

Lesson 04 Demo 05

Implementing Functions and Prototyping

Objective: To implement and demonstrate the use of JavaScript functions and prototypes for creating employee objects and calculating their ages

Tools required: Visual Studio Code

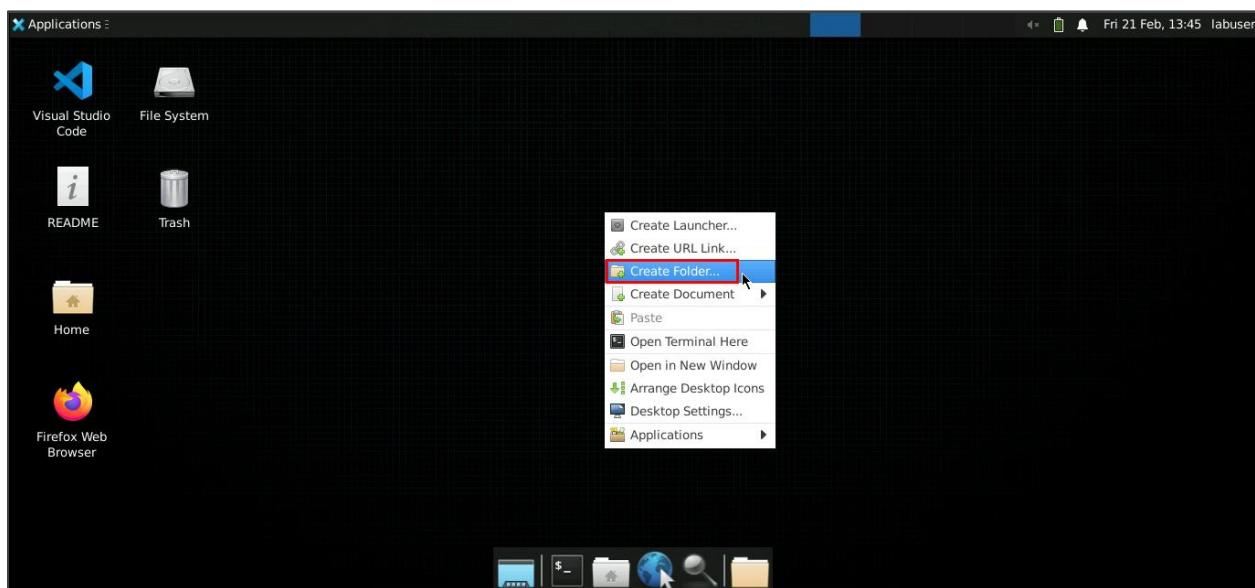
Prerequisites: A basic understanding of functions and prototyping in JavaScript

Steps to be followed:

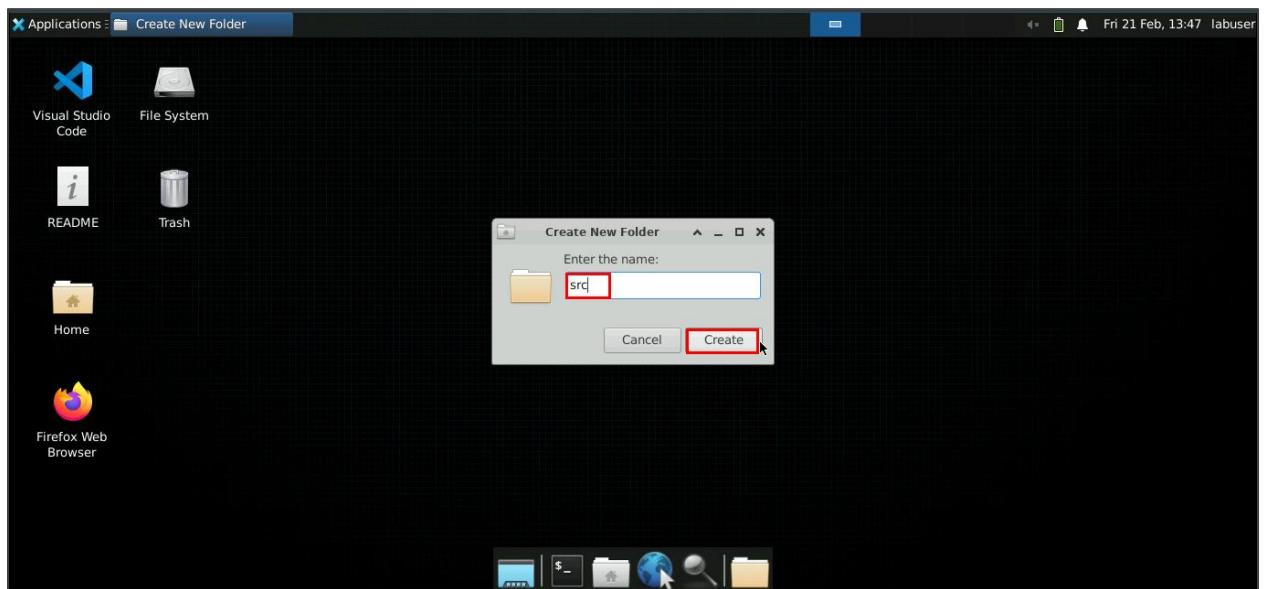
1. Create an index.html file and execute it

Step 1: Create an index.html file and execute it

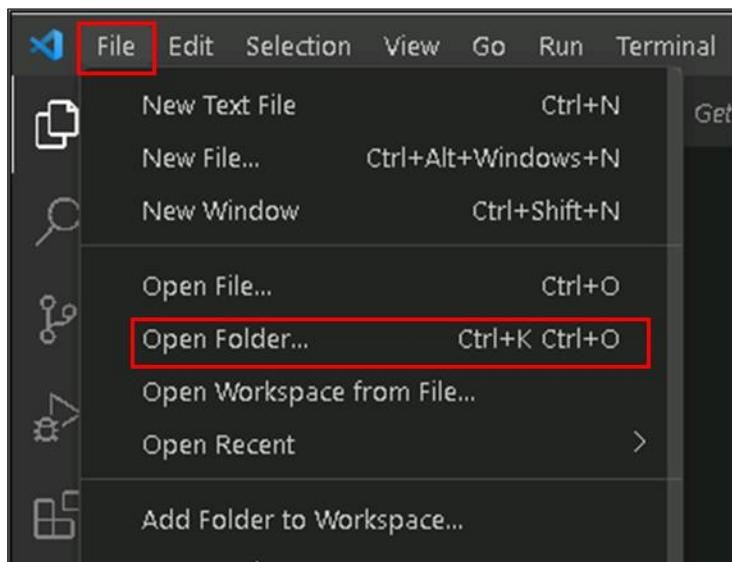
1.1 Right-click on the desktop and select **Create Folder...**



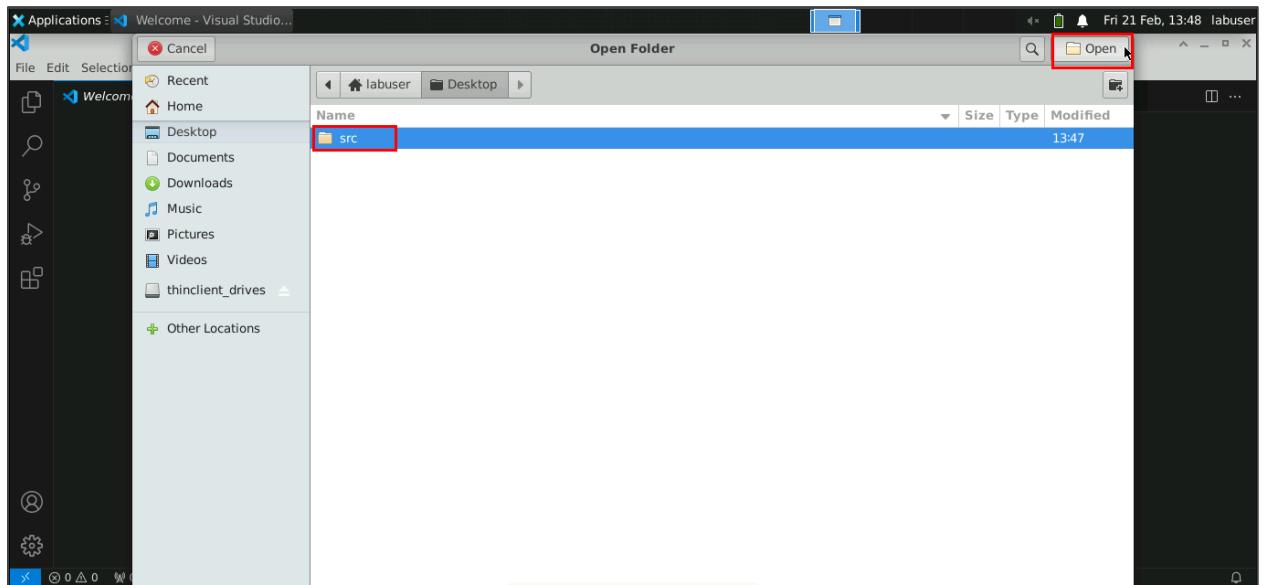
1.2 Enter the folder name as **src** and click on **Create**



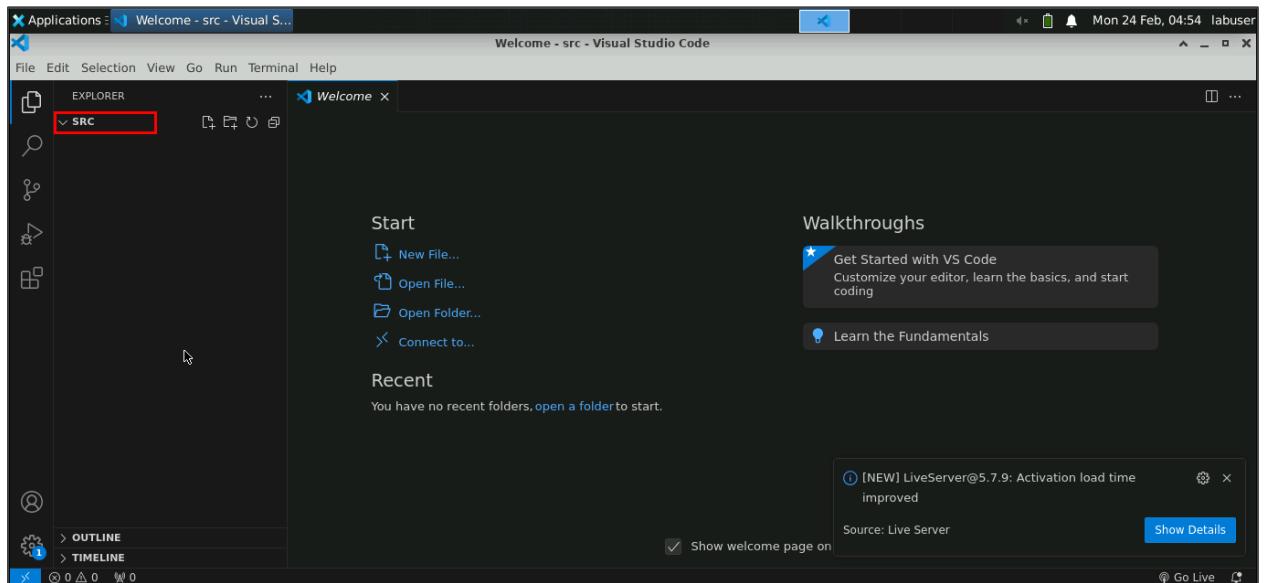
1.3 Open Visual Studio Code, right-click on **File**, and select the **Open Folder...** option



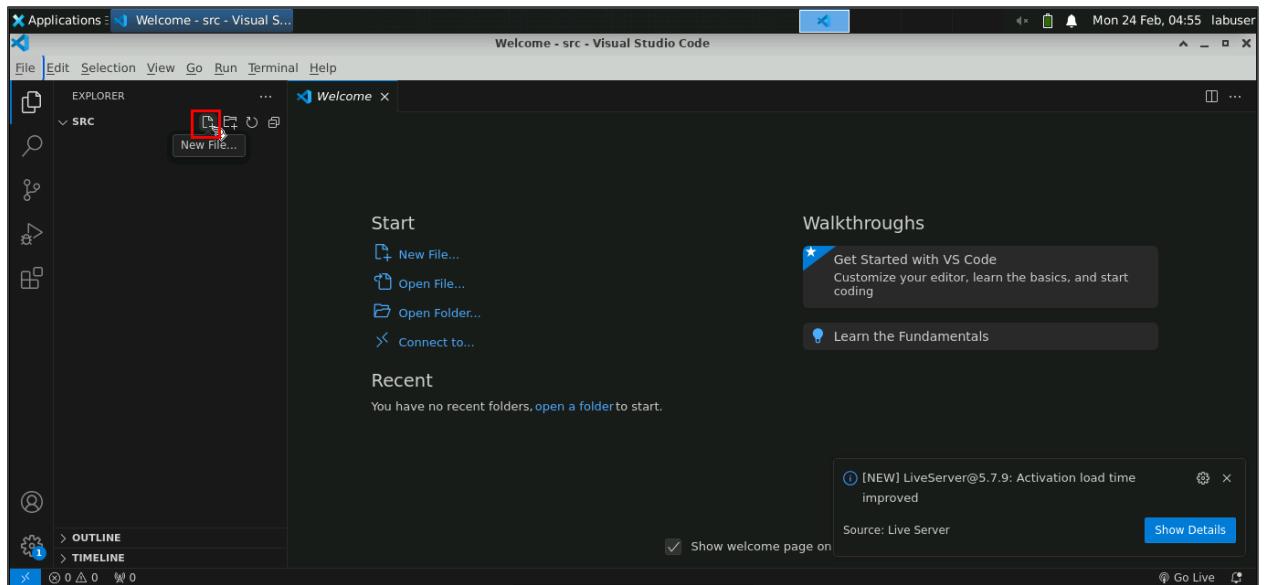
1.4 Select the **src** folder and click on **Open**



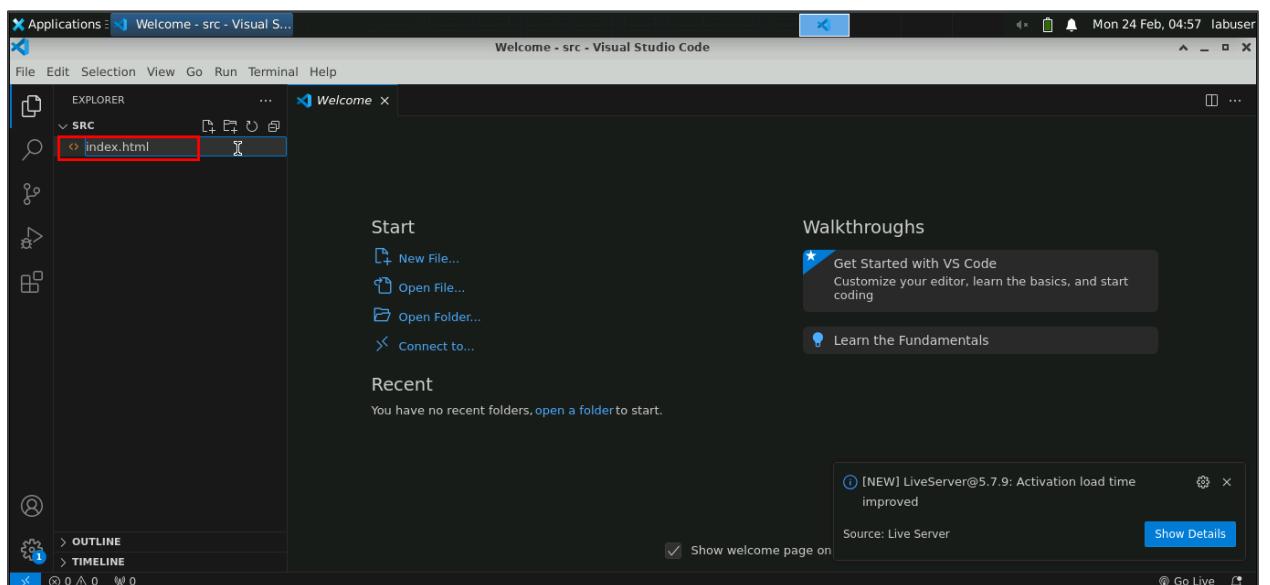
The SRC folder opens in Visual Studio Code as shown below:



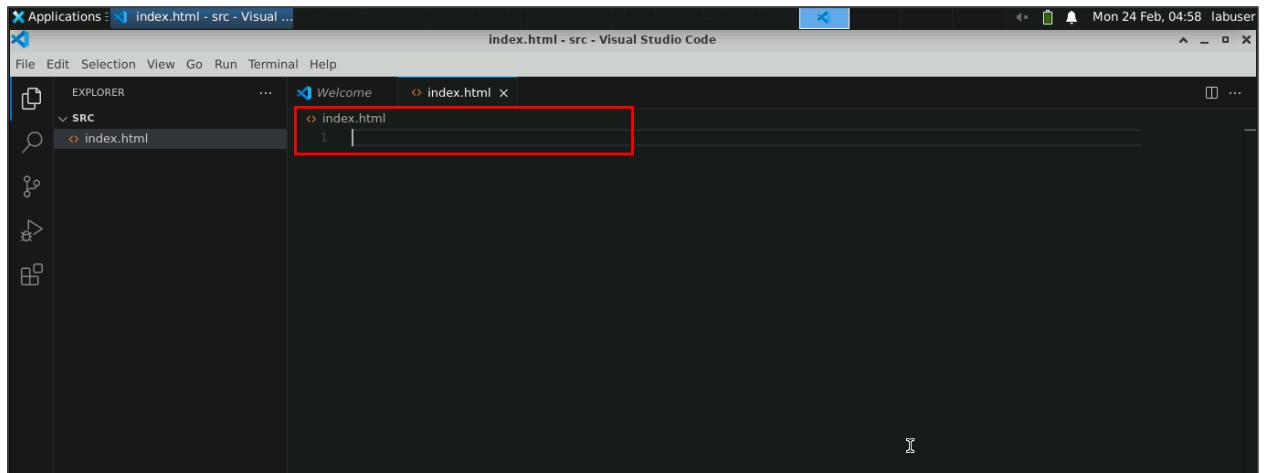
1.5 Select the SRC folder of the project and click on the **New File** icon



1.6 Enter the filename as **index.html** and press **enter**

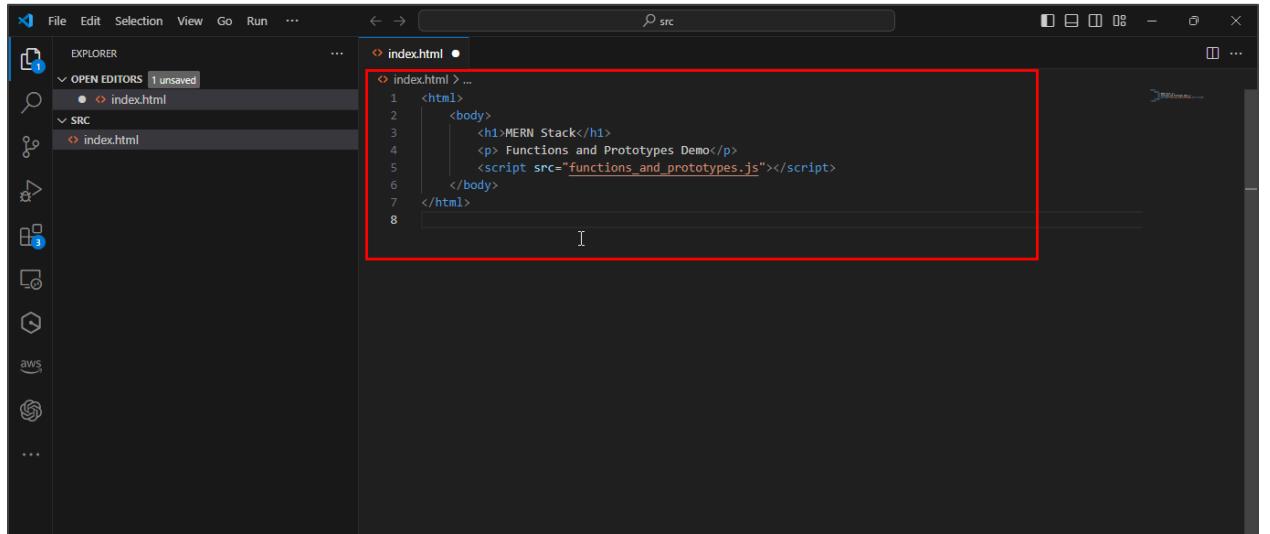


The index.html file gets created as shown below:

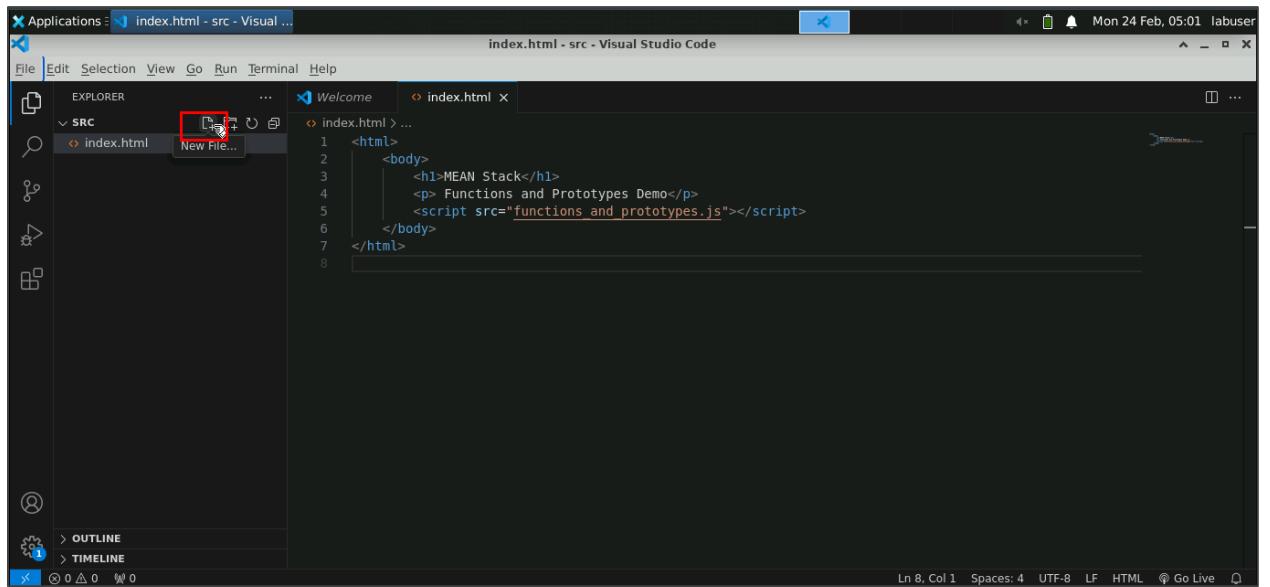


1.7 Add the following code to the index.html file:

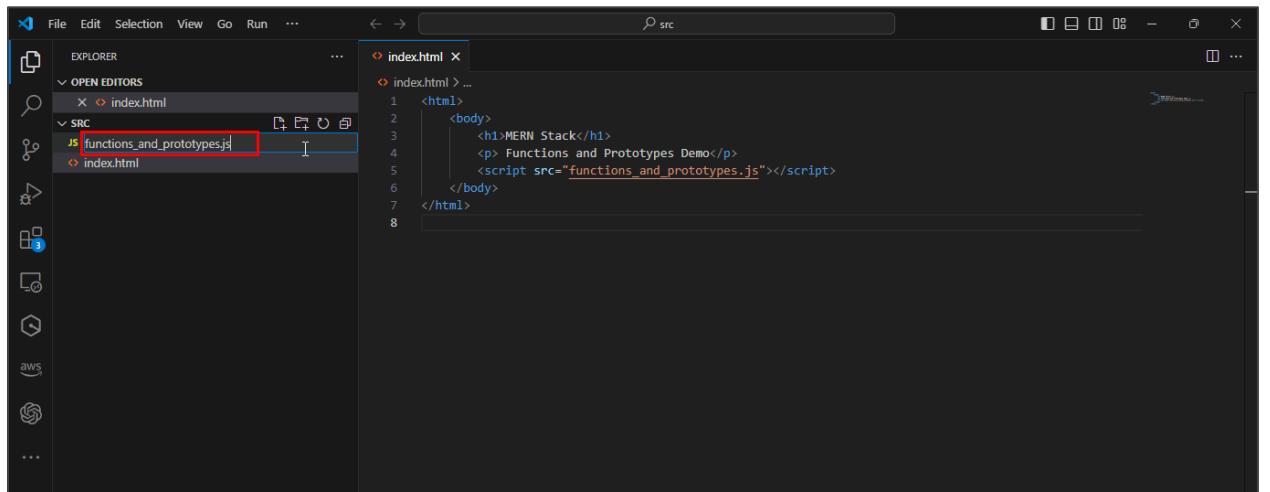
```
<html>
<body>
  <h1>MERN Stack</h1>
  <p> Functions and Prototypes Demo</p>
  <script src="functions_and_prototypes.js"></script>
</body>
</html>
```



1.8 Right-click on the SRC folder of the project and select the New File icon



1.9 Name the file **functions_and_prototypes.js** and press enter



1.10 Enter the following code into the functions_and_prototypes.js file:

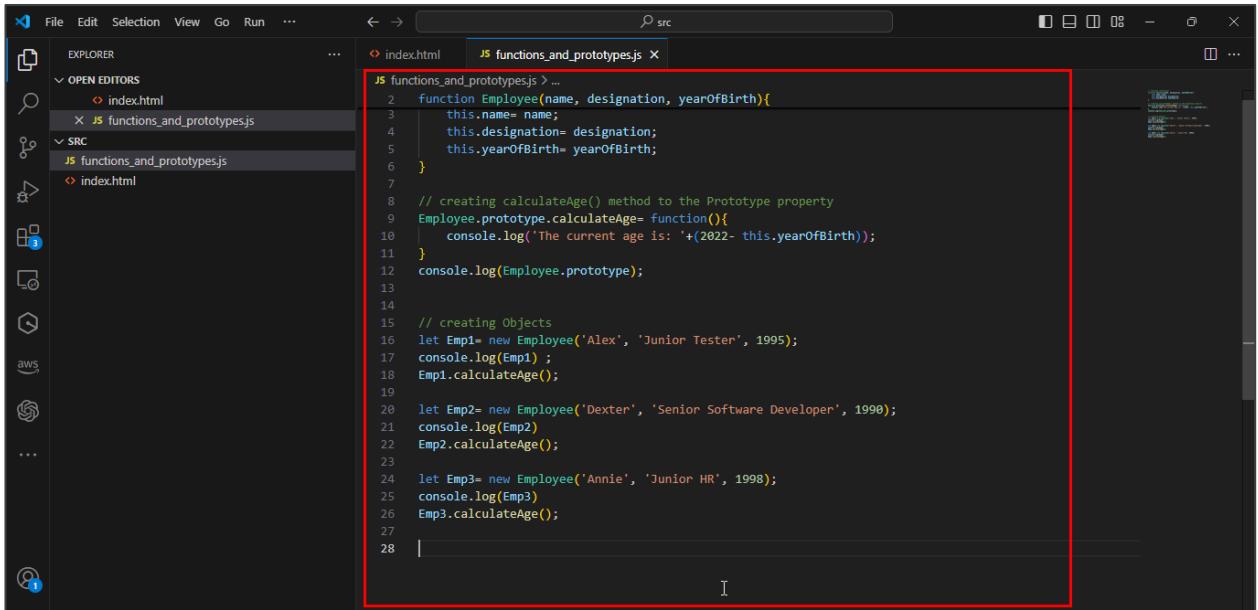
```
// function constructor
function Employee(name, designation, yearOfBirth){
    this.name= name;
    this.designation= designation;
    this.yearOfBirth= yearOfBirth;
}

// creating calculateAge() method to the Prototype property
Employee.prototype.calculateAge= function(){
    console.log('The current age is: '+ (2022- this.yearOfBirth));
}
console.log(Employee.prototype);

// creating Objects
let Emp1= new Employee('Alex', 'Junior Tester', 1995);
console.log(Emp1) ;
Emp1.calculateAge();

let Emp2= new Employee('Dexter', 'Senior Software Developer', 1990);
console.log(Emp2)
Emp2.calculateAge();

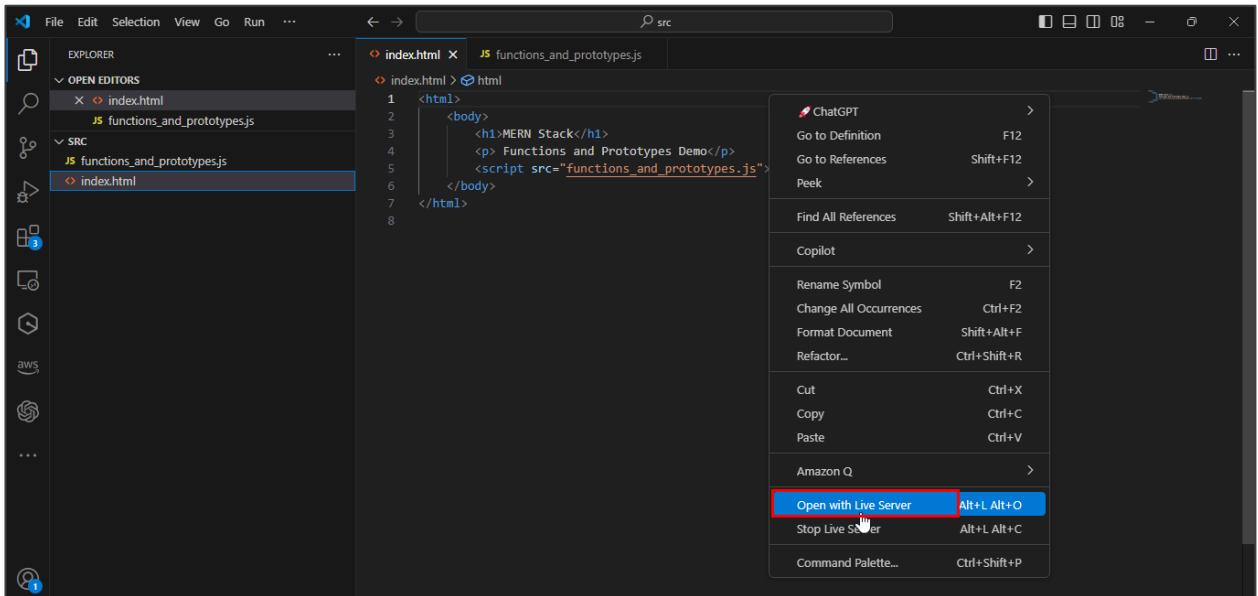
let Emp3= new Employee('Annie', 'Junior HR', 1998);
console.log(Emp3)
Emp3.calculateAge();
```



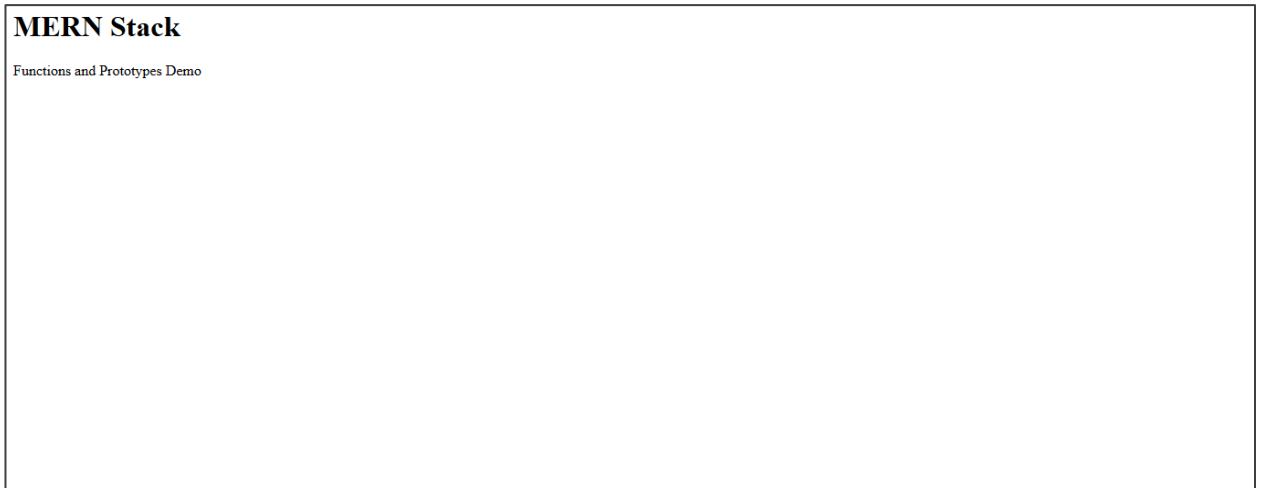
```
function Employee(name, designation, yearOfBirth){  
    this.name= name;  
    this.designation= designation;  
    this.yearOfBirth= yearOfBirth;  
}  
  
Employee.prototype.calculateAge= function(){  
    console.log('The current age is: '+(2022- this.yearOfBirth));  
}  
  
let Emp1= new Employee('Alex', 'Junior Tester', 1995);  
console.log(Emp1);  
Emp1.calculateAge();  
  
let Emp2= new Employee('Dexter', 'Senior Software Developer', 1990);  
console.log(Emp2);  
Emp2.calculateAge();  
  
let Emp3= new Employee('Annie', 'Junior HR', 1998);  
console.log(Emp3);  
Emp3.calculateAge();
```

Note: Press **Ctrl + S** to save the file

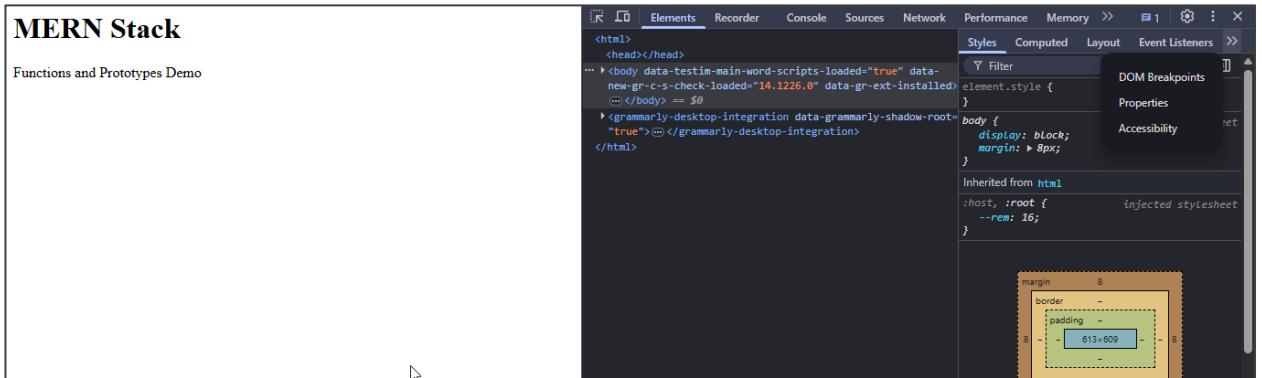
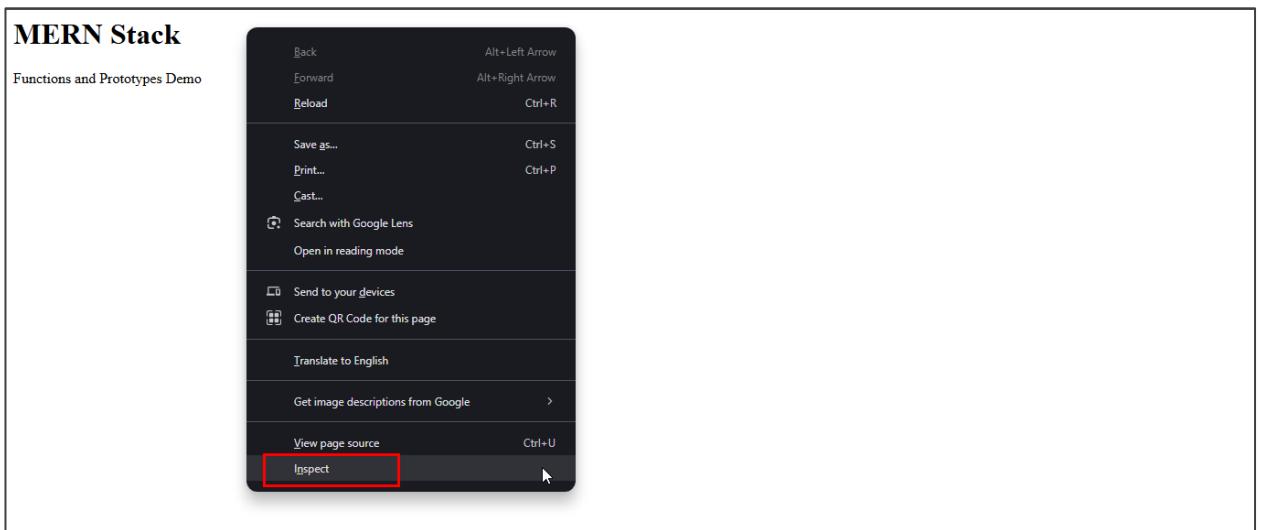
1.11 Right-click on the index.html file of the project and select the **Open with Live Server** option



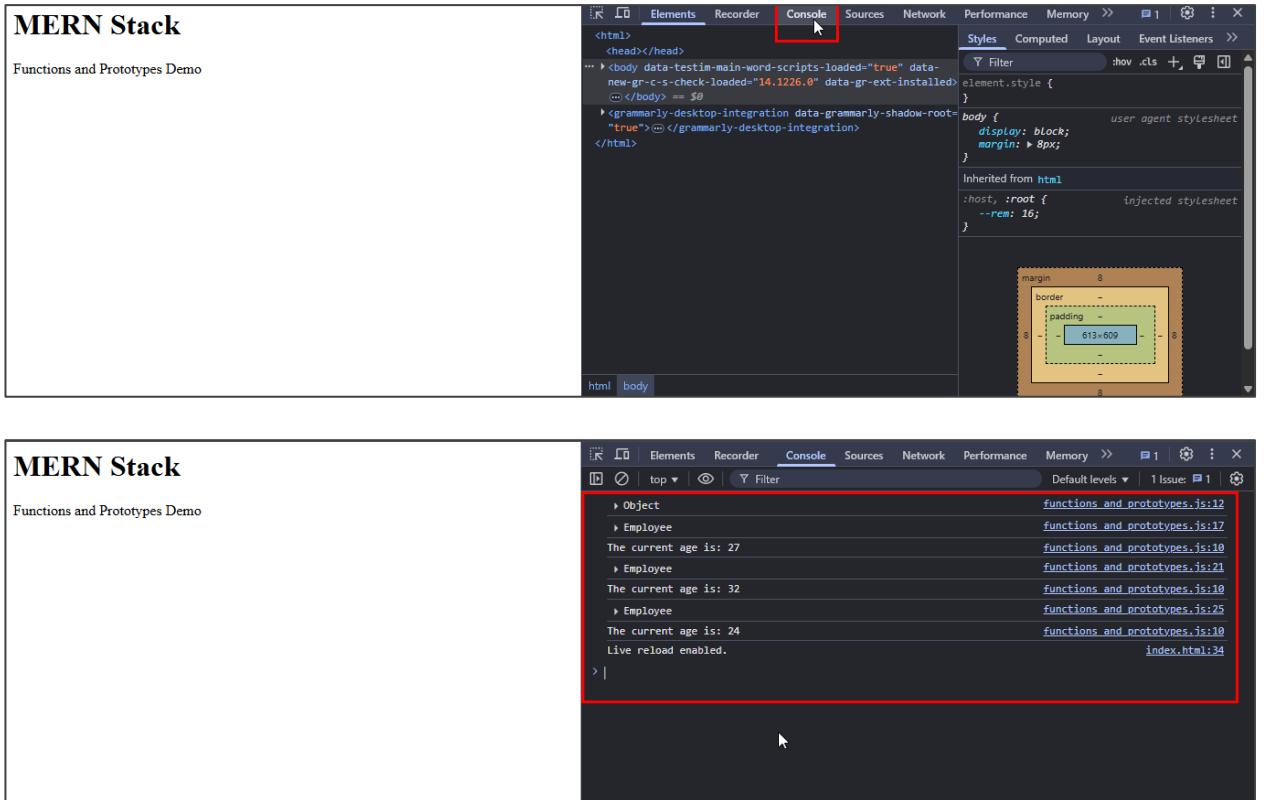
The output is as shown below:



1.12 Once the server starts running, right-click and select the **Inspect** option



1.13 Click on **Console** tab



The provided code demonstrates the implementation of functions and prototypes. It involves creating objects using a function constructor, adding a method to the prototype property, and displaying the results in the browser console.

By following these steps, you have successfully implemented and verified the use of JavaScript functions and prototypes to create and manage employee objects and calculate their ages.