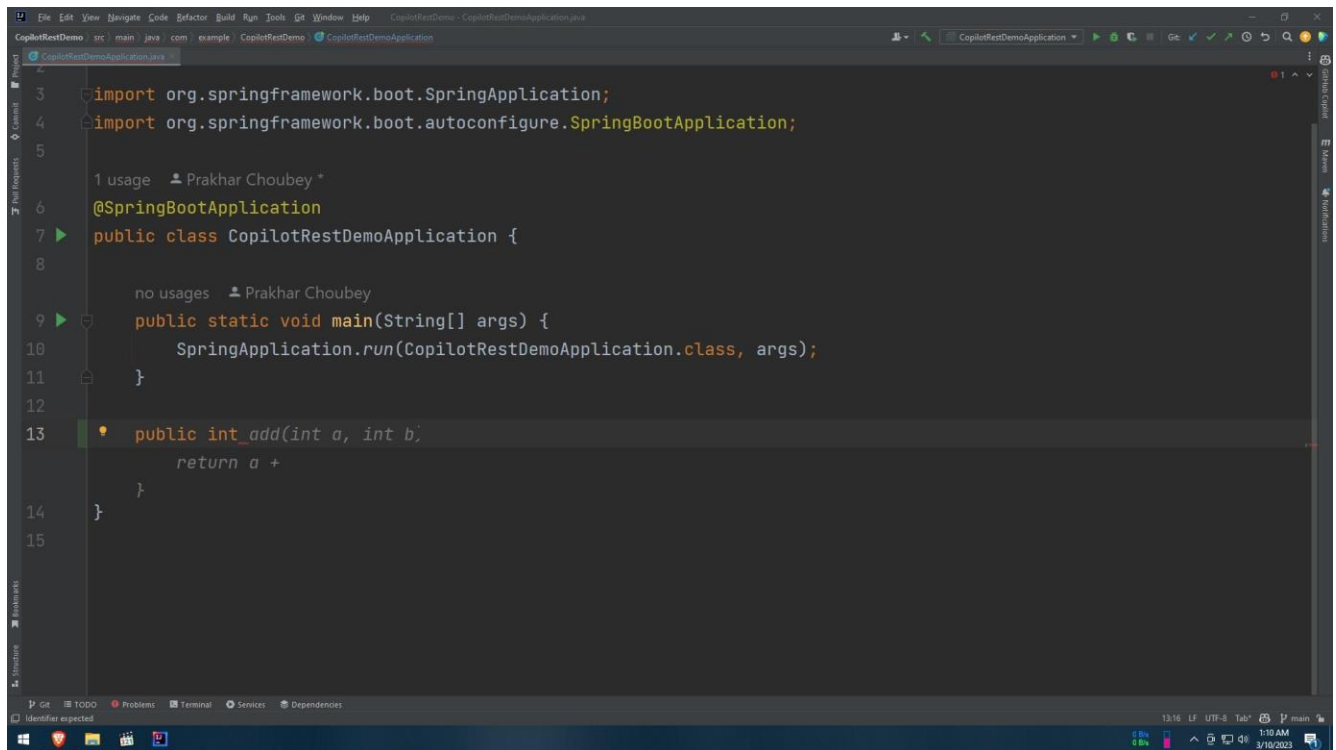


## GitHub Copilot use cases: -

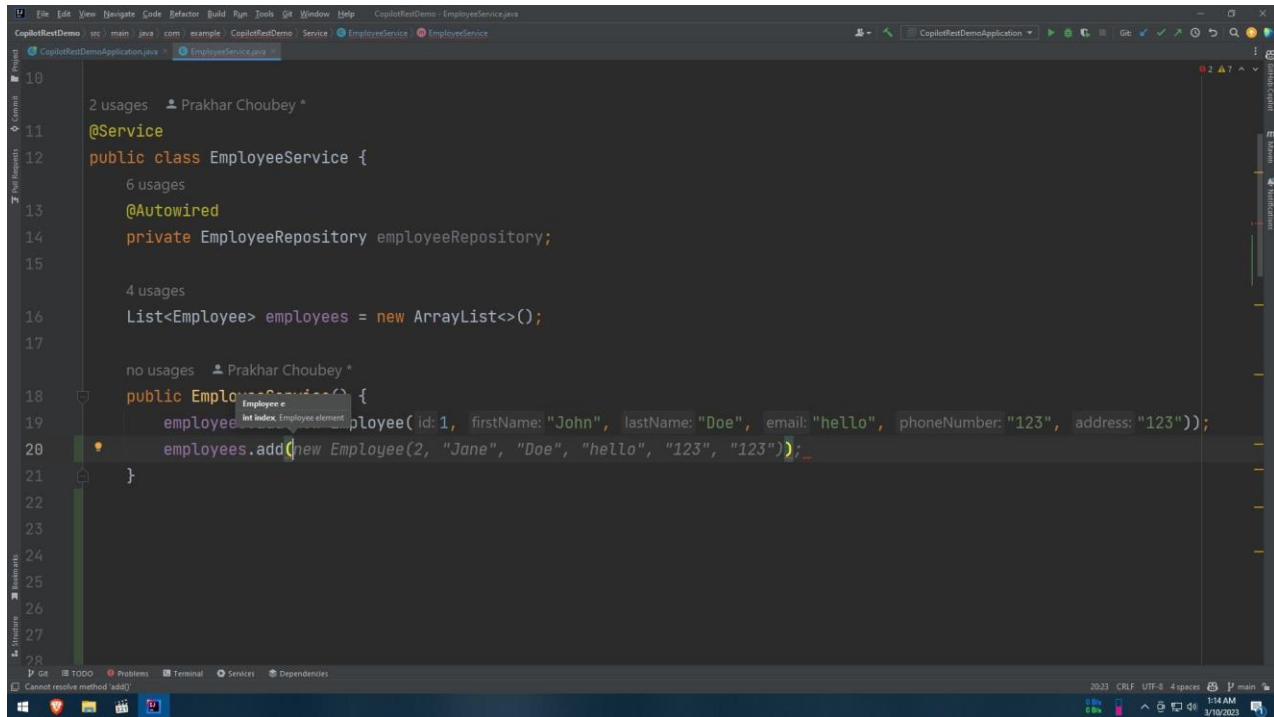
### 1. Auto Completing codes



### 2. Suggests dummy data for objects and different data structures.

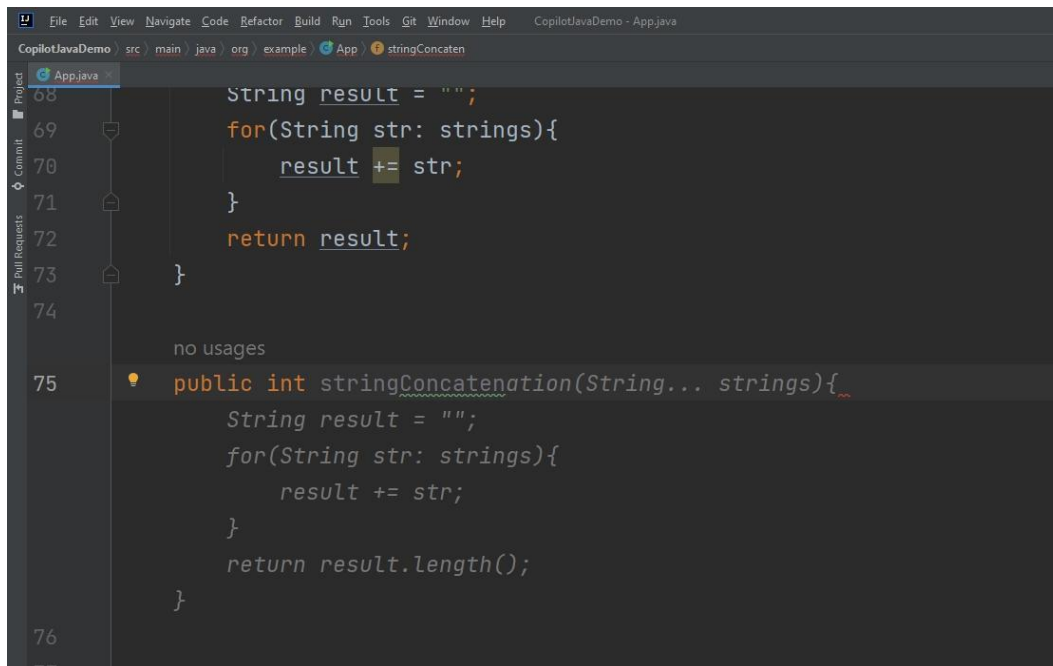


### 3. Suggest using our past-used codes.



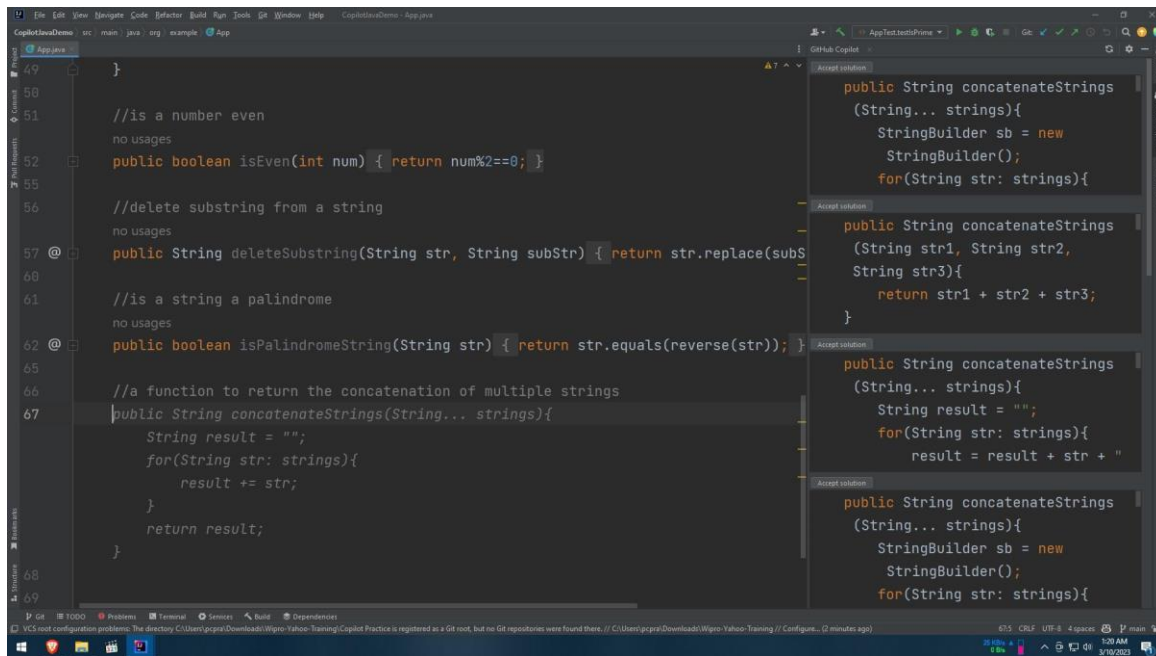
```
10 2 usages  Prakhar Choubey *
11 @Service
12 public class EmployeeService {
13     6 usages
14     @Autowired
15     private EmployeeRepository employeeRepository;
16
17     4 usages
18     List<Employee> employees = new ArrayList<>();
19
20     no usages  Prakhar Choubey *
21     public EmployeeService() {
22         employeeRepository.save(new Employee(1, "John", "Doe", "hello", "123", "123"));
23     }
24     employees.add(new Employee(2, "Jane", "Doe", "hello", "123", "123"));
25 }
26
27
28
```

### 4. Generates return statement, by analyzing the prototype of the method.



```
68 String result = "";
69 for(String str: strings){
70     result += str;
71 }
72 return result;
73 }
74
75 no usages
76 public int stringConcatenation(String... strings){
77     String result = "";
78     for(String str: strings){
79         result += str;
80     }
81     return result.length();
82 }
83
```

## 5. Generates full functions, just by writing a brief comment



The screenshot shows an IDE with a Java file named `App.java`. The code includes several comments and a few implemented methods. Copilot has generated three full methods for the `concatenateStrings` function based on the comments. The comments are:

- `//is a number even`
- `//delete substring from a string`
- `//is a string a palindrome`
- `//a function to return the concatenation of multiple strings`

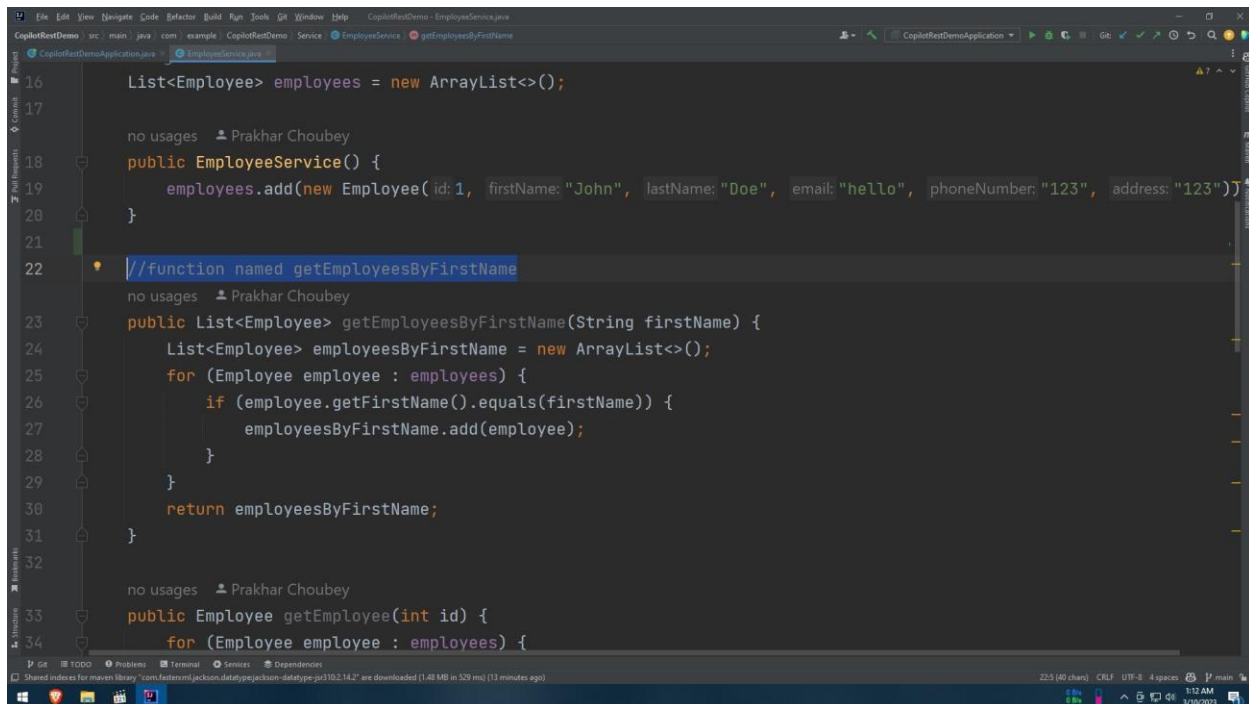
The generated methods are:

```
public boolean isEven(int num) { return num%2==0; }

public String deleteSubString(String str, String subStr) { return str.replace(subStr, ""); }

public boolean isPalindromeString(String str) { return str.equals(reverse(str)); }

public String concatenateStrings(String... strings){
    String result = "";
    for(String str: strings){
        result = result + str + " ";
    }
    return result;
}
```



The screenshot shows an IDE with a Java file named `EmployeeService.java`. The code includes a list of employees and a method to get employees by first name. The comments and code are:

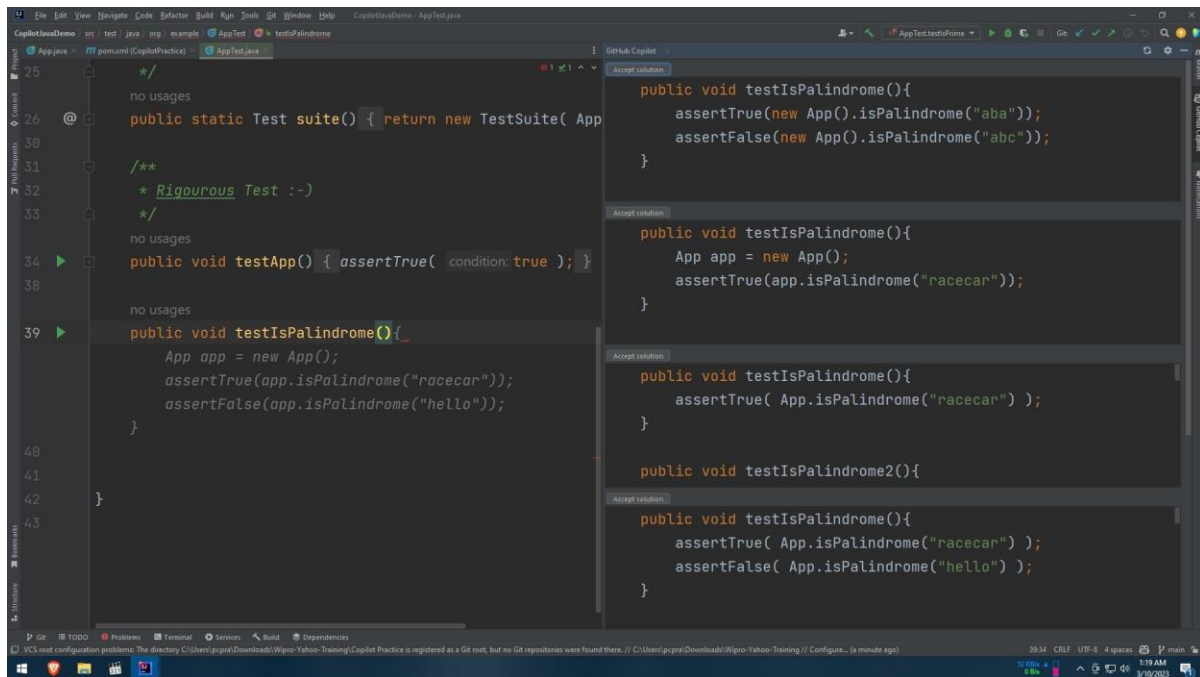
```
List<Employee> employees = new ArrayList<>();

no usages  Prakhar Choubey
public EmployeeService() {
    employees.add(new Employee( id: 1,  firstName: "John",  lastName: "Doe",  email: "hello",  phoneNumber: "123",  address: "123"));
}

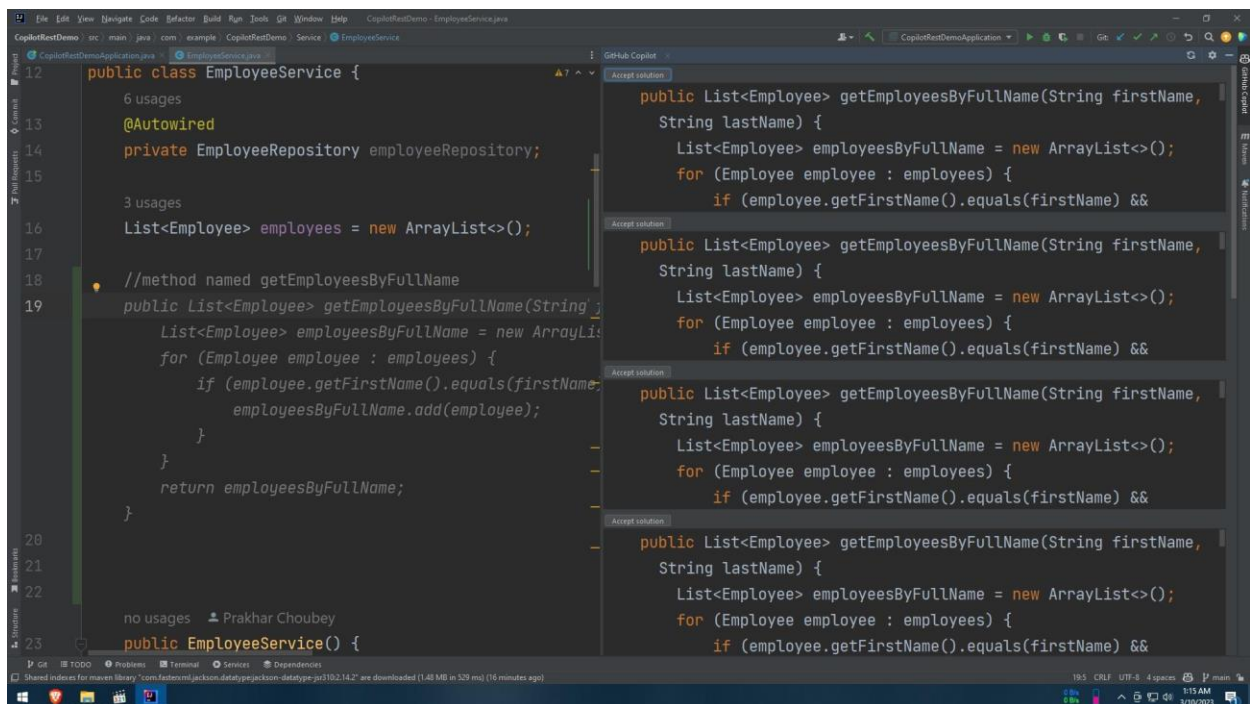
//Function named getEmployeesByFirstName
no usages  Prakhar Choubey
public List<Employee> getEmployeesByFirstName(String firstName) {
    List<Employee> employeesByFirstName = new ArrayList<>();
    for (Employee employee : employees) {
        if (employee.getFirstName().equals(firstName)) {
            employeesByFirstName.add(employee);
        }
    }
    return employeesByFirstName;
}

no usages  Prakhar Choubey
public Employee getEmployee(int id) {
    for (Employee employee : employees) {
```

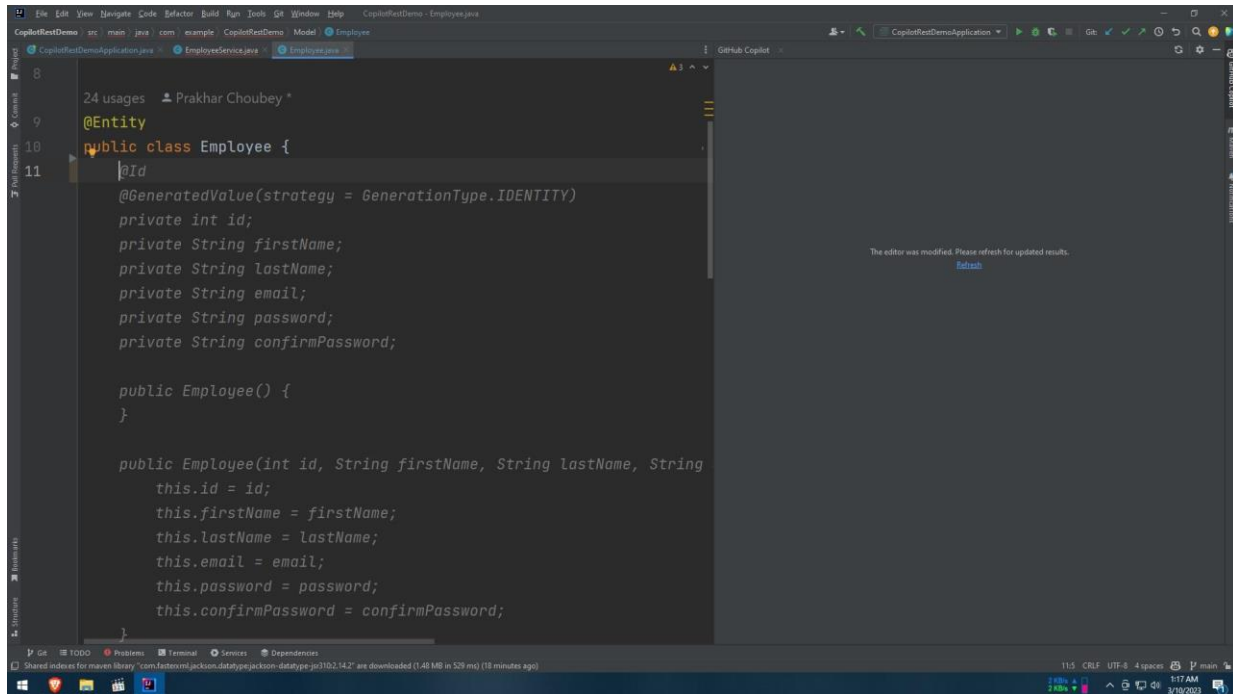
## 6. Generates test cases



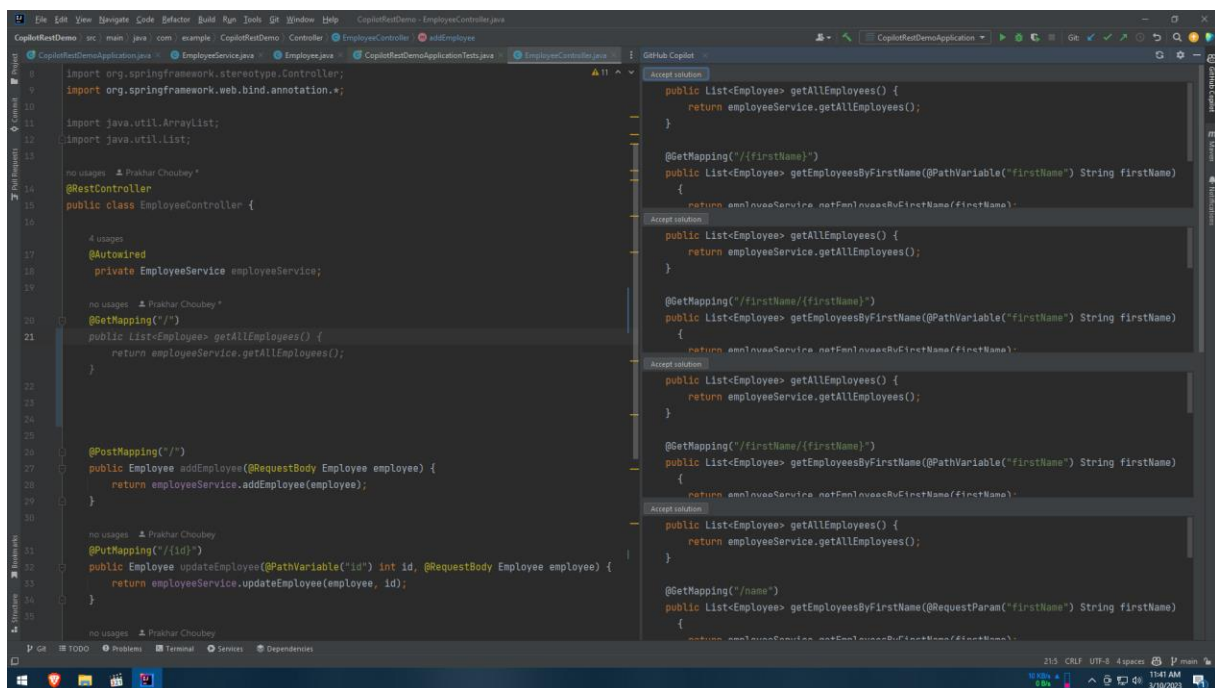
## 7. Allows us to choose from different suggestions.

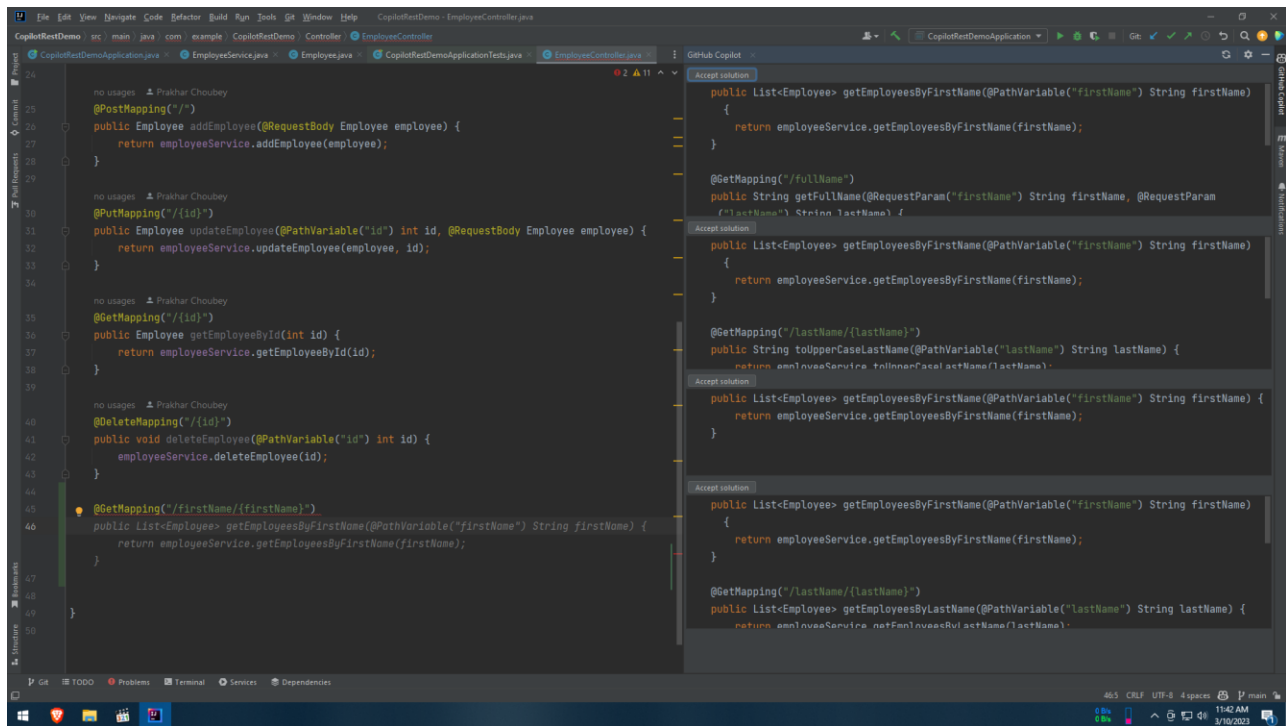


8. Shows suggestions of methods and data members according to the class name.



9. Auto Completion in Controller class





Problem Faced: -

1. The import statement needed to be added manually.
2. Annotations suggestions are not always correct.