

# Web Basics - JavaScript

## Lab Book

### Table of Contents

<b>Getting Started.....</b>	<b>3</b>
Overview.....	3
Setup Checklist for JavaScript .....	3
<b>Lab 1: Basics Concepts of JavaScript .....</b>	<b>4</b>
1.1: Create a page to display “Welcome to JavaScript” .....	4
1.2: Create prob2.html to display Formatted Hello World by using JavaScript by embedding Hello World in <H1> tag. ....	5
1.3: Create page to show use of external JavaScript.....	6
1.4: Using Variable in many Script tags .....	9
<b>Lab 2: The JavaScript Language .....</b>	<b>12</b>
2.1: For loop in JavaScript.....	12
2.2: Create a web page to calculate the Compound Interest using the formula given below:.....	13
<b>Lab 3: Working with Predefined core objects .....</b>	<b>14</b>
3.1: Displaying Date using Date Object.....	14
3.2: Using indexOf function of String object .....	15
3.3: Using various String methods .....	15
<b>Lab 4: Working with Arrays.....</b>	<b>16</b>
4 .1: Using Array to display values .....	16
<b>Lab 5: Working with Document Object Model(DOM) .....</b>	<b>Error! Bookmark not defined.</b>
5 .1: Window object .....	<b>Error! Bookmark not defined.</b>
<b>Lab 6: Working with Document Object .....</b>	<b>17</b>
6.1: Working with Documents.....	17
<b>Lab 7: Working with Form Object.....</b>	<b>20</b>
8.1: Form Validation .....	20
8.2 Validate Field .....	21
<b>Appendices .....</b>	<b>Error! Bookmark not defined.</b>
Appendix A: JavaScript Standards .....	<b>Error! Bookmark not defined.</b>

## WEB BASICS – JAVASCRIPT LAB BOOK

Appendix B: Coding Best Practices.....	<b>Error! Bookmark not defined.</b>
Appendix C: Table of Figures.....	<b>Error! Bookmark not defined.</b>
Appendix D: Table of Examples .....	<b>Error! Bookmark not defined.</b>

## Getting Started

### Overview

These Lab book is a guided tour for Learning JavaScript. It contains solved examples and To Do assignments. Follow the steps provided in the solved examples and then work out the 'to do' Assignments given.

### Setup Checklist for JavaScript

Here is what is expected on your machine in order for the lab to work.

#### Minimum System Requirements

- Hardware: Networked PCs with minimum 64 MB RAM and 60 MB HDD.
- Software: Window based Operating System having the latest version of Internet Explorer (IE) or Netscape Navigator installed.

#### Please ensure that the following is done:

- A text editor like Notepad, Eclipse Luna or Visual Studio 2008 is installed.

### Instructions

- For coding standards refer Appendix – A.
- All Lab assignments should follow the coding standards.
- Create a directory by your name in drive <drive> for JavaScript assignments.
- In this directory, create subdirectory javascript\_assgn.
- For each lab create directory as lab<lab number>.

### Learning More (Bibliography if applicable)

- Beginning JavaScript by Paul Wilton
- JavaScript: The Definitive Guide by David Flanagan
- JavaScript Application Cookbook by Jerry Bradenbaugh

**Lab 1: Basics Concepts of JavaScript**

<b>Goals</b>	<ul style="list-style-type: none"><li>• Learn to embed script tags in different parts of the HTML document.</li></ul>
<b>Time</b>	120 minutes

**1.1: Create a page to display “Welcome to JavaScript”.****Solution:**

**Step 1:** Complete the following code and save it as prob1.html

```
<!DOCTYPE html>
<html>
<head>
<title> Welcome to JavaScript</title>
</head>
<body>
<script>
document.write("Welcome to JavaScript - The Scripting Language")
</script>
</body>
</html>
```

**Example 1: Lab 1: Prob1.html**

**Step 2:** Start the editor to be used.

**Step 3:** Write the JavaScript program.

**Step 4:** Save the file with extension .html or htm.

**Step 5:** Select **Start → Programs → Internet Explorer**.

Alternatively select **Start → Programs → Netscape Navigator**.

**Step 6:** In the Internet Explorer, select **File → Open → Browse**, and select the file you have just saved.

**Step 7:** Click **OK** in the browser pop-up window.

**Step 8:** Verify that you get the output as shown in the figure given below.



**Figure 1: Welcome to JavaScript**

**Note:** Follow the above steps (3 - 8) for every Lab problem for verifying the output. You can also use other text editors like editplus, WordPad, MS Visual Interdev (if installed) to create your **html** and **.js** pages.

**1.2: Create prob2.html to display Formatted Hello World by using JavaScript by embedding Hello World in <H1> tag.**

**Solution:**

**Step 1:** Create **prob2.html** page to complete the following code and save in lab1 directory.

```
<html>
<head>
<title>Displaying Formatted Text using JavaScript</title>
</head>
<body>

<script>

//TODO: Display hello world embedded in h1 tag with align attribute value right

</script>

</body>
</html>
```

**Example 2: Lab 1: Prob2.html**

**Step 2:** Open **prob2.html** page in the browser, and verify that you get the same output as required.



**Figure 2: Formatting Text in JavaScript**

### 1.3: Create page to show use of external JavaScript

**Solution:**

**Step 1:** Create **Prob3.html** to complete the following code and save it in lab1 directory.

```
<html>
<head><title>Using External Script file in HTML Document</title>

<script src="HelloWorld.js">
</script>

</head>
<body>
<hr>
<p>The actual script is in external script file called "HelloWorld.js"</p>

<script>
//TODO: Insert the code here to invoke the function sayHello() in the file HelloWorld.js
</script>

<hr>
</body>
</html>
```

**Example 3: Lab 1: Prob3.html**

**Step 2:** Create a file **HelloWorld.js** which should have a function **sayHello()** that returns a string "Hello World".

```
function sayHello()
{
//TODO:return the string "Hello World"
}
```

**Example 4: Lab 1: HelloWorld.js**

**Step 3:** Open **prob3.html** page in the browser, and verify that you get the same output as required.

The actual script is in external script file called "HelloWorld.js"

This text is displayed by Calling external function : **hello world**

**Figure 3: Using external JavaScript File**

**Step 4:** Create **Prob4.html** page and complete the following code and save it in lab1 directory.

```
<html>
<head><title>Embedding Script tag in HTML Document</title>

<script>

//TODO:use write method in document object to display the desired output

</script>
<hr>
<script src="Hello.js">
</script>

</head>
<body>

<script>
//TODO: use write method in document object to display desired the output
</script>

<hr>

<p><code>The actual script is in external script file called "Hello.js"</code></p>
```

```
<script>
//TODO: Insert your code here to call the function dispHello() from the Hello.js file
</script>

<hr>

</body>
</html>
```

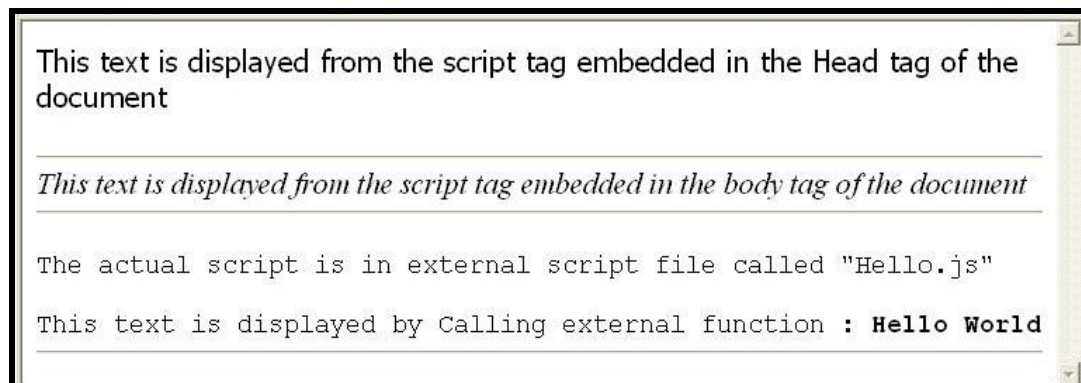
**Figure 4: Lab 1: Prob4.html**

**Step 5:** Create a file **Hello.js** which should have a function **dispHello()** that returns a string "Hello World".

```
function dispHello()
{
//TODO:return the string "Hello World"
}
```

**Example 5: Lab 1: Hello.js**

**Step 6:** Open **prob4.html** page in the browser, and verify that you get the same output as required.



**Figure 5: Embedding Script tags in HTML document**



#### 1.4: Using Variable in many Script tags

##### Solution:

**Step 1:** Create **Prob5.html** page, and complete the following code and save it in lab1 directory.

```
<html>
<head><title>Embedding Script tag in HTML Document</title>

<script>
/*
TODO:define variable headVar and initialize it to some integer value and display the value as
shown in the Fig 6
*/
</script>

<hr>
</head>
<body>

<script>
/*
TODO:define variable bodyVar and initialize it to some integer value and display the value as
shown in the Fig 6
*/
</script>

<hr>
<script src="common.js">
</script>

<script>
/*
TODO: Invoke the method addNos(headVar,bodyVar) defined in common.js file and pass the
two variables headVar and bodyVar defined in the head and the body script tag and display the
added result as shown in the Fig 6
*/
</script>

<hr>
```

```
</body>
</html>
```

### Example 6: Lab 1: Prob5.html

**Step 2:** Create a file **common.js** which has a function **addNos()** that adds two numbers and returns the addition of two numbers.

```
var msg;
msg="<p><code>The actual script is in external script file called common.js</code></p>";

function addNos(headVar,bodyVar)
{

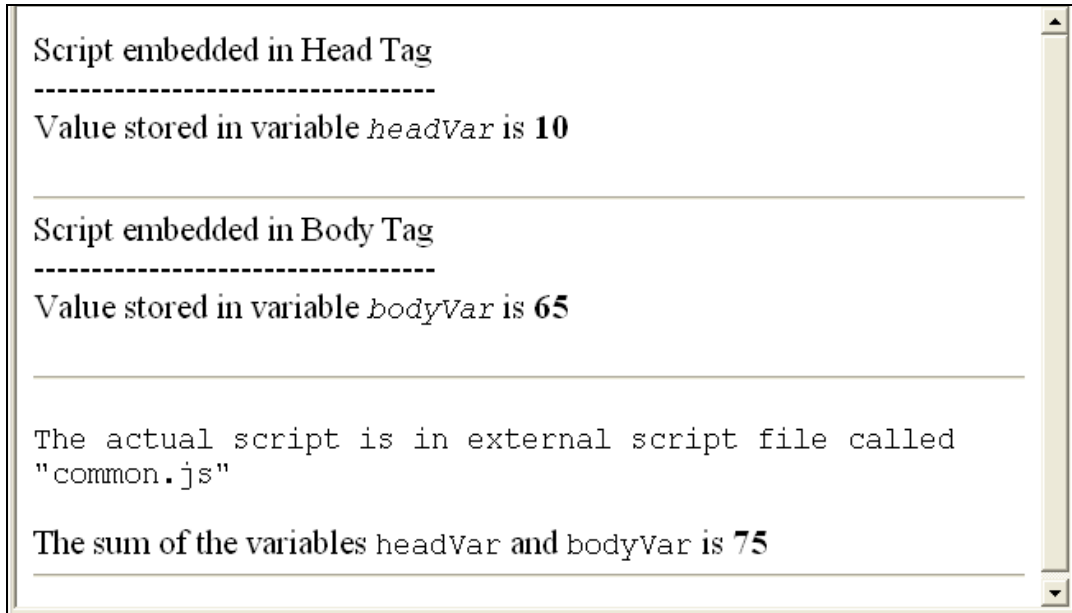
//TODO: display the contents of the variable "msg"

//TODO: display the addition of two numbers

}
```

### Example 7: Lab 1: common.js

**Step 3:** Open prob5.html page in the browser, and verify that you get the same output as required.



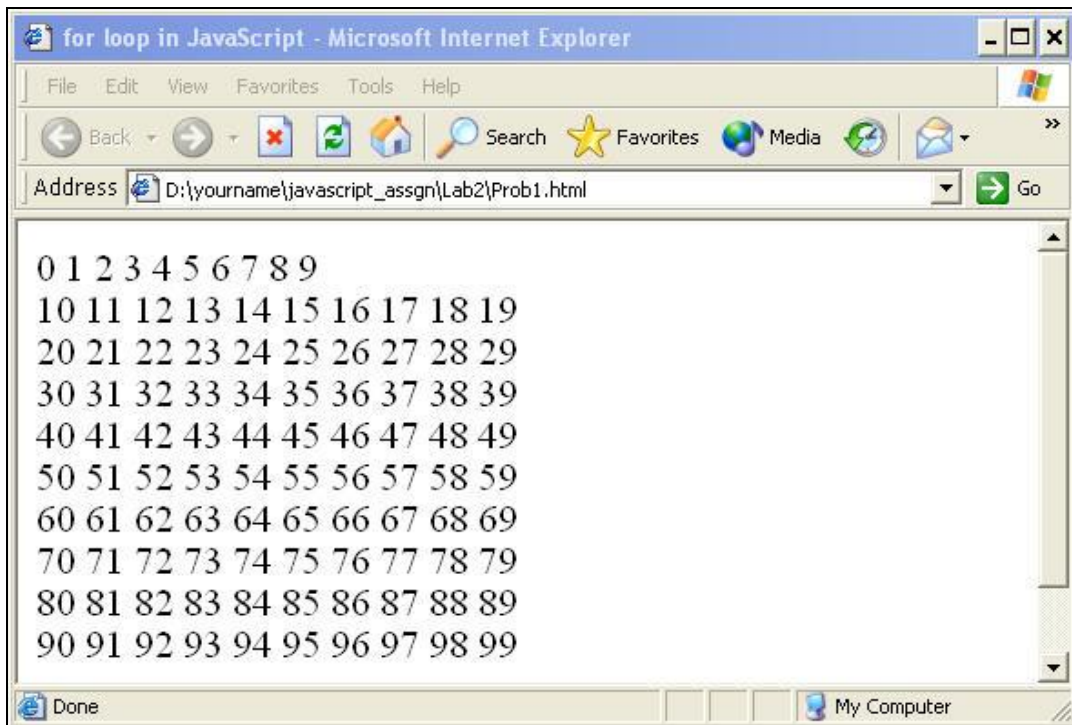
**Figure 6: Using Variable in many Script tags**

## Lab 2: The JavaScript Language

<b>Goals</b>	<ul style="list-style-type: none"> <li>Learn to use looping structures and operators in JavaScript.</li> </ul>
<b>Time</b>	20 minutes

### 2.1: For loop in JavaScript

Create a web page containing a heading “Layout is here” followed by a horizontal rule and a table with a single row as shown in the figure given below.



**Figure 7: For loop in JavaScript**

After completing the loop, the variable used, that is “**i**”, should be equal to **100**.

#### **Solution:**

**Step 1:** Write the code and save it as **Prob1.html** in lab2 directory.

**Step 2:** Open **prob1.html** page in the browser, and verify that you get the same output as required.

**Step 3:** Create **prob1\_dowhile.html** and **prob1\_whiledo.html** page using **do...while** and **while...do** control statements respectively to display similar output as shown in the figure given above.

**2.2: Create a web page to calculate the Compound Interest using the formula given below:**

$$\text{Compound Interest} \quad \left[ P * \left( 1 + \frac{r}{100} \right)^n \right] - P$$

**Where:**

**p = Principal,**

**r = Rate of Interest,**

**n = period in years**

**The values used in the example in the following figure are as follows:**

**P = 1000, n = 1, r = 10**

```

-----
*****Calculate Compound Interest*****
-----

Prinicipal          -      1000 rs
Rate of Interest    -      10%
Period              -      1 yr
Comp Interest       -      100

```

**Figure 8: Operators and Arithmetic Expression**

**Solution:**

**Step 1:** Write the code, and save it in lab2 directory.

**Step 2:** Open page in the browser, and verify that you get the same output as required.

**Lab 3: Working with Predefined core objects**

<b>Goals</b>	Understand Date, String Object
	Learn to use Date and String objects in HTML pages
<b>Time</b>	45 minutes

**3.1: Displaying Date using Date Object**

Create a web page **Prob1.html**. In this web page, create a **date** object and use the **getXXXX** functions of the date object to display today's date in the format as shown below in the figure and also greet the user depending on the time the user visits the page. The message to be displayed is given in the following table. The time column shows the current date hour value.

Time	Msg to be displayed
< 12	Good Morning
>= 12 and <= 17	Good Afternoon
> 17	Good Evening

**Figure 9: Displaying Date using Date Object****Solution:**

**Step 1:** Write the Code, and save it as Prob1.html in lab4 directory.

**Step 2:** Open prob1.html page in the browser, and verify that you get the same output as required.

### 3.2: Using indexOf function of String object

Create a web page **prob2.html**, which uses the **indexOf** method of string object and displays the index number of the substring searched for within the string.

Missing Fig.

**Figure 10: Using indexOf method of String object**

**Solution:**

**Step 1:** Write the Code and save it as **Prob2.html**.

**Step 2:** Open **prob2.html** page in the browser, and verify that you get the same output as required.

### 3.3: Using various String methods

Write **prob3.html** page by completing the following code that demonstrates some of the methods of the String objects like **match**, **substr**, **lowerCase**, and **upperCase** to produce the output as shown in the figure given below:

Missing Fig.

**Figure 11: Using various String methods**

**Solution:**

**Step 1:** Write the Code and save it as **Prob3.html**.

**Step 2:** Open **prob3.html** page in the browser, and verify that you get the same output as required.

**Lab 4:Working with Arrays**

<b>Goals</b>	Work with Array Object
<b>Time</b>	10 minutes

**4 .1: Using Array to display values**

Create a **prob1.html** web page containing script. In this script, declare an array of 6 employee names and display it in the browser as shown below:



**Figure 12: Using Array to display values**

**Solution:**

**Step 1:** Write the Code, and save it as **Prob1.html**.

**Step 2:** Open **prob1.html** page in the browser, and verify that you get the same output as required.



**Lab 5: Working with Document Object**

<b>Goals</b>	Understand Document Object
<b>Time</b>	120 minutes

**6.1: Working with Documents**

Create a **prob1.html** web page which displays products available as shown in the following figure. The product details comprise Product Name, Product description, and its price.

Name	Description	Price	Qty
Barbie Doll	Beautiful	20	<input type="text"/>
Calculator	Calculator with latest features	30	<input type="text"/>
Mobile Phone	Camera, Java Games, GPRS	40	<input type="text"/>
LG DVD	5 disc changer	50	<input type="text"/>

**Figure 13: Displaying Products**

Users can place orders specifying the quantity of each product. If the user does not enter quantity in any of the text fields, then an error message should be displayed as shown in the figure given below:

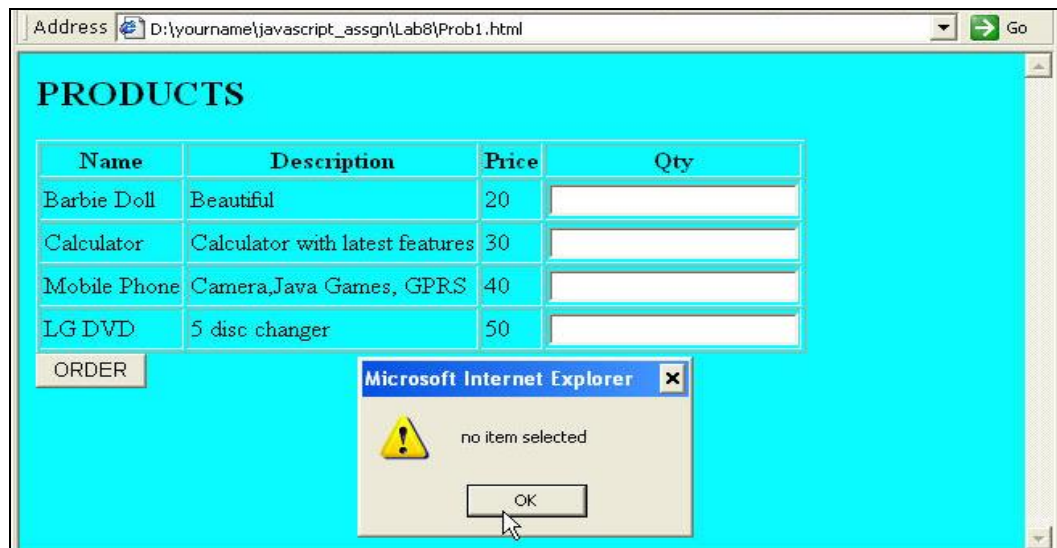


Figure 14: Validating Products

When the user clicks the **Order** button, the invoice for the current products transaction showing the product name, quantity ordered, price and total amount is displayed in a new window as shown in the figure given below:

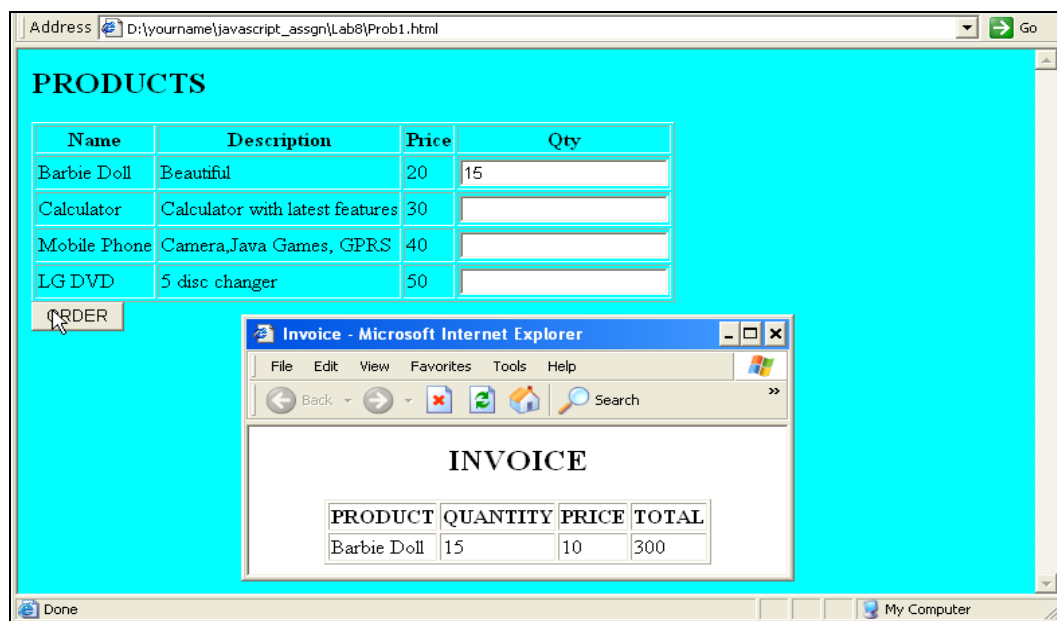


Figure 15: Displaying Invoice details in a new window

**Solution:**

**Step 1:** Write the code and save it as **Prob1.html**.

**Step 2:** Open **prob1.html** page in the browser, and verify that you get the same output as required.

## Lab 6: Working with Form Object

<b>Goals</b>	<ul style="list-style-type: none"> <li>Understand and use Form Object.</li> </ul>
<b>Time</b>	90 minutes

### 8.1: Form Validation

Create a **prob1.html** web page, as shown below, and calculate **Payment Information** based on **Loan Information**. Validate **Loan information** textfields for numbers. The **Payment Information** textfields should be uneditable. The other constraints are as follows:

- Amount of Loan should not be more than 15 lakhs.
- Repayment period should be between 7 yrs to 15 yrs.

**Computing Payment and Loan Information - Microsoft Internet Explorer**

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Media

Address  Go

**Enter Loan Information:**

1) Amount of the loan (any currency):

2) Annual percentage rate of interest:

3) Repayment period in years:

**Payment Information:**

4) Your monthly payment will be:

5) Your total payment will be:

6) Your total interest payments will be:

Done My Computer

Figure 16: Validating Form elements

If the repayment period is not between 7 and 15, then an error message should be displayed next to this control.

Similar kind of error message should be displayed if the amount of loan exceeds 15 lakh.

```
In the function calculatePayment()
/*
TODO:
calculate the monthly payment, total payment, total interest payment on click of the button with
label "compute"
*/
```

## Example 8: Lab 8: Prob1.html

Open **prob1.html** page in the browser, and verify that you get the same output as required.

### 8.2 Validate Field

Create a **prob2.html** page as shown in the below figure.

The image shows a web form titled "Product Details" with a blue background. The form contains the following elements:

- Category:** A dropdown menu with "Electronics" selected.
- Product:** A dropdown menu with "-----" selected.
- Quantity:** A text input field.
- Total Price:** A text input field.
- Submit:** A button.
- Clear:** A button.

**Figure 20: Lab 8.2 Product Details**

Data should be prepopulated in category list box (Electronics, Grocery). Based on selection of category, product list need to be populated automatically with values as given in the below table. Also Total price need to be calculated for the entered quantity as per the data in the below table. Total price field should be non-editable field.

Category	Product	Price per quantity in Rupees
Electronics	Television	20000
	Laptop	30000
	Phone	10000
Grocery	Soap	40
	Powder	90

While clicking on submit button, if all the text fields contains valid values then display the filled details in a popup window.