To create a JavaScript project and implement IIFEs (Immediately Invoked Function Expressions), callbacks, and closures, along with 2 employee details, you can follow these steps:

1. Open your IDE and create a new folder for your project. Name it something like "iifes-callbacks-closures-project".

2. Inside the project folder, create two files: "index.html" and "script.js".

3. Open "index.html" in your IDE and add the following basic HTML structure:

```html
<!DOCTYPE html>
<html>
<head>
   <title>IIFEs, Callbacks, and Closures</title>
   <script src="script1.js"></script>
</head>
<body>
   <h1>Employee Details</h1>
</body>
</html>
```

4. In the same project folder, open "script.js" in your IDE and write the JavaScript code to implement IIFEs, callbacks, closures, and employee details. Here's an example:

```javascript
// IIFE (Immediately Invoked Function Expression)
(function () {
   console.log('This is an IIFE');
})();

// Callback function example
function processEmployeeDetails(employee, callback) {
   console.log('Processing employee details:', employee);
   callback(employee);
}
function displayEmployeeDetails(employee) {
   console.log('Employee Name:', employee.empname);
   console.log('Age:', employee.age);
   console.log('City:', employee.city);
   console.log('Phone Number:', employee.phnNumber);
   console.log('Employee ID:', employee.empid);
   console.log('-----------------------');
}

// Employee details
var employee1 = {
   empname: 'John Doe',
   age: 30,
   city: 'New York',
   phnNumber: '123-456-7890',
   empid: 'EMP001'
};

var employee2 = {
   empname: 'Jane Smith',
   age: 28,
   city: 'San Francisco',
   phnNumber: '987-654-3210',
   empid: 'EMP002'
};
// Process employee details with callback
processEmployeeDetails(employee1, displayEmployeeDetails);
processEmployeeDetails(employee2, displayEmployeeDetails);
```

In the above code, we define an IIFE that is executed immediately when the script is loaded. It logs a message to the console.

We then define a `processEmployeeDetails` function that takes an employee object and a callback function as arguments. Inside `processEmployeeDetails`, we log the employee details being processed and invoke the callback function with the employee object. We provide `displayEmployeeDetails` as the callback, which logs the employee details (name, age, city, phone number, and employee ID) to the console.

We also define two employee objects, `employee1` and `employee2`, with their respective details.

Lastly, we call `processEmployeeDetails` twice, passing in the employee objects and the `displayEmployeeDetails` callback function to display the employee details.

5. Save both files.

6. Open "index.html" in a web browser, and you should see a blank page.

7. Open the browser's developer tools (usually accessible through right-clicking on the page and selecting "Inspect" or "Inspect Element").

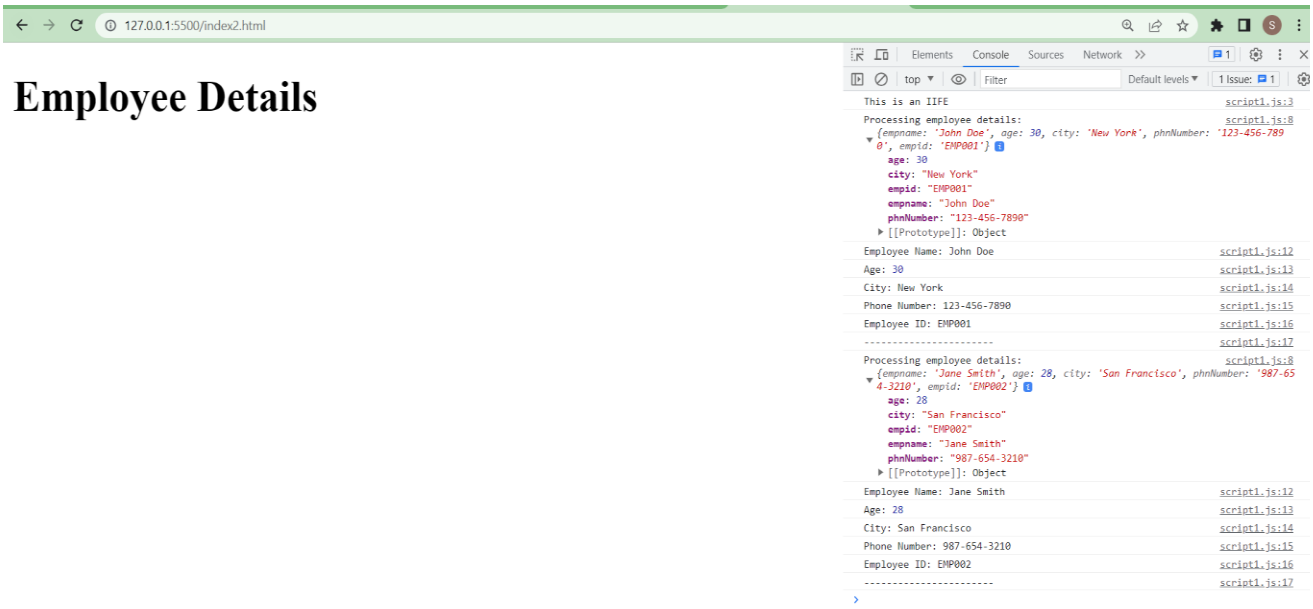
8. Switch to the "Console" tab in the developer tools.

9. Reload the "index.html" page.

10. In the console, you should see the output of the IIFE, as well as the employee details for both employees.

11. To push the code to your GitHub repository, follow the same steps as mentioned in subsection 2.3.3.

Now you have a JavaScript project with examples of IIFEs, callbacks, closures, and employee details, and their output is printed in the browser's console.



Employee Details

```
This is an IIFE                                                                script1.js:3
Processing employee details:                                                   script1.js:8
▼ {empname: 'John Doe', age: 30, city: 'New York', phnNumber: '123-456-789
  0', empid: 'EMP001'}
    age: 30
    city: "New York"
    empid: "EMP001"
    empname: "John Doe"
    phnNumber: "123-456-7890"
  ▶ [[Prototype]]: Object
Employee Name: John Doe                                                       script1.js:12
Age: 30                                                                       script1.js:13
City: New York                                                                script1.js:14
Phone Number: 123-456-7890                                                    script1.js:15
Employee ID: EMP001                                                           script1.js:16
----------------------                                                        script1.js:17
Processing employee details:                                                   script1.js:8
▼ {empname: 'Jane Smith', age: 28, city: 'San Francisco', phnNumber: '987-65
  4-3210', empid: 'EMP002'}
    age: 28
    city: "San Francisco"
    empid: "EMP002"
    empname: "Jane Smith"
    phnNumber: "987-654-3210"
  ▶ [[Prototype]]: Object
Employee Name: Jane Smith                                                     script1.js:12
Age: 28                                                                       script1.js:13
City: San Francisco                                                           script1.js:14
Phone Number: 987-654-3210                                                    script1.js:15
Employee ID: EMP002                                                           script1.js:16
----------------------                                                        script1.js:17
>
```