Problem :

Here is the assignment:

Gordon Ramsey, a very smart guy, like seating food. Now, Gordon is at a restaurant and he has many different types of food to choose from. Gordon gets x amount of satisfaction and requires y amount of time to eat an item from the menu. Given t minutes, write a java program that reads the text file and ﬁnds out the maximum satisfaction that Gordon can get from eating at the restaurant. You will be given a text file with the following format:

[t][Number of items on menu]

[amount of satisfaction from eating dish 1][time taken for dish 1]

[amount of satisfaction from dish 2][time taken for dish 2]

The file is attached below.

Note:

1. Make sure to setup Spring Boot Application & follow TDD Approach for the Assignment.

2. Follow Standard Coding Practices( Naming Conventions, Class/Method/In-line Comments etc..)

3. Time given for the test to complete is 1 day.

4. Share the source code solution using Git Hub Link.

5. Brief the approach you took to solve the problem and how you achieved the end result.

6. Provide Maximum Satisfaction Value you achieved in your response.

Solution

This above problem is similar to knapsack algorithm. Given weights and values of n items, put these items in a capacity W to get the maximum total value in the knapsack. In other words, given two integer arrays val[0..n-1] and wt[0..n-1] which represent values and weights associated with n items respectively. Also given an integer W which represents knapsack capacity, find out the maximum value subset of val[] such that sum of the weights of this subset is smaller than or equal to W. You cannot break an item, either pick the complete item, or don’t pick it (0-1 property).

A simple solution is to consider all subsets of items and calculate the total weight and value of all subsets. Consider the only subsets whose total weight is smaller than W. From all such subsets, pick the maximum value subset.

Better way to solve: This problem has Overlapping Subprolems property. So the 0-1 Knapsack problem has both properties (see this and this) of a dynamic programming problem. Like other typical Dynamic Programming problems, recomputations of same subproblems can be avoided by constructing a temporary array K[][] in bottom up manner.

My Class are below,

Source file

1. MaxValueApplication : This class is Spring boot class. It starts spring boot.
2. MaximumValueController : This class is for Rest controller
3. MaximumValueService : This class following functionalities
4. Parse the data txt file
5. Find the maximum satisfaction number

Test files:

1. MaxValueApplicationTests : This test class unit test for spring boot file.
2. MaximumValueServiceTests : this test class is for unit testing for MaximumValueService class. It cover all the test cases.

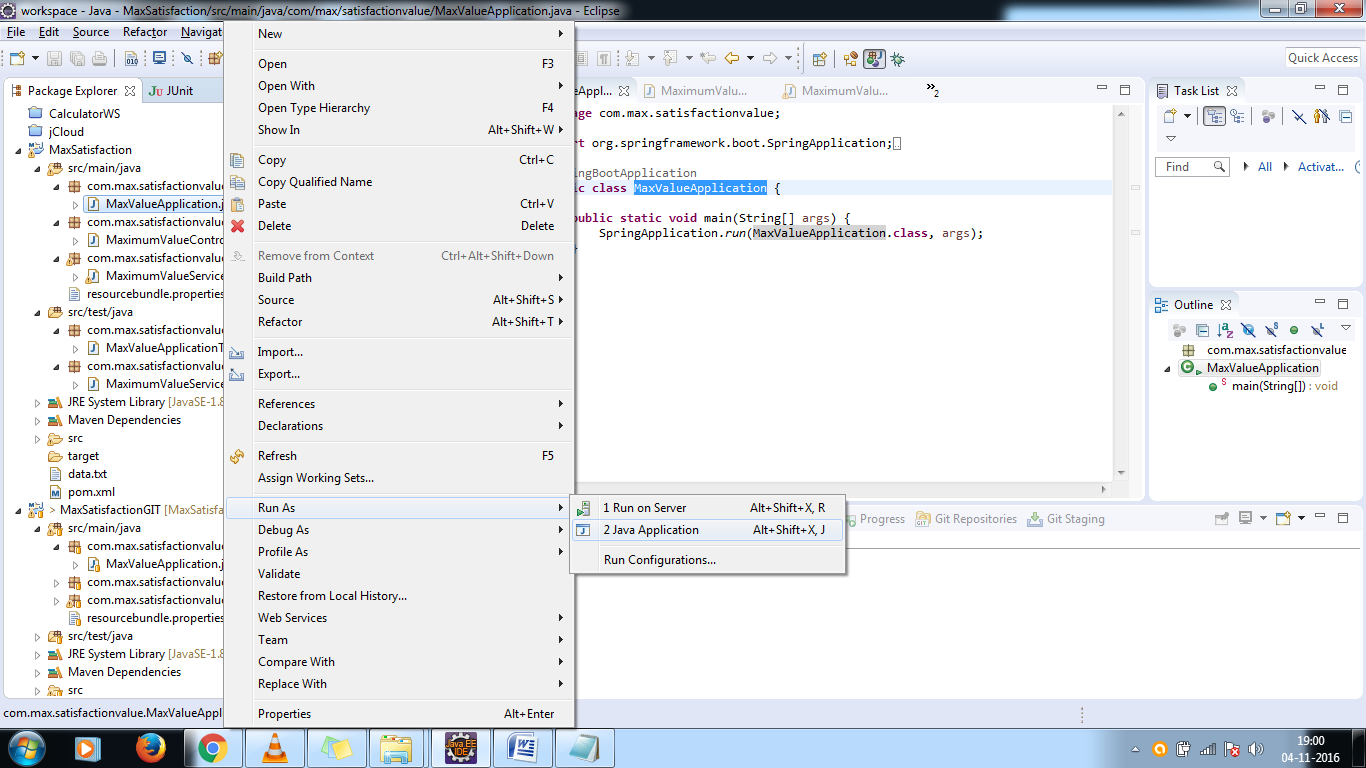
Other Files

1. resourcebundle.properties : This is for remove hard coding from the source file.
2. data.txt : It contains all the input data.
3. pom.xml : This is the maven file for dependencies.

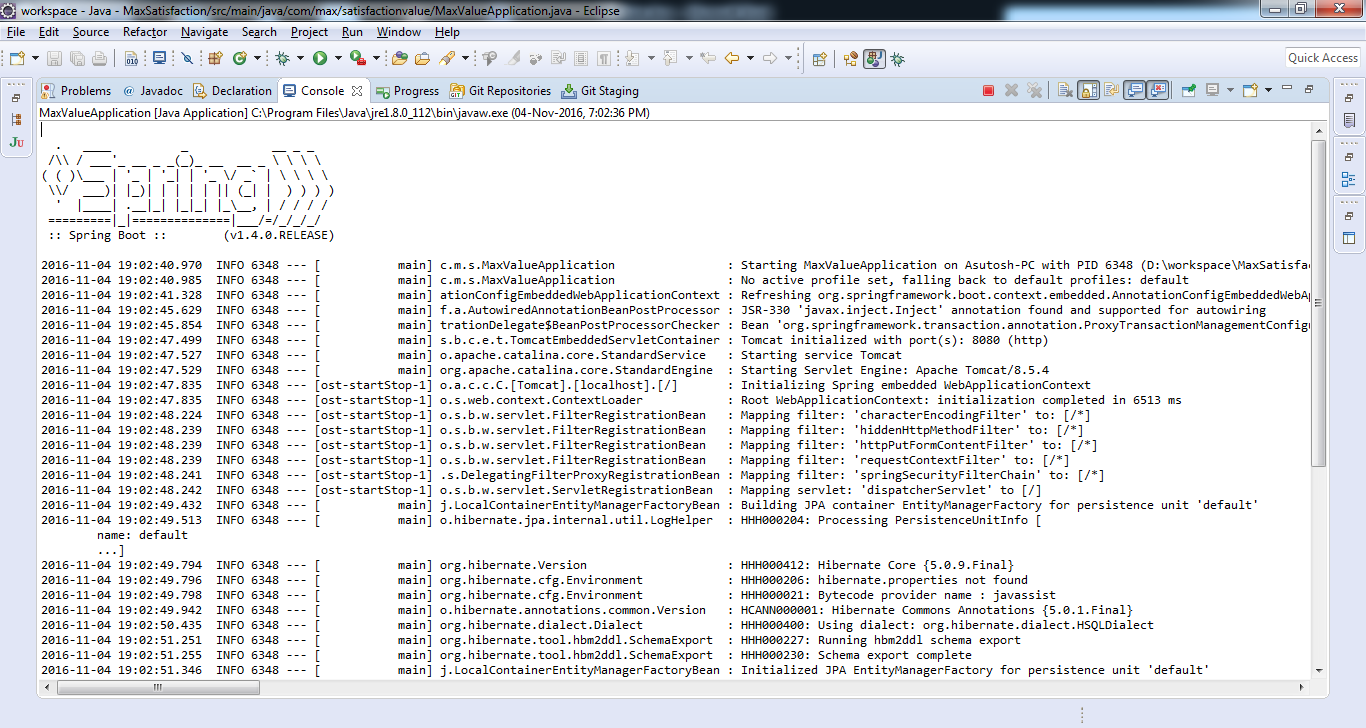
How to run the Project:

Running Source File

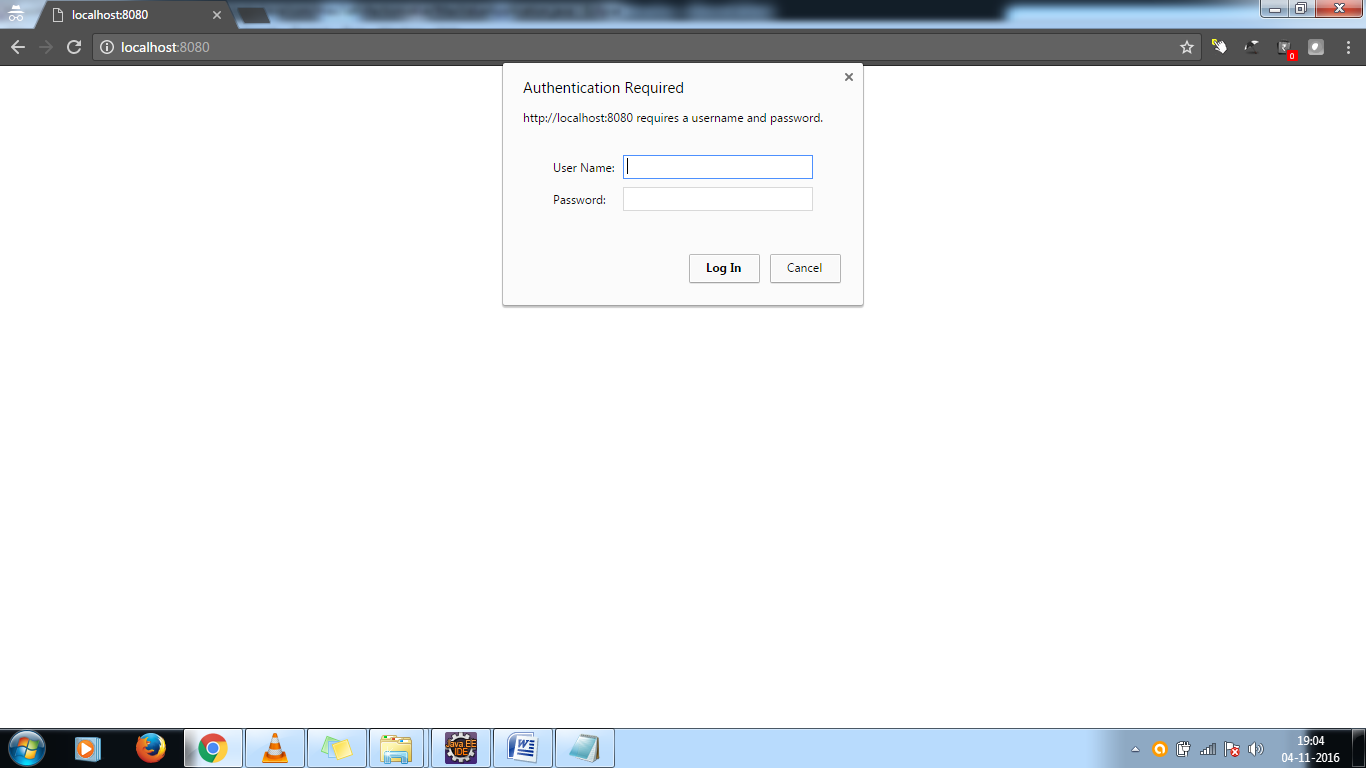
1. Right click on MaxValueApplication file. Select ‘**Run As’** from the menu then select ‘Java Application’ from sub menu.



1. It will start the Spring boot application as in the below picture.



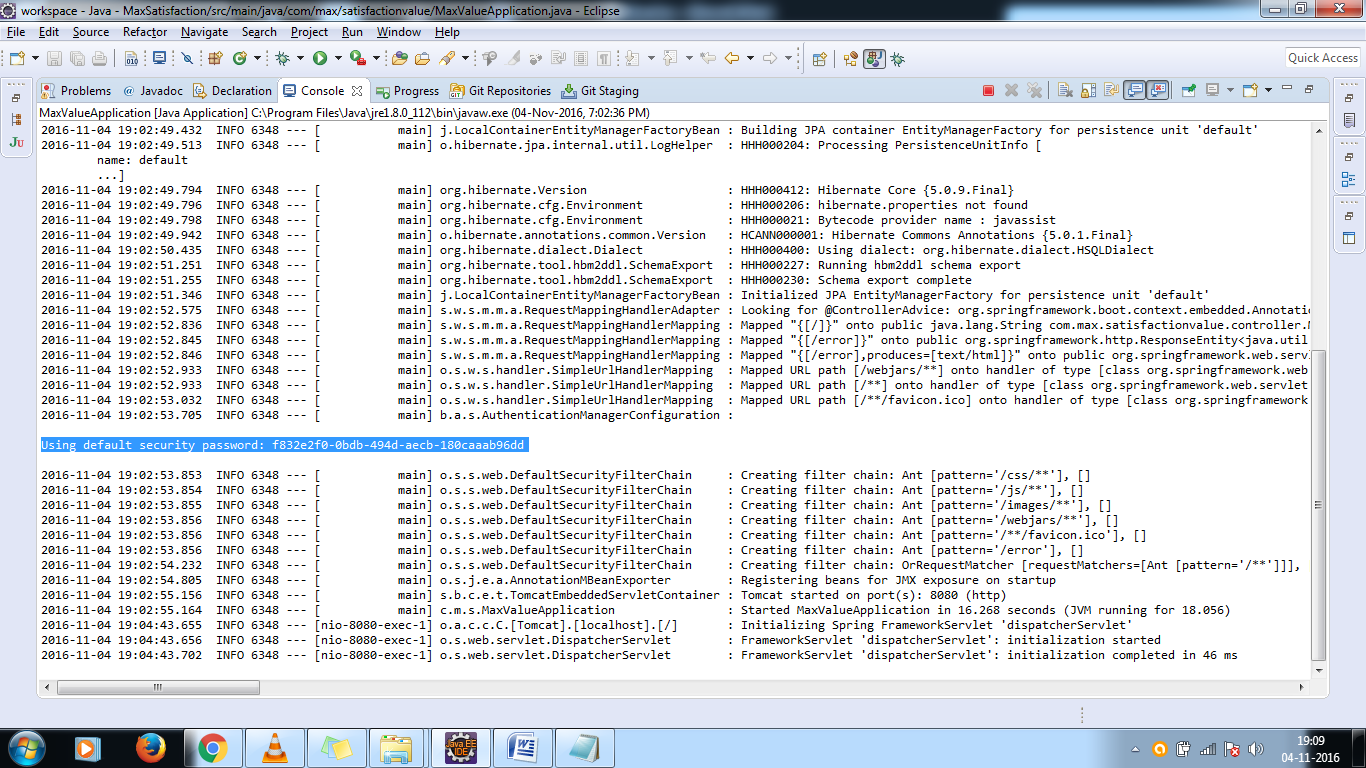
1. Then open a browser and type ‘**http://localhost:8080/**’



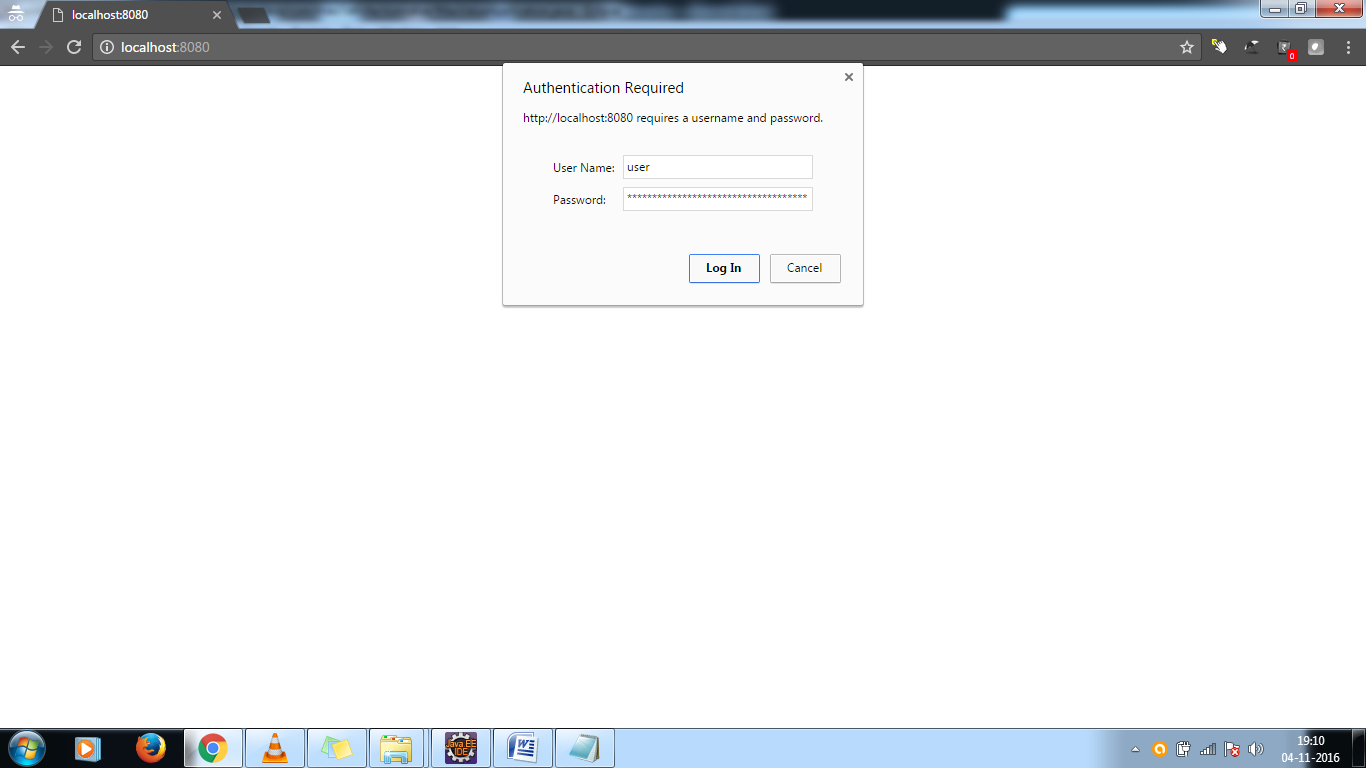
1. It will ask for the credential which are below,

User Name: user

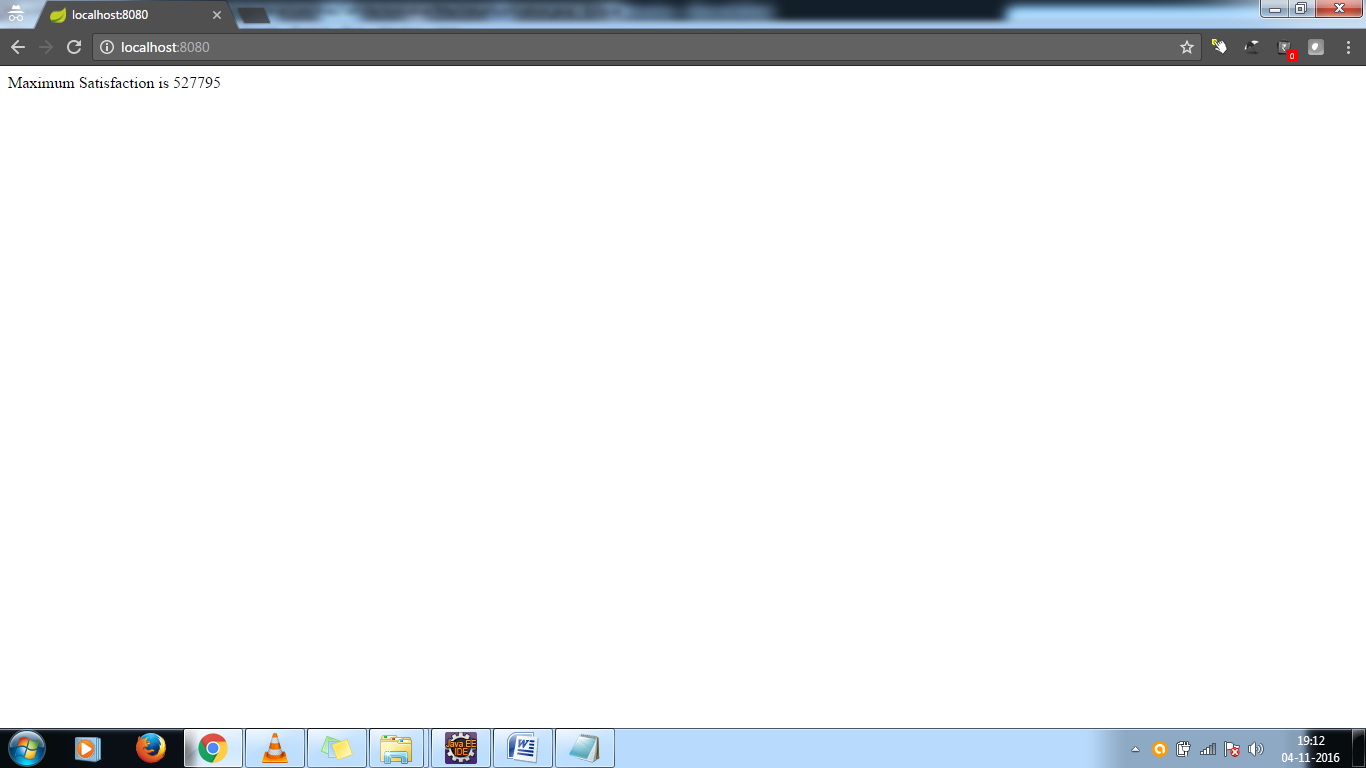
Password: it is auto generated by spring boot application which is present in the console.



1. Then click on login button.

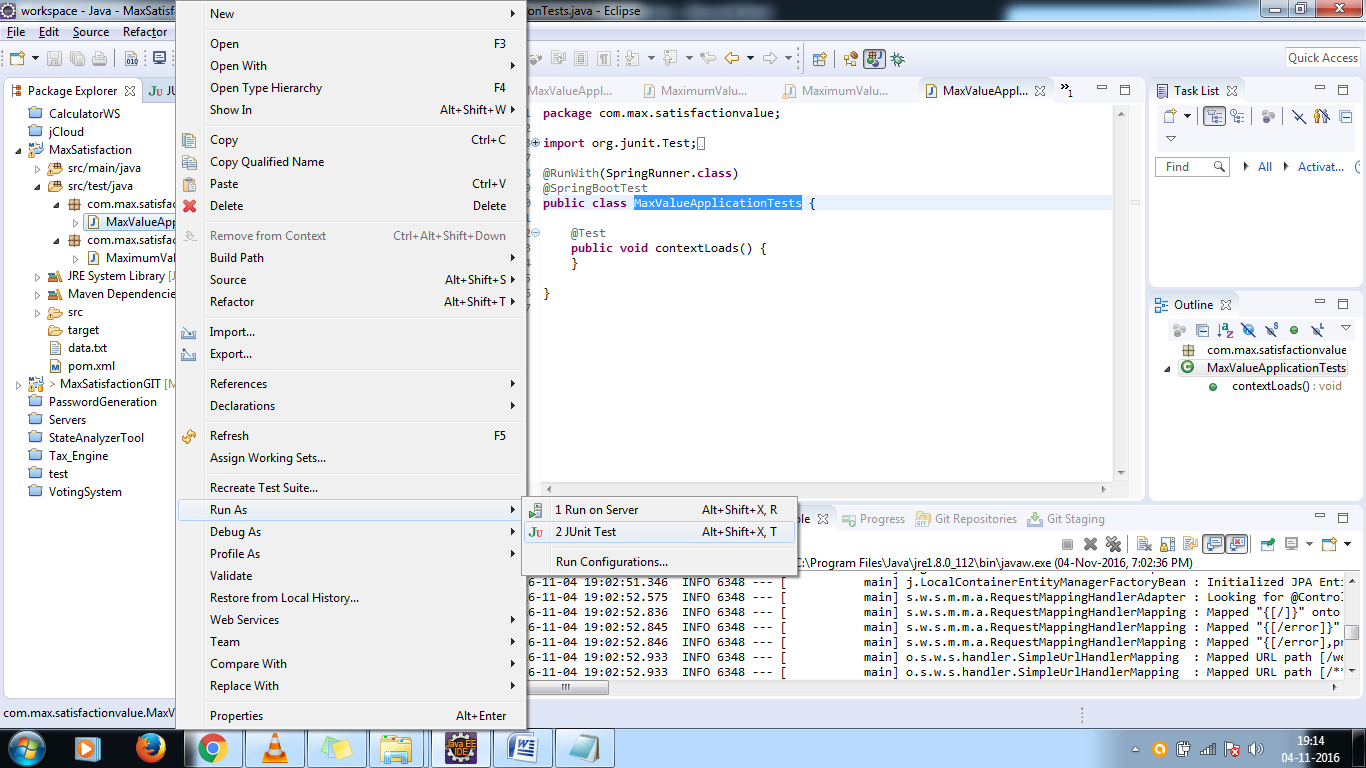


1. Then in the screen you can able to see Maximum Satisfaction number which is 527795

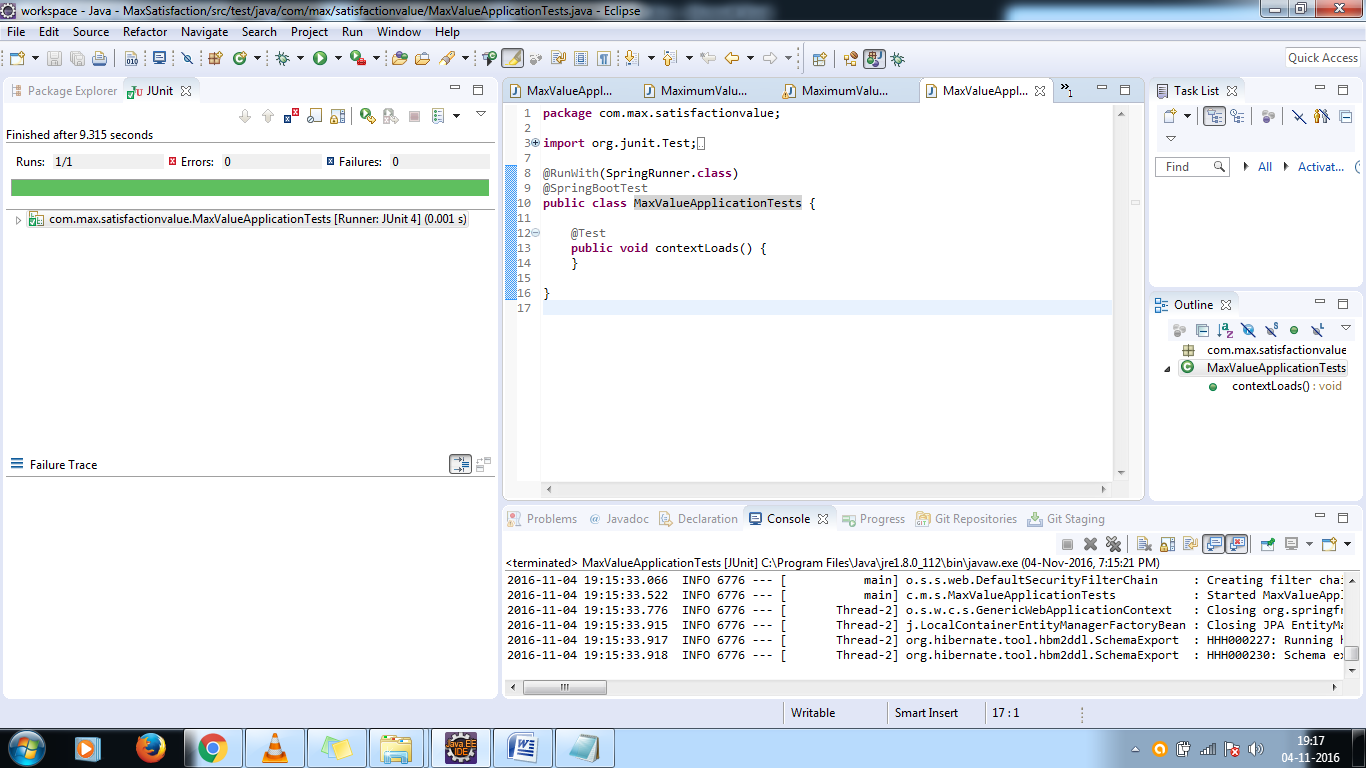


Running Test Files

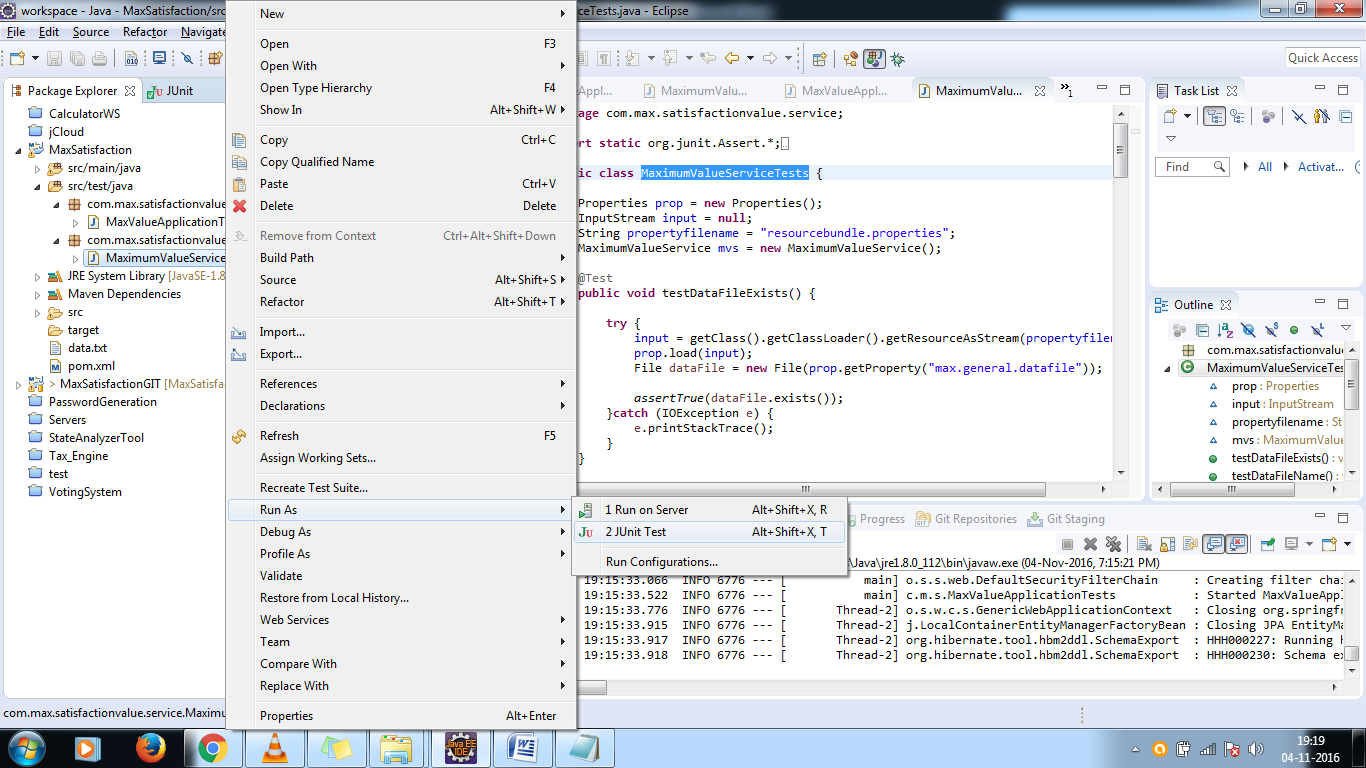
1. Go to MaxValueApplicationTests file in test folder and right click on it and select ‘Run As’ -> JUnit Test.



1. Output : Success



1. Go to MaximumValueServiceTests file in test folder and right click on it and select ‘Run As’ -> JUnit Test.



1. Output: All Success

