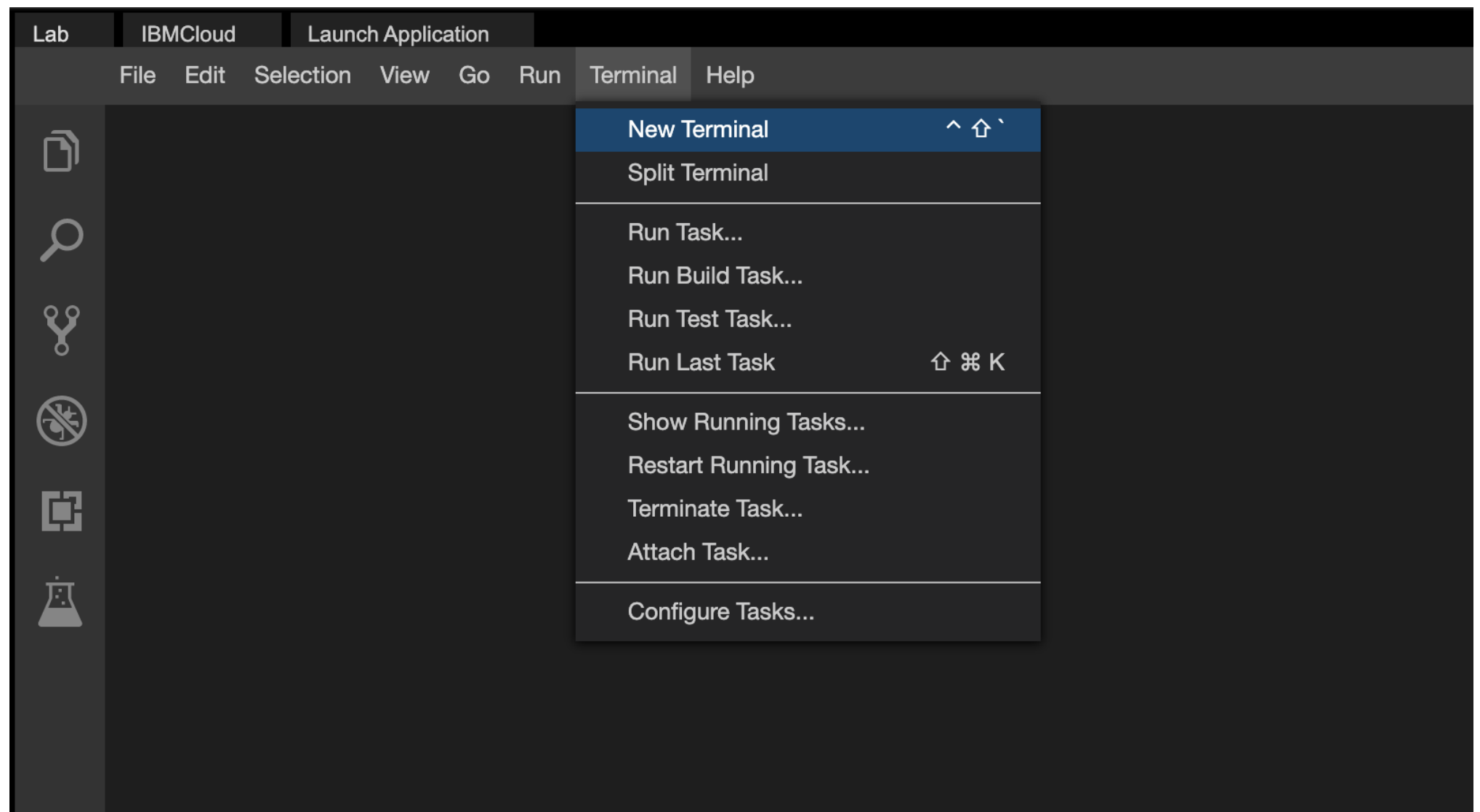

Objectives

1. Download the project repo.
2. Modify the html file as per requirements.
3. Modify the css file as per requirements.
4. Modify the javascript file as per requirements.
5. Verify if the app is working properly.
6. Submit the releavnt screenshots of the app for the peer review.

Exercise 1 : Download the project folder.

1. Open a new terminal.

2. Use the wget command to download the project folder and then unzip it.

```
wget https://github.com/ibm-developer-skills-network/vftvk-Simple-Interest-Calculator/archive/refs/heads/master.zip
```

```
unzip master.zip
```

3. cd to the project folder

```
cd vftvk-Simple-Interest-Calculator-master
```

4. Verify that the project contains the following files:

5. index.html

6. style.css

7. script.js

8. You can view your app by viewing your index.html file using Live Server. If you need help to use the Live Server, Click here.

Exercise 2 : Modify the html file.

In this exercise, you will correct any mistakes in the existing code and also add any missing tags.

1. On the file explorer navigate to the index.html.
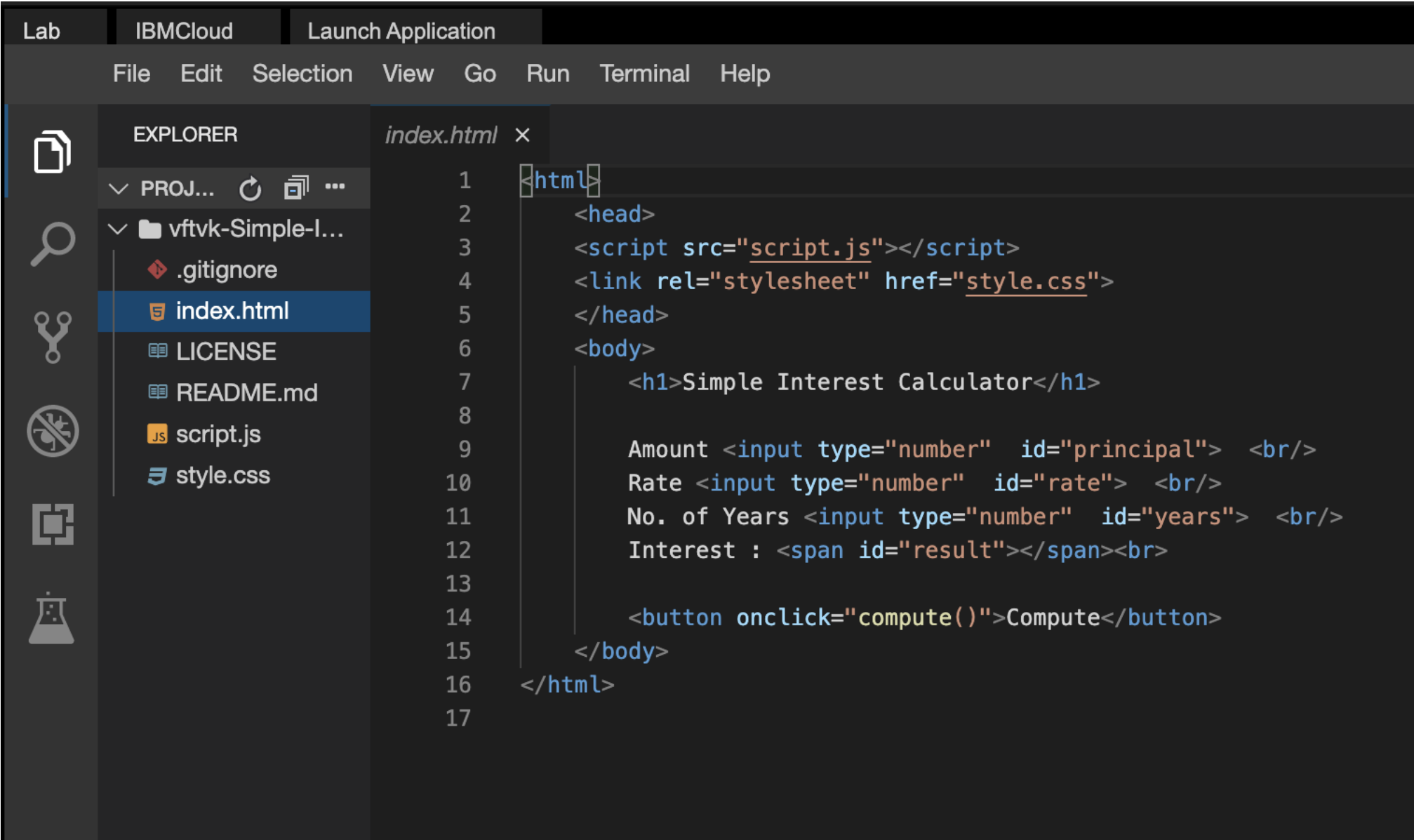
1. Doctype is missing. Add the doctype.

```
<!DOCTYPE html>
```

2. Add "Simple Interest Calculator" as title.

```
<title>Any relevant title you want to give</title>
```

3. Move all the content which currently is in the `<body>` to a `<div>` tag.

4. Set the class attribute of this div to 'maindiv'.

```
<div class="maindiv">
.
.
.
.
</div>
```

5. Modify the input text box for interest rate to a slider using the `<input type="range">` tag.

```
<input type="range"  id="rate" min="minval" max="maxval" value="default_val">
```

6. For the rate input, set the attributes min to 1, max to 20 and step to 0.25 and the default value to 10.25.

Range is an elegant way to input numeric input. But the drawback is that it does not visually show value the user has selected.

7. To show the value selected by the `range`, create a `<span>` element right after the range.

```
<span id="rate_val">
</span>
```

8. Inside the '`` tag add the text "10.25 %".

```
<span id="rate_val">
10.25%
</span>
```

9. Modify the input text box for "No of years" into a dropdown box with options 1 to 10.

```
<select id="years">
    <option value="1">1</option>
    <!-- fill in the rest of the values-->
</select>
```

10. Change the name of "Compute" button to "Compute Interest".

11. Below the "Compute Interest" button create an empty `<span>` and set its id to 'result'.

```
<span id="result">

</span>
```
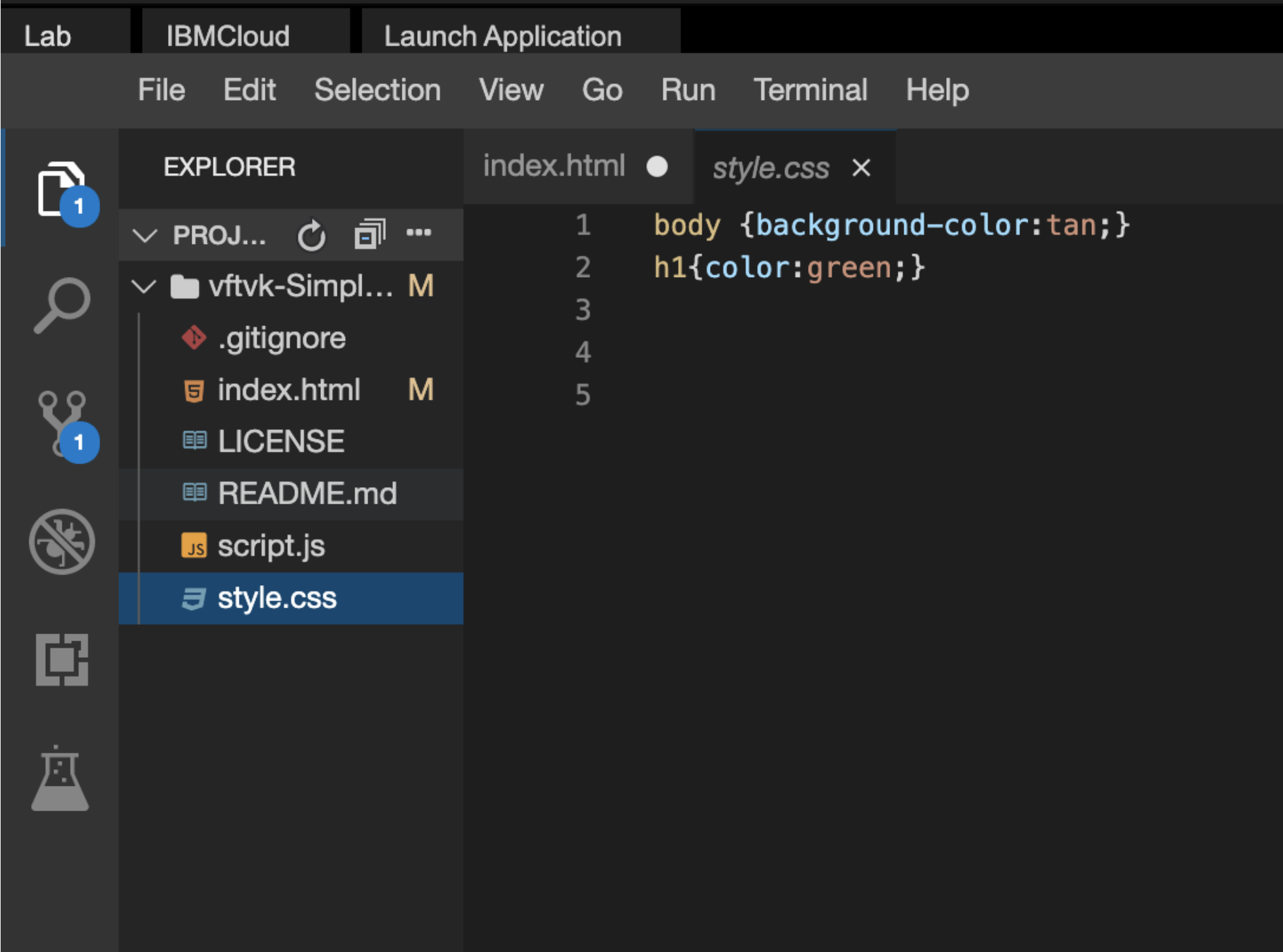
12. Add a copyright message using the `<footer>` tag.

```
<footer>

    &#169; This Calculator belongs to --your name--


</footer>
```

Exercise 3 : Modify the css file.

In this exercise, you will correct the look and feel of the web page.

1. On the file explorer navigate to the style sheet.

2. Set the body color to 'black', font to arial and font color to white.

```
body {background-color:black;font-family:arial;color:white}
```

3. Set the `<h1>` color to 'grey' and font to verdana.

```
h1{color:grey;font-family:verdana}
```

4. Create an entry for class 'maindiv'.

```
.maindiv {
}
```

5. Set the background color to white, font color to black, width to 300 pixels, and padding to 20px in the newly created maindiv class.

6. Set the border radius to 25 px.

7. Align the div to the center of the page.

▶ Click to see how the new class should look like

8. Visualize your html file in the Live Server and make sure that you have not missed anything.

Exercise 4 : Modify the javascript file.

In this exercise, you will write the javascript code in the file **script.js** to implement the simple interest calculation logic.

1. Before we start writing any javascript, make sure that the button in html file has the onclick attribute set to "compute()"

2. Create a variable called 'principal' and assign to it the value of the input element "principal"

```
var principal = document.getElementById("principal").value;
```

3. Create a variable called rate and assign to it the value of the input element "rate"

```
var rate = document.getElementById("rate").value;
```

4. Create a variable called years and assign to it the value of the input element "years"

```
var years = document.getElementById("years").value;
```

5. Create a variable called interest and assign to it the value of `principal * years * rate / 100`

```
var interest = principal * years * rate /100;
```

6. Write the logic to convert the 'No of Years' into the actual year in the future.

```
var year = new Date().getFullYear()+parseInt(years);
```

7. Write a function that reads the value of the range slider and displays it the `<span>`adjacent to the slider.

```
function updateRate()
{
    var rateval = document.getElementById("rate").value;
    document.getElementById("rate_val").innerText=rateval;
}
```

8. Link this function with an "onchange" event on the range input.

9. Change the slider, and test if the display in the span is being updated dynamically.

10. Get the reference to the element named 'result'

11. When "Compute Interest" is clicked, set its inner html property to the below text.

   If you deposit 100,
   at an interest rate of 3.5%.
   You will receive an amount of 175000,
   in the year 2025

The number above are for indication only. Make sure the output contains the relevant values by using the correct variables

▶ Click here for a sample

12. Make sure that the input you have taken as "No of Years" is converted into an actual year.

13. Make sure the numbers in the result are highlighted.

14. Add validation for "Principal" input box. If the user enters zero or negative values, show an alert "Enter a positive number"

15. Once the user clicks on the alert "OK" button, take the user back to the "Principal" input box, by setting the focus on this box. You can refer to the [Javascript Form Validation lab](#).

16. You are done with the coding.

> 📷 Take a screenshot of your app and save it as a .jpg or .png with the filename `task_1.png`. You will be prompted to upload the screenshot in the Peer Assignement.

Let us proceed to next exercise where we test the code.

17. Write comments in your code. They not only help you score more marks in the project, but also help you to debug and maintain the code in the long term.

Exercise 5 : Test the app.

Now that you have finished coding your app. Let us do some basic testing, before we release it.

1. Enter these values in the form.

   Amount = 0

   Rate = 1

No. of Years = 1

Click on Compute button.

You should see an alert "Enter a positive number".

> 📷 Take a screenshot of your app and save it as a .jpg or .png with the filename `task_8.png`. You will be prompted to upload the screenshot in the Peer Assignement.

2. Enter these values in the form.

Amount = 1000

Rate = 10%

No. of Years = 10

Click on Compute button. You should see the following output.

```
If you deposit 1000,<br>
at an interest rate of 10%.<br>
You will receive an amount of 1000,<br>
in the year 2032<br>
```

The year is 2022 at the time this lab was written, 2032 is therefore correct.

> 📷 Take a screenshot of your app and save it as a .jpg or .png with the filename `task_6.png`. You will be prompted to upload the screenshot in the Peer Assignement.

3. Enter these values in the form.

Amount = 4800

Rate = 15.25%

No. of Years = 5

Click on Compute button. You should see the following output.

```
If you deposit 4800,<br>
at an interest rate of 15.25%.<br>
You will receive an amount of 3660,<br>
in the year 2027<br>
```

The year is 2022 at the time this lab was written, 2027 is therefore correct.

> 📷 Take a screenshot of your app and save it as a .jpg or .png with the filename `task_7.png`. You will be prompted to upload the screenshot in the Peer Assignement.

If you app has passed all the test cases. Then you can proceed to your peer review submission.

Otherwise, start debugging your code to see where you went wrong.

When you are done with all the changes in the project, submit the relevant screenshots for the peer review and download the below files:

1. index.html
2. style.css
3. script.js

Author(s)

Ramesh Sannareddy

Other Contributor(s)

Rav Ahuja

Changelog

Date	Version	Changed by	Change Description
2022-02-18	2.0	Samaah Sarang	Instructions updated
2020-09-09	1.2	Lavanya	Added some pics and explicit instructions
2020-09-09	1.1	Ramesh Sannareddy	Added 'publishing to gitpages'
2020-08-23	1.0	Ramesh Sannareddy	Initial version created