# Junit, Mockito, Logging Assignment

## Prabhakar

### Tasks:
- Use Spring MVC or any Spring Boot application, add more validations to it.

- Write Junit test cases for each and every use case with 100% coverage

- Coverage screenshot has to be attached along with assignment

- Each test case should contain at least one assert statement

- No real time calls should happen, all calls should be completely mocked

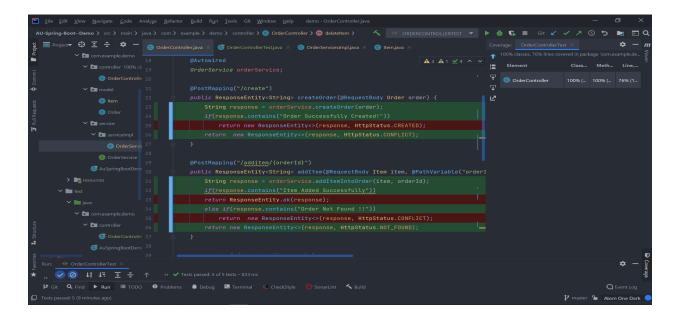- Add appropriate logs using Logger wherever applicable, possibly use all levels of logging

### Logging:



### Testing:
- Code coverage without 100% coverage

-   **Junit test case passes:**



-   **Code coverage with 100% line coverage**

```java
        return ResponseEntity.ok(itemList);
    }


    @DeleteMapping("/deleteitem/{orderId}/{itemId}")
    public ResponseEntity<String> deleteItem(@PathVariable("orderId") int orderId, @PathVariable("itemId") int itemId){
        String response = orderService.deleteItemFromOrder(itemId ,orderId);

        if(response.contains("Item deleted!!"))
            return ResponseEntity.ok(response);
        return new ResponseEntity<>(response, HttpStatus.NOT_FOUND);
    }


    @PutMapping("/updateitem/{orderId}/{itemId}")
    public ResponseEntity<Item> updateItem(@RequestBody Item item, @PathVariable("orderId") int orderId, @PathVariable("itemId") int

        Item item1 = orderService.updateItemIntoOrder(item, orderId, itemId);
        if(item1 != null)
            return ResponseEntity.ok(item1);
        return new ResponseEntity<Item>(item1, HttpStatus.NOT_FOUND);
    }
}
```