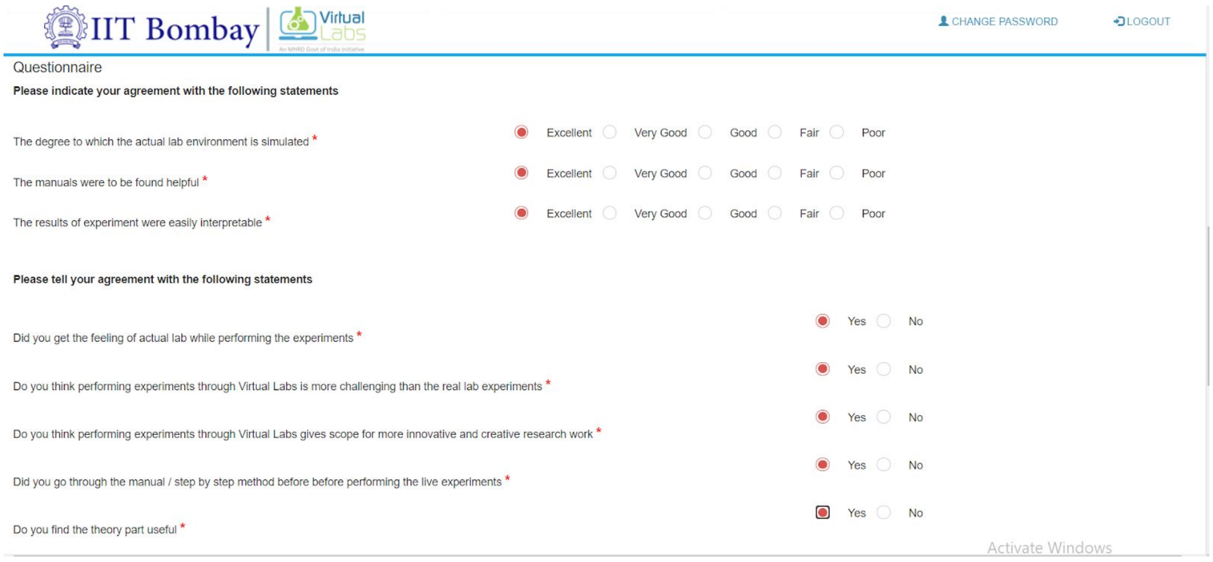
VLAB FEEDBACK:



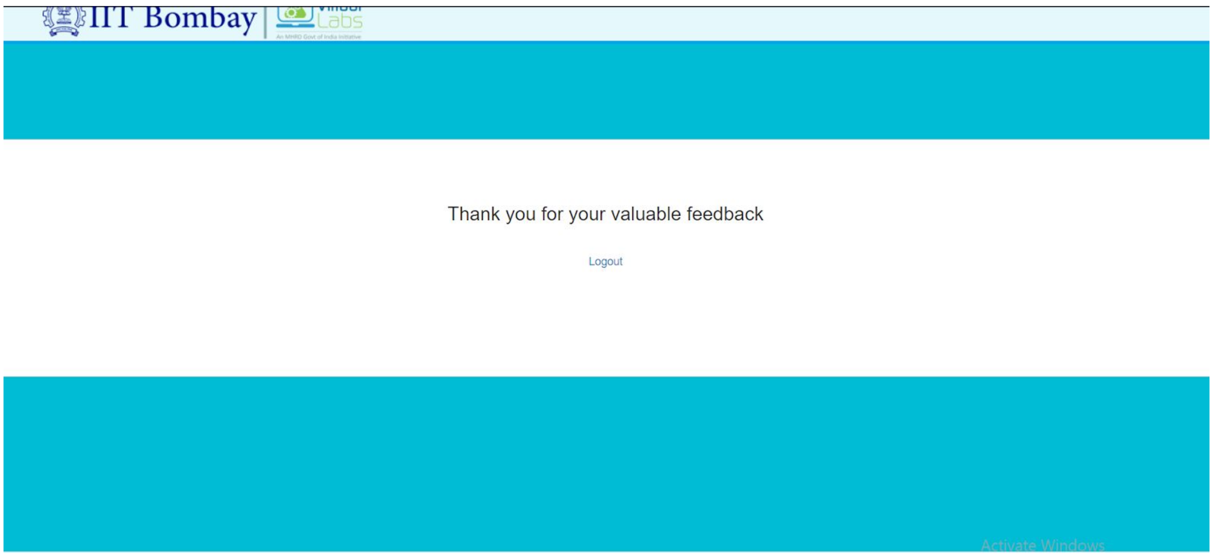
Questionnaire

Please indicate your agreement with the following statements

The degree to which the actual lab environment is simulated *	[✓ Excellent]	Very Good	Good	Fair	Poor
The manuals were to be found helpful *	[✓ Excellent]	Very Good	Good	Fair	Poor
The results of experiment were easily interpretable *	[✓ Excellent]	Very Good	Good	Fair	Poor

Please tell your agreement with the following statements

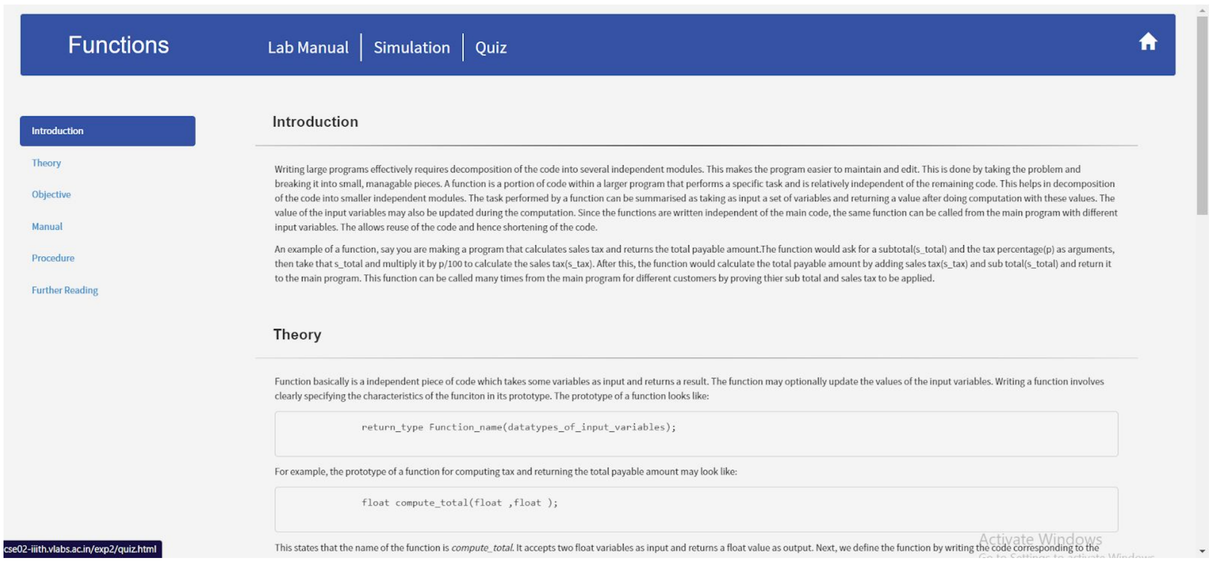
Did you get the feeling of actual lab while performing the experiments *	[✓ Yes]	No
Do you think performing experiments through Virtual Labs is more challenging than the real lab experiments *	[✓ Yes]	No
Do you think performing experiments through Virtual Labs gives scope for more innovative and creative research work *	[✓ Yes]	No
Did you go through the manual / step by step method before before performing the live experiments *	[✓ Yes]	No
Do you find the theory part useful *	[✓ Yes]	No



Thank you for your valuable feedback

Logout

VLAB Experiments:



Introduction

Writing large programs effectively requires decomposition of the code into several independent modules. This makes the program easier to maintain and edit. This is done by taking the problem and breaking it into small, managable pieces. A function is a portion of code within a larger program that performs a specific task and is relatively independent of the remaining code. This helps in decomposition of the code into smaller independent modules. The task performed by a function can be summarised as taking as input a set of variables and returning a value after doing computation with these values. The value of the input variables may also be updated during the computation. Since the functions are written independent of the main code, the same function can be called from the main program with different input variables. The allows reuse of the code and hence shortening of the code.

An example of a function, say you are making a program that calculates sales tax and returns the total payable amount.The function would ask for a subtotal(s_total) and the tax percentage(p) as arguments, then take that s_total and multiply it by p/100 to calculate the sales tax(s_tax). After this, the function would calculate the total payable amount by adding sales tax(s_tax) and sub total(s_total) and return it to the main program. This function can be called many times from the main program for different customers by proving thier sub total and sales tax to be applied.

Theory

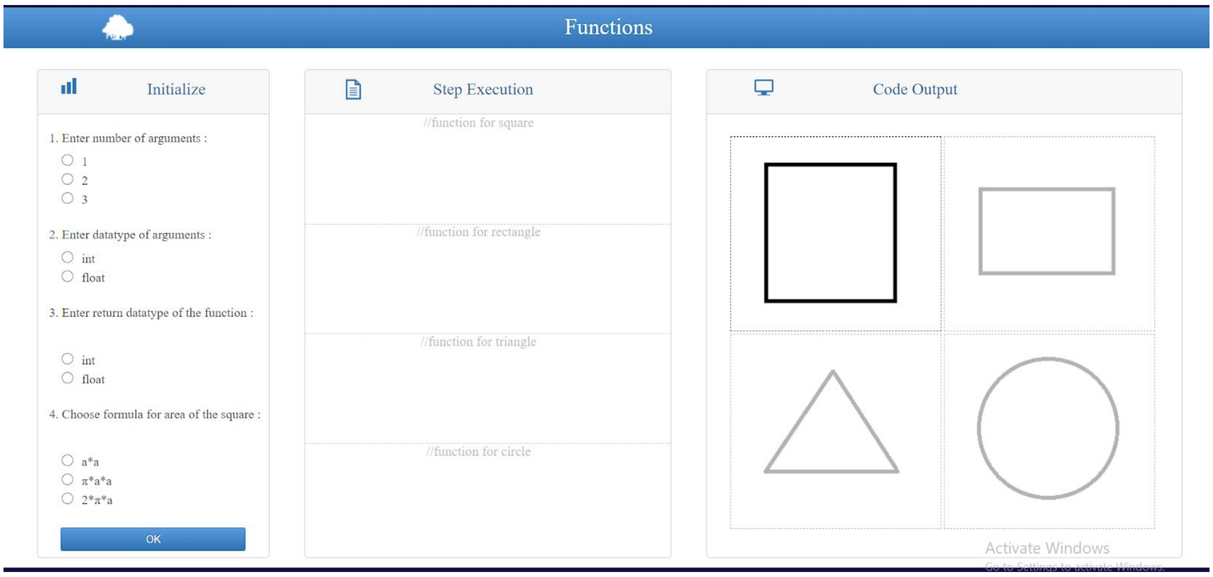
Function basically is a independent piece of code which takes some variables as input and returns a result. The function may optionally update the values of the input variables. Writing a function involves clearly specifying the characteristics of the funciton in its prototype. The prototype of a function looks like:

```
return_type Function_name(datatypes_of_input_variables);
```

For example, the prototype of a function for computing tax and returning the total payable amount may look like:

```
float compute_total(float ,float );
```

This states that the name of the function is *compute_total*. It accepts two float variables as input and returns a float value as output. Next, we define the function by writing the code corresponding to the



1. Enter number of arguments :
   - ○ 1
   - ○ 2
   - ○ 3
2. Enter datatype of arguments :
   - ○ int
   - ○ float
3. Enter return datatype of the function :
   - ○ int
   - ○ float
4. Choose formula for area of the square :
   - ○ a*a
   - ○ π*a*a
   - ○ 2*π*a