CS 261: - Object Oriented Programming Systems Project

TOPIC

FOOD WASTAGE MANAGEMENT WEBSITE

COOKit.com

Group Members

Deepanshu Singh (201951054)

Aditya Singh (201952202)

Sidharth Kumar Saini (201952234)

Ekansh Verma (201851043)

INDEX

Major Problem: Food Wastage

Problem Statement: The main motto of our project to save the items in fridge spoiling and rotting

The main Problems that will be Encountered will be:

- Making the program User friendly
- Inputting logical dates which are in correct format
- Calculating the days for item to Rot
- Eliminating the Rotten items
- Making the dish and including all ingredients required
- Explaining the Recipe
- As java codes are not so popular hence also implementing it using Web Development
- Making responsive Front End
- Learning how to make API in java through jersey and connecting it with frontend.
- Learning and implementing database fo the website
- Connecting all the three aspects namely front end, server side api and database and deploying it

Solution: To build a program/website that can help you reduce food wastage by giving you information about expiry date, current state of your food. Not only this it will

suggest you some recipes out of those food supplies which would be going to rot otherwise.

Detailed explanation/ Working

1. You have to take a quick look of your fridge and select the items present.
2. Then you will be telling us the date of purchase of those items and todays date.
3. Our programme then will inform you about the items that are going to rote.
4. Then among those items you can select the items you want to make a dish with.
5. Finally we will show you a list of delicious recipes that you can try,

Programming Languages/ Technologies Used: Java, JavaScript, CSS, Jersey framework, React.js, MySQL Database, Postman.

Implementation:

1. We will build a web interface for our program in the form of a website. On the front end we are using react which mainly includes writing code in JavaScript and CSS. It will have several pages including landing (in ss), ingredients and recipes. To implement this, we are using React framework and various npm libraries including react router to make the website load faster, redux etc. We will be using AJAX calls for communication to our server.

2. On the backend we are using jersey framework to create a server-side web API to send and receive calls to our frontend. Basically, we are working on 2 programs here. Since jersey and developing API are completely new to us so we have created a simple java program as an API which we are testing on postman. Since this is an OOPS project so we are also working on a more complex java program. We will try to connect our front end to this more complex java program but in case if that's not possible we will deploy the website with the simple API and our java program with more complex features that will be running on an IDE separately.

3. The Java Code will cover from all basic to advance features and we will try to make it as user friendly as possible. When the Code will start, it will ask the User the Date when food Items were bought and the date that Day. Our Code will then be checking the credibility of the dates. If the Date inputted is correct and logical, we will move forward else a reminder of the wrong date and its cause will be provided to the user and asked for reconsideration.

4. Once the Date is correct, a list of items will be displayed, and the User will have to choose the grocery he/she bought that day. Then the days of items decency(pleasantness) will be calculated by the average survival days of the items. If the food is rotten, the user will be facilitated to remove the item and it is rotten for more than 2 days. The user will be given a reminder of the item being removed and again will be facilitated to add the item back. After this process the days Left of Particular items will be displayed to the User.

5. Then the User will have a choice to make, from the displayed items to make a dish, selecting them. Obviously, there will be strictness to use items with less than 2 days left to make the dish, to stop wastage.

6. To store and fetch data required by the program we will use MySQL database.

7. The hosting services we will be using will be among google firebase or Heroku.

Codes

Github repo links:

1. (java code) https://github.com/sidharth3000/cookit-java-code
2. (front end) https://github.com/sidharth3000/COOKit-2.0
3. (API) https://github.com/sidharth3000/cookit.api

All the different parts of the codes are given below.

(a) Java Code

```
package project_1p;
import java.util.*;
import java.io.*;

//class retaled to functions with Items Inputed
class items_class
{
private String ing[]={"Butter","Milk","Egg","Tomato","Cucumber","Mushroom","Chili
","Corn","Ginger","Chicken","Chevon","Pork","Tuna","Salmon","Crab","Prawn","Soup"
,"Cola","Jam","Sauce"};
private int which[]=new int[20];
private int da[]=new int[20];
private int ex[]= {30,3,25,10,6,10,12,3,20,2,2,2,3,2,1,2,3,4,3,48};
private int arr[]={0,0,0};
Scanner z =new Scanner(System.in);
// function to display all items
public void display()
{
int i=0;
for(i=0;i<20;i++)
{
if(ing[i].length()<=7)
System.out.print(ing[i]+"               ");
else
```

```
System.out.print(ing[i]+"           ");
if(i%2!=0)
System.out.println();
}
}
// function for User to select Items For Dishes
public void ingredients()
{
int i=0;
System.out.println(" Enter Stop to Poceed Further  :-)");
String s1="";
ac:while(!s1.equals("stop"))
{
s1=z.next();
if(s1.equals("stop"))
break ac;
for(i=0;i<20;i++)
if(s1.equalsIgnoreCase(ing[i]))
which[i]=1;
else
if(which[i]!=1)
which[i]=0;
}
}
// function for calculations Days Left for item to rot
public void days_left(int r3)
{
int i1=0;
for(i1=0;i1<=19;i1++)
{
da[i1]=ex[i1]-r3;
}
}
/** function for displaying the calculated Days Left for item to rot to let user
decide
about the dish*/
public void days_of_selected()
{
int i1=0;
for(i1=0;i1<20;i1++)
{
if(which[i1]==1)
{
if(da[i1]<=0)
{
```

```
arr[0]++;
if(da[i1] !=0 && ((ex[i1]-da[i1])/(-da[i1]))<3)
{
System.out.println("O-
: YOUR ITEM: "+ing[i1]+" in the fridge has became INEDIBLE : Check the fridge to
maintain Hygeine amd Discard, if Rotten :-O");
System.out.println("O-
: For Health Pourpose, the item has been removed from your Cooking List :-O");
System.out.println("If the item seems FINE and you want to Re-
enter the item back in the list,\nPress \"Yes\"/\"Y\"/\"1\" else press \"No\"/\"N
\"/\"0\" ");
String s1;
aaa:while(true)
{
s1=z.next();
if(s1.equalsIgnoreCase("Yes") || s1.equalsIgnoreCase("Y") ||s1.equals("1"))
break aaa;
else
{
if(s1.equalsIgnoreCase("no") || s1.equalsIgnoreCase("n") ||s1.equals("0"))
{
which[i1]=0;
break aaa;
}
else
System.out.println("Wrong Input :-
(\nEnter \"Yes\"/\"Y\"/\"1\" to proceed further & \"No\"/N\"/\"0\" to Skip this S
tep");
}
}
}
else
{
System.out.println(")-
: HURRY !!! ITEM: "+ing[i1]+" in the fridge is in CRITICAL State :-(");
System.out.println("If the item seems ROTTEN and you want to Remove the item from
 the list,\nPress \"Yes\"/\"Y\"/\"1\" else press \"No\"/\"N\"/\"0\" \n This item
wont appear in your cooking items and will be deleted from database");
String s1;
aaa:while(true)
{
s1=z.next();
if(s1.equalsIgnoreCase("no") || s1.equalsIgnoreCase("n") ||s1.equals("0"))
break aaa;
else
```

```
{
if(s1.equalsIgnoreCase("Yes") || s1.equalsIgnoreCase("Y") ||s1.equals("1"))
{
which[i1]=0;
break aaa;
}
else
System.out.println("Wrong Input :-
(\nEnter \"Yes\"/\"Y\"/\"1\" to proceed further & \"No\"/N\"/\"0\" to Skip this S
tep");
}
}
}
}

if(da[i1]<=2 && da[i1]>0)
{
arr[1]++;
System.out.println("|-
: ITEM: "+ing[i1]+" in the fridge, has about "+ da[i1] + " days left ONLY!!! :-
| ");
}
if(da[i1]>2)
{
arr[2]++;
System.out.println("(-
:ITEM: "+ing[i1]+" in the fridge, has about "+ da[i1] + " days left!!!  :-
)");
}
}
}
}

// function for User to select Items For Dishes
public void selection()
{
System.out.println();
int i1=0;
int b=0;
if(arr[0]!=0)
{
System.out.println(")-
: Check these first and Proceed!!!As they are ALMOST ROTTEN :-(");
for(i1=0;i1<20;i1++)
{
```

```
if(which[i1]==1)
{
if(da[i1]<=0)
{
b++;
if(ing[i1].length()<=7)
System.out.print(ing[i1]+"              ");
else
System.out.print(ing[i1]+"          ");
if(b%2==0)
System.out.println();
}
}
}
if(b%2!=0)
System.out.println();
b=0;
System.out.println();
}
if(arr[1]!=0)
{
System.out.println("|-: CONSIDER ME !st :-|");
for(i1=0;i1<20;i1++)
{
if(which[i1]==1)
{
if(da[i1]>0 && da[i1]<=2)
{
b++;
if(ing[i1].length()<=7)
System.out.print(ing[i1]+"              ");
else
System.out.print(ing[i1]+"          ");
if(b%2==0)
System.out.println();
}
}
}
if(b%2!=0)
System.out.println();
b=0;
System.out.println();
}
if(arr[2]!=0)
{
```

```
System.out.println(":-) AVAILABLE (-:");
for(i1=0;i1<20;i1++)
{
if(which[i1]==1)
{
if(da[i1]>2)
{
b++;
if(ing[i1].length()<=7)
System.out.print(ing[i1]+"              ");
else
System.out.print(ing[i1]+"          ");
if(b%2==0)
System.out.println();
}
}
}
if(b%2!=0)
System.out.println();
System.out.println();
}
}
}
//class retaled to functions with Dates Inputed
class date
{
Scanner z=new Scanner(System.in);
private int d1,m1,y1,d2,m2,y2;
private boolean b1,b2;
// function taking Input of Starting Date given by User
public void start_d()
{
d1=z.nextInt();
m1=z.nextInt();
y1=z.nextInt();
date ob=new date();
b1=ob.correctdate(d1,m1,y1);
}
// function taking Input of That Date given by User
public void end_d()
{
d2=z.nextInt();
m2=z.nextInt();
y2=z.nextInt();
date ob=new date();
```

```
b2=ob.correctdate(d2,m2,y2);
}
// function taking Difference of Dates if Correct
public int diff_d()
{
date ob=new date();
int r1=0,r2=0;
if(b1 && b2)
{
r1=ob.func(d1,m1,y1);
r2=ob.func(d2,m2,y2);
if(r1>=0 && r2>=0)
{
if((r2-r1)>=0)
return r2-r1;
else
{
System.out.println("You Are Living in a Parallel Universe with Time Moving Backwa
rds : > \n Enter correct sequence of Dates");
return -1;
}
}
else
{
System.out.println("You Are Living in a Parallel Universe with an bygone Calender
 : > \n Input should be a year more than or 2020");
return -1;
}
}
else
{
System.out.println("You Are Living in a Parallel Universe with an Different span
of time intervals: > \n Define Proper Date");
return -1;
}
}
// function checking the faultlessness of Dates given by User
public boolean correctdate(int d,int m,int y)
{
int j=m;
boolean b=false;
if(j==1||j==3||j==5||j==7||j==8||j==10||j==12)
{
if(d<=31)
b=true;
```

```
}
if(j==4||j==6||j==9||j==11)
{
if(d<=30)
b=true;
}
if(j==2)
{
if(y%4==0 && d<=29)
b=true;
if(y%4!=0 && d<=28)
b=true;
if(y%400==0 && d>=29)
b=false;
}
return b;
}
// function calculating the number of days
public int func(int d,int m,int y)
{
int i,j,l=0,o=0,m1=m;
if(y<2020)
{
return -1;
}
for(i=2020;i<=y;i++)
{
if(i!=y)
m1=13;
else
m1=m;
for(j=1;j<m1;j++)
{
if(j==1||j==3||j==5||j==7||j==8||j==10||j==12)
o=31;
if(j==4||j==6||j==9||j==11)
o=30;
if(j==2)
{
if(i%4==0)
o=29;
else
o=28;
}
l=l+o;
```

```
}
}
return l+d;
}
}


//Main Class
public class cook
{
public static void main(String as[])
{
Scanner z=new Scanner(System.in);
date ob=new date();
int r3=0;
aa:while(true)
{
System.out.println("Enter the Date of Purchase of Food Items \"In DD MM YYYY\" ")
;
ob.start_d();
System.out.println("Enter the Date Today In \"DD MM YYYY\" ");
ob.end_d();
r3=ob.diff_d();
if(r3!=-1)
break aa;
else
System.out.println("Please Repeat the process and give Valid Dates");
}
items_class ob1=new items_class();
System.out.println("( : Please, Enter the items from the above list :)");
ob1.display();
ob1.ingredients();
ob1.days_left(r3);
ob1.days_of_selected();
System.out.println("\nWould You like to Create a Dish from Groceries");
System.out.println("Enter \"Yes\"/\"Y\"/\"1\" to proceed further & \"No\"/N\"/\"0
\" to Skip this Step");
ab:while(true)
{
String s1="";
s1=z.next();
if(s1.equalsIgnoreCase("Yes") || s1.equalsIgnoreCase("Y") ||s1.equals("1"))
{
ob1.selection();
break ab;
```

```
}
else if(s1.equalsIgnoreCase("no") || s1.equalsIgnoreCase("n") ||s1.equals("0"))
{
System.out.println("Ok");
break ab;
}
else
System.out.println("Wrong Input :-
(\nEnter \"Yes\"/\"Y\"/\"1\" to proceed further & \"No\"/N\"/\"0\" to Skip this S
tep");
}
}
}
```

OUTPUT of the Program

- When the Dates Inputed are wrong or illogical

```
Enter the Date of Purchase of Food Items "In DD MM YYYY"
32
1
2020
Enter the Date Today In "DD MM YYYY"
29
2
2020
You Are Living in a Parallel Universe with an Different span of time intervals: >
 Define Proper Date
Please Repeat the process and give Valid Dates
Enter the Date of Purchase of Food Items "In DD MM YYYY"
```

- When Dates are Correct and all items are fine

```
Enter the Date of Purchase of Food Items "In DD MM YYYY"
29
2
2020
Enter the Date Today In "DD MM YYYY"
1
3
2020
( : Please, Enter the items from the above list :)
Butter                          Milk
Egg                             Tomato
Cucumber                        Mushroom
Chili                           Corn
Ginger                          Chicken
Chevon                          Pork
Tuna                            Salmon
Crab                            Prawn
Soup                            Cola
Jam                             Sauce
 Enter Stop to Poceed Further  :-)
egg
chicken
chili
stop
(-:ITEM: Egg in the fridge, has about 24 days left!!!  :-)
(-:ITEM: Chili in the fridge, has about 11 days left!!!  :-)
|-: ITEM: Chicken in the fridge, has about 1 days left ONLY!!! :-|

Would You like to Create a Dish from Groceries
Enter "Yes"/"Y"/"1" to proceed further & "No"/N"/"0" to Skip this Step
0
Ok
```

- When the items are rotten

```
Enter the Date of Purchase of Food Items "In DD MM YYYY"
1
1
2020
Enter the Date Today In "DD MM YYYY"
5
1
2020
( : Please, Enter the items from the above list :)
Butter                          Milk
Egg                             Tomato
Cucumber                        Mushroom
Chili                           Corn
Ginger                          Chicken
Chevon                          Pork
Tuna                            Salmon
Crab                            Prawn
Soup                            Cola
Jam                             Sauce
 Enter Stop to Poceed Further  :-)
chicken
milk
cola
stop
)-: HURRY !!! ITEM: Milk in the fridge is in CRITICAL State :-(
If the item seems ROTTEN and you want to Remove the item from the list,
Press "Yes"/"Y"/"1" else press "No"/"N"/"0"
 This item wont appear in your cooking items and will be deleted from database
0
O-: YOUR ITEM: Chicken in the fridge has became INEDIBLE :
 Check the fridge to maintain Hygeine amd Discard, if Rotten :-O
O-: For Health Pourpose, the item has been removed from your Cooking List :-O
If the item seems FINE and you want to Re-enter the item back in the list,
Press "Yes"/"Y"/"1" else press "No"/"N"/"0"
1
)-: HURRY !!! ITEM: Cola in the fridge is in CRITICAL State :-(
If the item seems ROTTEN and you want to Remove the item from the list,
Press "Yes"/"Y"/"1" else press "No"/"N"/"0"
 This item wont appear in your cooking items and will be deleted from database
0

Would You like to Create a Dish from Groceries
```

- When all items are managed

```
Enter the Date of Purchase of Food Items "In DD MM YYYY"
1
1
2020
Enter the Date Today In "DD MM YYYY"
5
1
2020
( : Please, Enter the items from the above list :)
Butter                          Milk
Egg                             Tomato
Cucumber                        Mushroom
Chili                           Corn
Ginger                          Chicken
Chevon                          Pork
Tuna                            Salmon
Crab                            Prawn
Soup                            Cola
Jam                             Sauce
 Enter Stop to Poceed Further  :-)
milk
cucumber
stop
)-: HURRY !!! ITEM: Milk in the fridge is in CRITICAL State :-(
If the item seems ROTTEN and you want to Remove the item from the list,
Press "Yes"/"Y"/"1" else press "No"/"N"/"0"
 This item wont appear in your cooking items and will be deleted from database
1
|-: ITEM: Cucumber in the fridge, has about 2 days left ONLY!!! :-|

Would You like to Create a Dish from Groceries
Enter "Yes"/"Y"/"1" to proceed further & "No"/N"/"0" to Skip this Step
1

)-: Check these first and Proceed!!!As they are ALMOST ROTTEN :-(

|-: CONSIDER ME !st :-|
Cucumber

/n/nThe feature to make dishes and their recepie will be added soon
```

(a) Frontend JavaScript

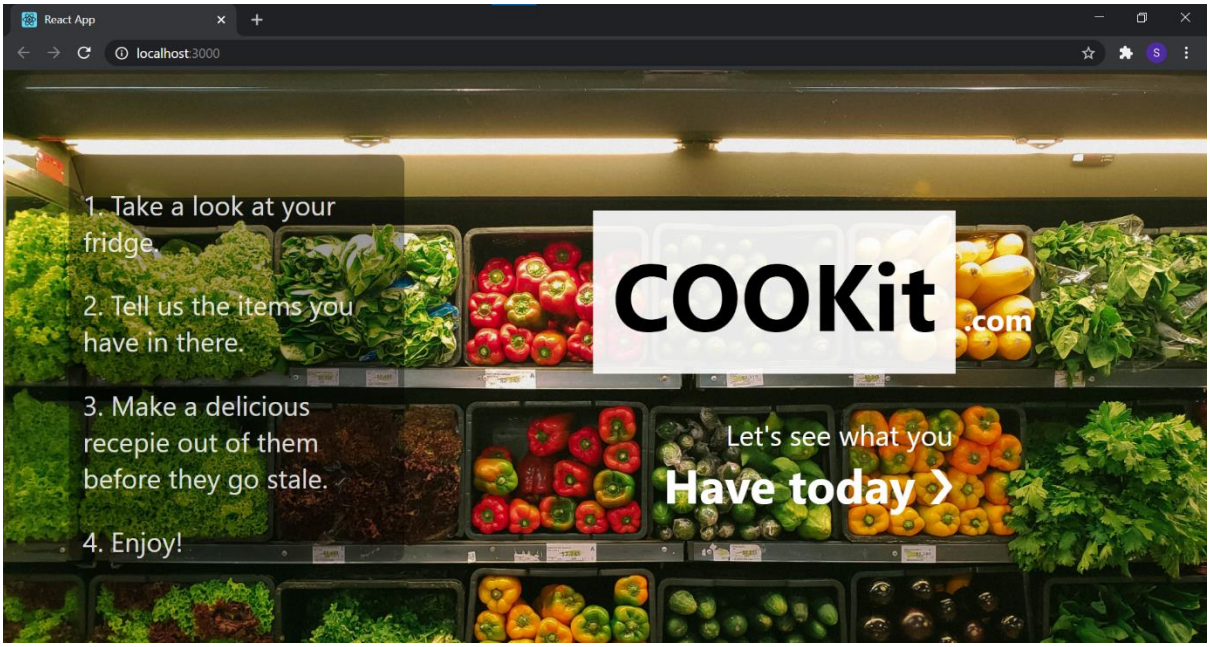
```
(a)import React, {Component} from 'react';
(b)
(c)import classes from './Landing.css';
(d)
(e)class Signin extends Component {
(f)    render() {
(g)        return(
(h)            <div className={classes.container}>
(i)
(j)                <div className={classes.left}>
(k)                    <p  className={classes.head}>1. Take a look at your fr
   idge.</p>
(l)                    <p  className={classes.head}>2. Tell us the items you
   have in there.</p>
(m)                    <p  className={classes.head}>3. Make a delicious recep
   ie out of them before they go stale.</p>
(n)                    <p  className={classes.head}>4. Enjoy!</p>
(o)                </div>
(p)
(q)                <div className={classes.right}>
(r)                    <h1><span className={classes.name}>COOKit</span><span
   className={classes.plain}> .com</span></h1><br/><br/>
(s)                    <div className={classes.start}>
(t)                        <p className={classes.h1}>Let's see what you</p>
(u)                        <p className={classes.h2}>Have today  &#10095;</p>
(v)                    </div>
(w)
(x)                </div>
(y)
(z)            </div>
(aa)
(bb)             )
(cc)         }
(dd)     }
(ee)
(ff)     export default Signin;
```

(b) Frontend CSS

```css
.container{
  width: 100%;
  height: 100%;
  position: absolute;
  /* position:relative; */
  background-image: url('../../assets/grocery.jpg');
  /* opacity: 0.32; */
  background-size: cover;
  z-index: 0;

  display: grid;
  grid-template-columns: 1fr 2fr;

  justify-content: center;
  align-items: center;
}

.left{
  margin: 0px 70px;
  height: 70%;
  width: 80%;

  padding: 5px 0px 0px 15px;
  background-color: rgba(0, 0, 0, 0.349);
  font-size: 1.8em;
  color: rgba(255, 255, 255, 0.842);
  border-radius: 10px;
}

.right{
  margin: 0px 70px;
  margin-right: 70px;
  padding: 10px 0px 0px 15px;
  /* background-color: rgba(0, 0, 0, 0.349); */
  text-align: center;
  /* border-radius: 10px; */
  /* color: white; */
}

.name{
  background-color: rgba(255, 255, 255, 0.904);
  color: rgb(0, 0, 0);
  font-size: 100px;
```

```
  width: 100%;
  padding: 20px;
}

.plain{
  background-color: none;
  color: rgb(255, 255, 255);
}

.start{
  color: white;
  text-align: right;
  padding-right: 200px;
}

.h1{
  font-size: 30px;
  margin: 0px;
}

.h2{
  font-size: 50px;
  font-weight: bold;
  margin: 0px;
  cursor: pointer;
  transition: all 0.3s;
  /* background-color: white; */
  /* color: black; */
  /* wi */
}

.h2:hover{
  background-color: white;
  border-radius: 5px;
  padding-left: 5px;
  color: black;
  text-align: left;
}
```

(c) Interface (Landing page)



(d) Backend/Server-side API (for testing)

```
package com.javatpoint.rest;
import javax.ws.rs.GET;
import javax.ws.rs.Path;
import javax.ws.rs.Produces;
import javax.ws.rs.core.MediaType;

@Path("/hello")
public class Hello {

  @GET
  @Produces(MediaType.TEXT_PLAIN)
  public String sayPlainTextHello() {
    return "Hello Jersey Plain";
  }


  @GET
  @Produces(MediaType.TEXT_XML)
  public String sayXMLHello() {
    return "<?xml version=\"1.0\"?>" + "<hello> Hello Jersey" + "</hello>";
  }
```

```
  // This method is called if HTML is requested
  @GET
  @Produces(MediaType.TEXT_HTML)
  public String sayHtmlHello() {
    return "<html> " + "<title>" + "Hello Jersey" + "</title>"
        + "<body><h1>" + "Hello Jersey HTML" + "</h1></body>" + "</html> ";
  }
}
```

Future…

1. Currently we are using a file in the jersey project only acting as a dummy database. In future we will be deploying a real database for the project.
2. Till date we have made landing page of the website. Currently we are testing backend on postman and as soon its ready we will further develop the frontend.
3. The program will be fully installed with all the useful functions for making the recipe.
4. Diversifying the number of items and take new non existing inputs from the user.
5. Reducing the work of user utmost by making inputs as few as possible.
6. We will try to fetch the information about which items were purchased on a particular day by implementing a database so you don't have to manually enter it.

Distant future

1. We can add a barcode scanner beside the refrigerator and scan every item to feed its expiry date to the database that will be synced to your smartphone. Every time any item is going to expire, you will get a notification and you can act!
2. Not limiting code to a restricted item and using as many items as possible.