# Assignment 1

## 1.

## 2.

## 3.

Suggestions for such systems

1. The absence of the item stock feature so that users cannot know the number of available goods
2. In each product sold can only be purchased a number of 1 item, it cannot be more
3. There is no menu for historical and tracing of purchased items
4. At checkout, there is no payment method selection feature and cannot input the delivery/recipient location

## 4.

Bugs Report

1. When logging in using the standard\_user there is no validation during the items checkout because even though the user does not add the items to the cart the system still continues the checkout process.
2. When logging in using problem\_user every click on the icon / picture items will direct into a different product description, and this also happens when we have added products in the cart and at checkout cannot fill in personal data in the last name field so that it results in the product cannot be checked out as well as the filter on the homepage does not work.
3. When logging in using performance\_glitch\_user each time doing a certain event the time interval needed is quite time consuming such as when logging in adding goods to the cart and others

# Assignment 2

# Assignment 3

Run this Code on your IDE

## Code

import java.util.Arrays;

public class Assigment3 {

static void Case1(Integer arr[], int n) {

int l = 0, r = n - 1, k = 0;

while (l < r) {

while (arr[l] % 2 != 0) {

l++;

k++;

while (arr[r] % 2 == 0)

r--;

if (l < r) {

int temp = arr[l];

arr[l] = arr[r];

arr[r] = temp;

}

Arrays.sort(arr, 0, k);

}

}

}

public static void main(String[] args) {

Integer arr[] = {3, 2, 5, 1, 8, 9, 6};

System.out.println("Input ");

System.out.print(Arrays.toString(arr));

System.out.println("\nOutput ");

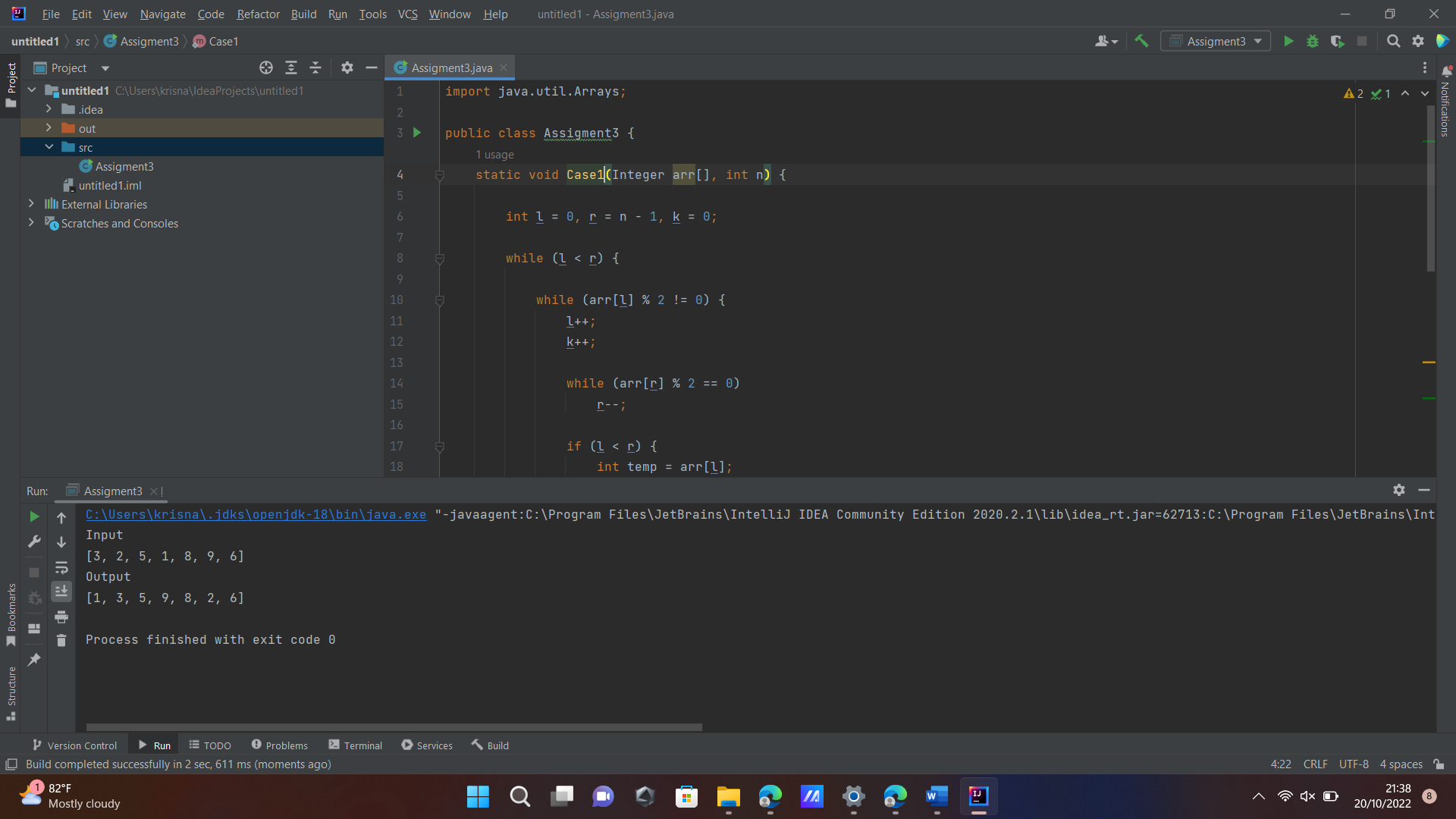
Case1(arr, arr.length);

System.out.println(Arrays.toString(arr));

}

}

## Output



# Assignment 4

Please Upload This Attachment SQL to Localhost

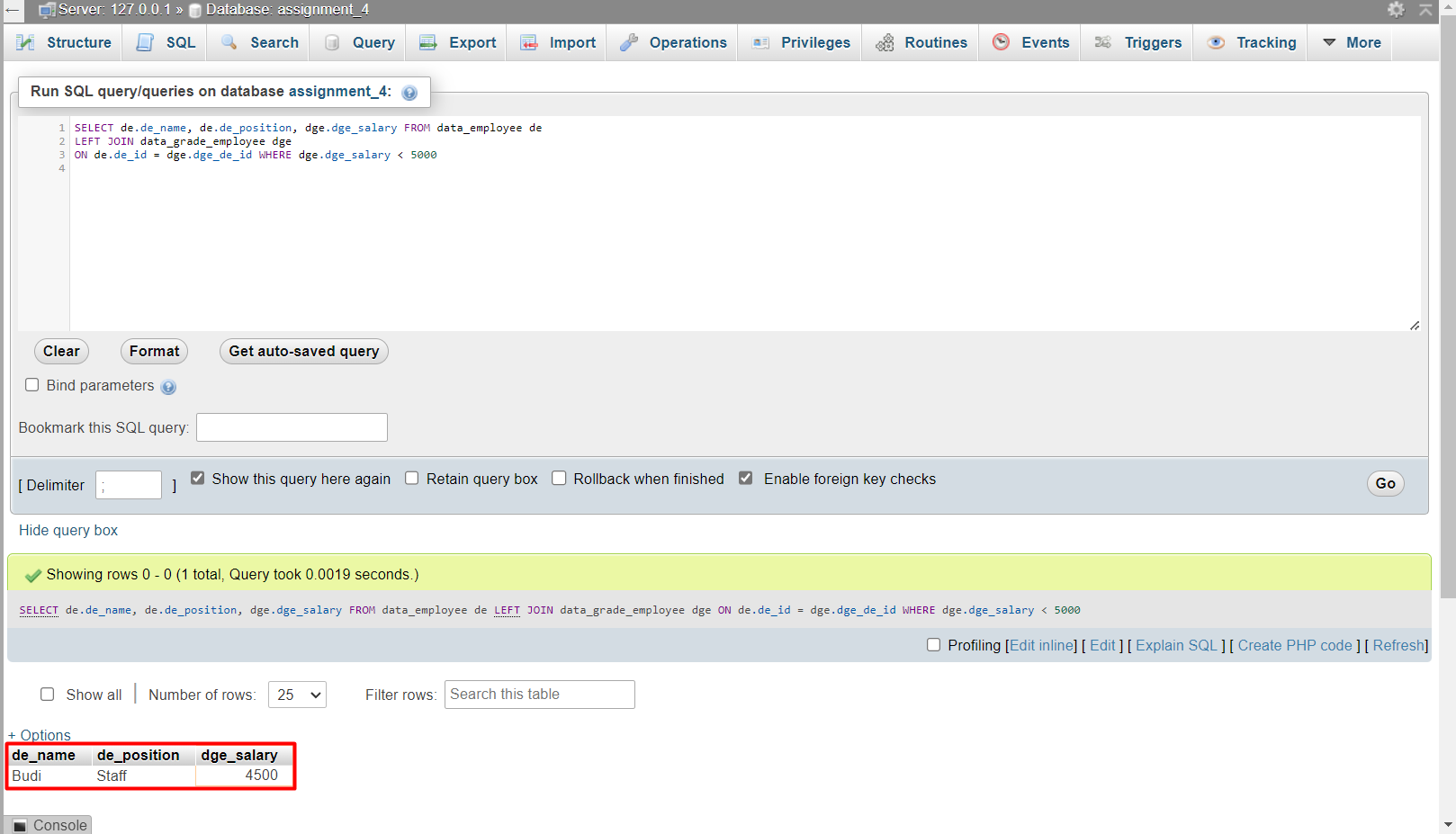


Then Run Query Below

## 1.

SELECT de.de\_name, de.de\_position, dge.dge\_salary FROM data\_employee de

LEFT JOIN data\_grade\_employee dge

ON de.de\_id = dge.dge\_de\_id WHERE dge.dge\_salary < 5000

## 2.

SELECT de.de\_position AS position,

COUNT(de.de\_position) AS total\_employee,

SUM(dge.dge\_salary) AS total\_salary

FROM data\_employee de, data\_grade\_employee dge

WHERE dge.dge\_de\_id= de.de\_id

GROUP BY de.de\_position

ORDER BY total\_salary ASC;

