DEMANDEst - AI POWERED FOOD DEMAND FORECASTER

TEAM ID : PNT2022TMID32677

A PROJECT REPORT

Submitted by

S.RAGASUDHA (813819104070)

S.RAMYA (813819104075)

R.SANGEETHA (813819104082)

J.SRIDEVI (813819104097)

CONTENTS

1. **INTRODUCTION**
    - 1.1 Project Overview
    - 1.2 Purpose
2. **LITERATURE SURVEY**
    - 2.1 Existing problem
    - 2.2 Problem Statement Definition
    - 2.3 References
3. **IDEATION & PROPOSED SOLUTION**
    - 3.1 Empathy Map Canvas
    - 3.2 Ideation & Brainstorming
    - 3.3 Proposed Solution
    - 3.4 Problem Solution fit
4. **REQUIREMENT ANALYSIS**
    - 4.1 Functional requirement
    - 4.2 Non-Functional requirements
5. **PROJECT DESIGN**
    - 5.1 Data Flow Diagrams
    - 5.2 Solution & Technical Architecture
    - 5.3 User Stories
6. **PROJECT PLANNING & SCHEDULING**
    - 6.1 Sprint Planning & Estimation
    - 6.2 Sprint Delivery Schedule
    - 6.3Reports from JIRA
7. **CODING & SOLUTIONING**
    - 7.1 Feature 1
    - 7.2 Feature 2
8. **TESTING**
    - 8.1 Test Cases
    - 8.2 User Acceptance Testing
9. **RESULT**
10. **ADVANTAGES & DISADVANTAGES**
11. **CONCLUSION**

12. **FUTURE SCOPE**

13. **APPENDIX**

Source Code

GitHub & Project Demo Link

LIST OF FIGURES

1.Empathy map canvas

2. Ideation

3.Brainstorming

4.Problem Solution Fit

5.Data Flow Diagram

6.Solution and Technical Architecture

7. JIRA Roadmap

8.Sprint 1

9.Sprint 2

10.Sprint 3

11.Sprint 4

LIST OF TABLES

1.Proposed Solution

2.Functional Requirement

3.Problem statement Definition

4.Non-Functional Requirement

5.Components and Technologies

6.Application Characteristics

7.User Stories

8.Sprint Planning and Estimation

9.Sprint Delivery Schedule

1. INTRODUCTION

1.1 PROJECT OVERVIEW

A food delivery service has to deal with a lot of perishable raw materials which makes it all, the most important factor for such a company is to accurately forecast daily and weekly demand. Too much inventory in the warehouse means more risk of wastage, and not sufficient could lead to out-of-stocks - and push customers to seek solutions from your competitors. The replenishment of majority of raw materials is done on weekly basis and since the raw material is perishable, the procurement planning is of utmost importance, the task is to predict the demand for the next 10 weeks.

1.2 PURPOSE

The main aim of this project is to create an appropriate machine learning model to forecast the number of orders to gather raw materials for next ten weeks. To achieve this, we should know the information about of fulfillment center like area, city etc., and meal information like category of food sub category of food price of the food or discount in particular week. By using this data, we can use any classification algorithm to forecast the quantity for 10 weeks. A web application is integrated with the ml model .

2. LITERATURE SURVEY

S.NO	TITLE	YEAR OF PUBLISHING	AUTHORS	THEME	INFERENCE
1.	Demand Forecasting for production planning in a food company.	Jan-2015	• Nathalia Barbosa • Kelly Alonso Costa	Food demand in beverage industry.	The food products have a factor that limits the maintenance of stocks, the short perishability. These products have a period in which they keep their characteristics and should be consumed before being considered unsuitable for consuming. Thus, it is suggested for future works that the short perishability of products must be taken into account when evaluating the results obtained by the quantitative methods. To make possible not only plan the production to satisfy the forecasted demand, but also contribute to minimize the loss of products due to its short perishability and consequently, improving the profitability of the company.
2.	Food demand Forecast – A case study in the agri-food sector.	Nov –2021	• Syrine Guinoubi • Yasmina Hani	Food demand Forecasting.	In this article, they have developed a literature review of different forecasting methods, both qualitative and quantitative. The fact that the demand for the products is continuous; not intermittent; this could justify their choice to work with quantitative forecasting methods.
3.	Reducing fresh fish waste while ensuring fish availability.	May-2022	• Vera Lucia Migueis • Joao Pereira	Demand forecasting using censored data and machine learning.	In recent years, retailers have used advanced methods of collecting data to gain more information about their customers and their buying behavior. This data collection has promoted a huge opportunity for improving operations. Thus, retailers have concentrated on developing more accurate forecasting models that help them make decisions that are more data-driven and less intuition-based.
4.	Demand forecasting in supply chains.	Jan-2007	• Rustam Vahidov • Kevin Laframboise	Machine learning based demand forecasting.	From the results we can see that one of the ML approaches, the SVM under the super wide modeling approach is at the top of all three data sets by providing consistently better performance. If we ignore the super wide models, we find that the results of previous research and the very large M3 competition were essentially reproduced, that is, simple techniques outperform the more complicated and sophisticated approaches.

2.1 EXISTING PROBLEM

The replenishment of majority of raw materials is done on weekly basis and since the raw material is perishable , the procurement planning is of utmost importance . Also the recruiting staff members at the fulfillment center is an prospect wherein the prediction of orders would be beneficial. Although this is a process which can be done manually .

2.2 PROBLEM STATEMENT DEFINITION

Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	Organization	Get accurate prediction as a result.	Need prediction for long duration.	It takes long time to do the analysis	Disappointed
PS-2	Small Scale	To get prediction in low cost	Its not affordable for an accurate prediction	Prediction takes much time and effort	Depressed

2.3 REFERENCES

[1] Patrick Meulstee and Mykola Pechenizkiy," Food Sales Prediction: "If Only It Knew What We Know"" 2008 IEEE International Conference on Data Mining Workshop.

[2] Yoichi Motomura, Baysian network, Technical Report of IEICE, Vol.103, No.285, pp.25-30, 2003.

[3] Yoichi Motomura, Baysian Network Softwares, Journal of the Japanese Society for Artificial Intelligence, Vol.17 No.5, pp.1-6, 2002.

[4] D. Adebanjo and R. Mann. Identifying problems in forecast-ing consumer demand within the fast paced commodity sector. Benchmarking: An International Journal, 7(3):223– 230, 2000.

[5] Bohdan M. Pavlyshenko," Machine-Learning Models for Sales Time Series Forecasting"

3. IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

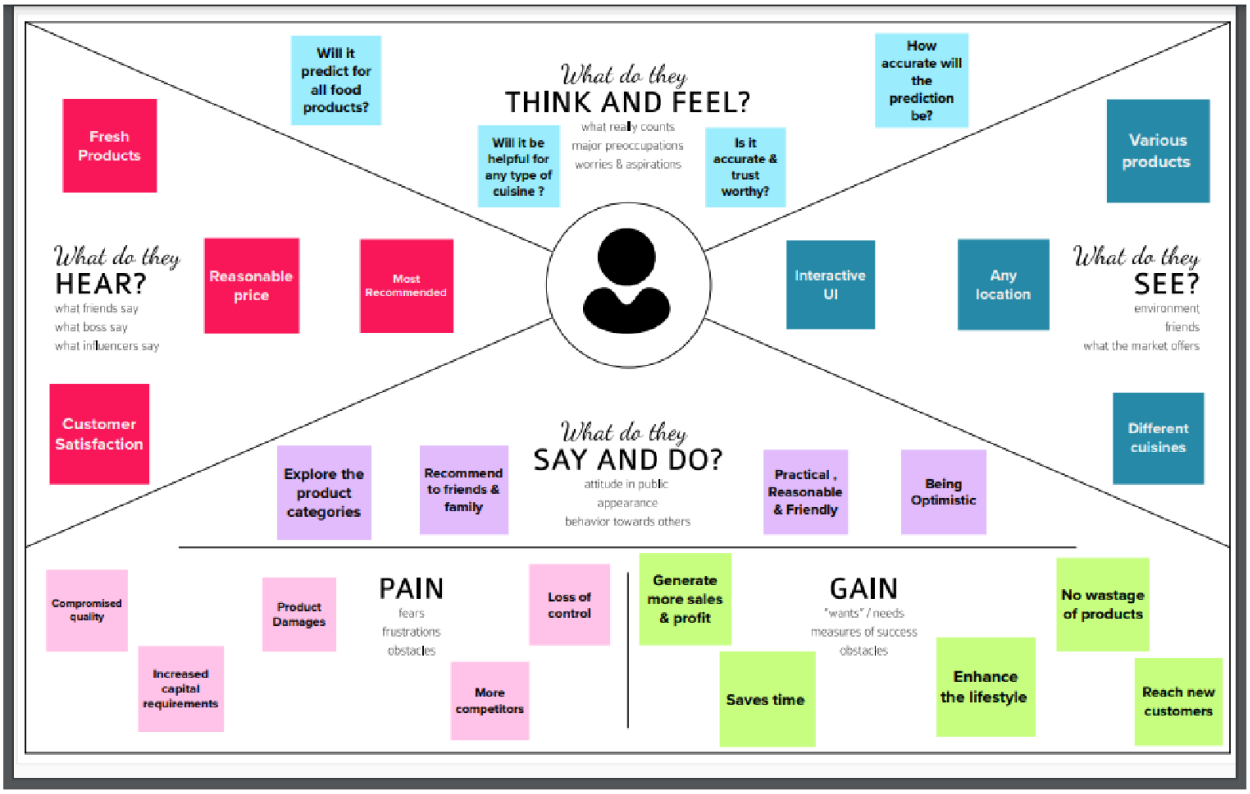

Figure - 1

3.2 IDEATION & BRAINSTORMING

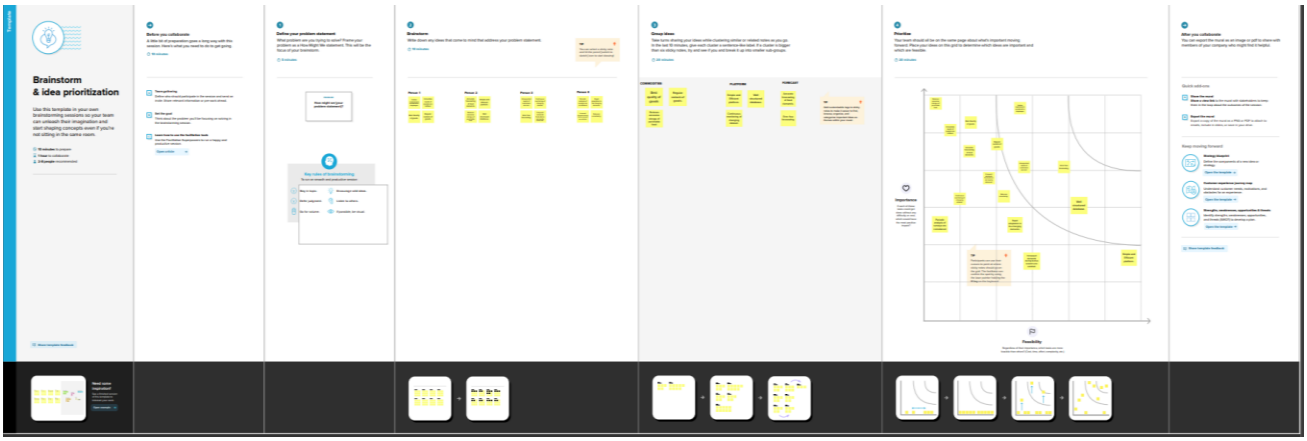

Figure - 2

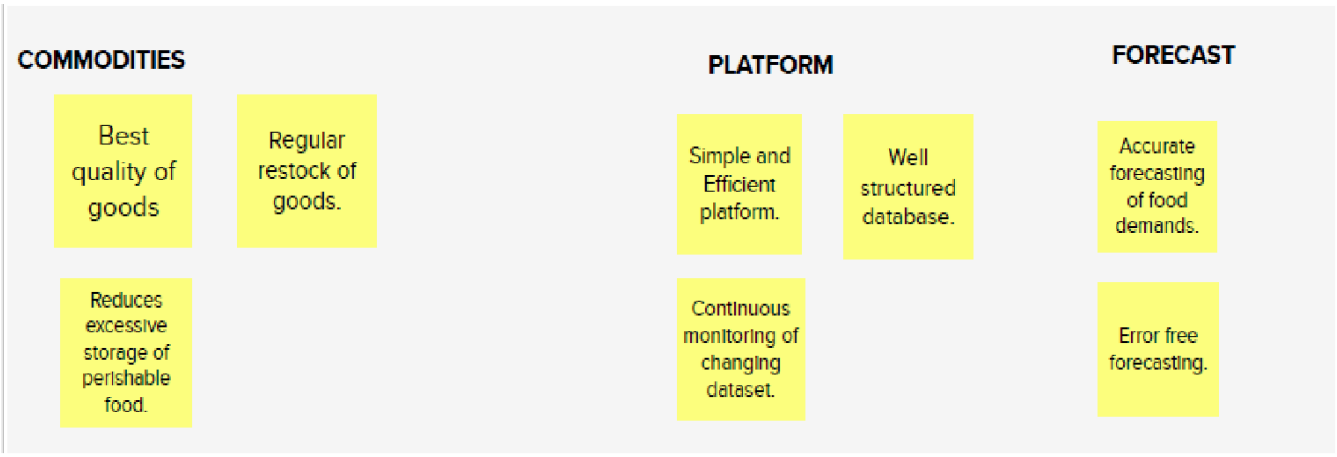

Figure - 3

3.3 PROPOSED SOLUTION

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	To predict the number of orders according to the user's requirements.
2.	Idea / Solution description	To create an appropriate machine learning model to forecast the number of orders to gather raw materials for next ten weeks.
3.	Novelty / Uniqueness	Creative and animatory user interface, by which the user will be attracted and easy to use.
4.	Social Impact / Customer Satisfaction	By accurate predictions, wastage of food is reduced.
5.	Business Model (Revenue Model)	Advertisements, premium for large no. of users (organization)
6.	Scalability of the Solution	Large no. of users can access since the website is hosted in cloud .

3.4 PROBLEM SOLUTION FIT

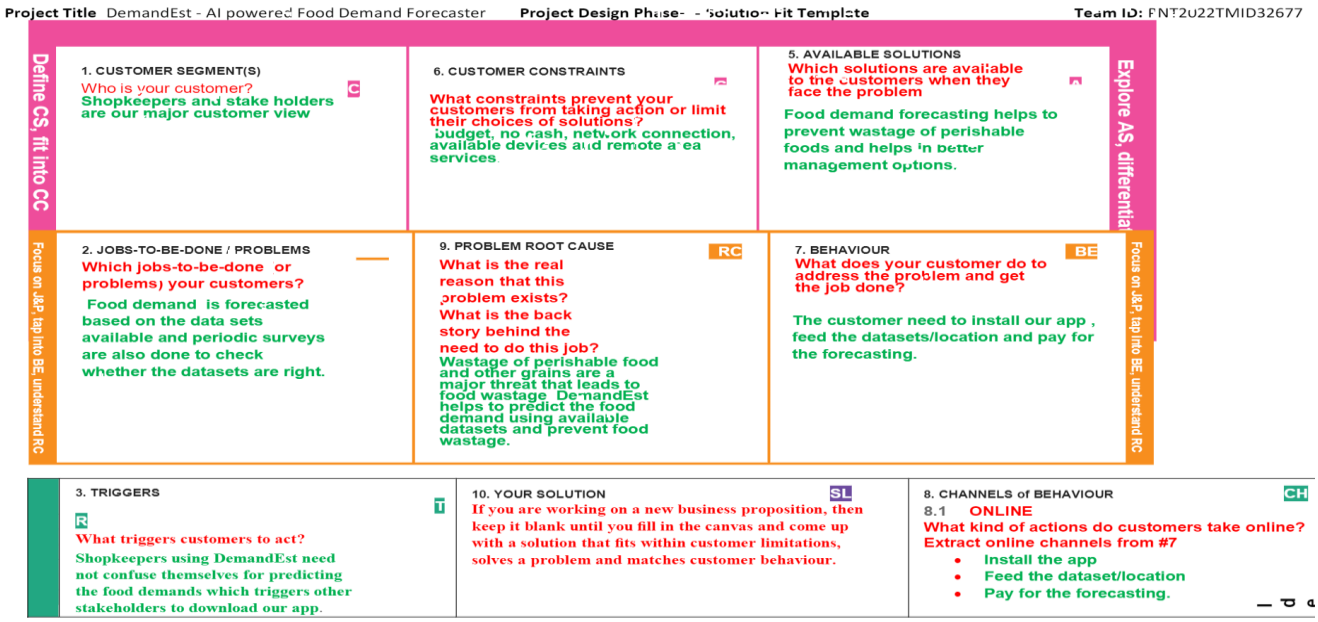

Figure-4

4. REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENT

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Gmail
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Home Page	User is directed to interactive home page.
FR-4	Premium	Organizations which need a prediction to made in frequent intervals can make use of premium.
FR-5	Updates & Support	Frequent updates can be expected which makes the work smoother and easy.
FR-6	Own space	Create their own organizational space which can be used by the admin & co-workers.

4.2 NON-FUNCTIONAL REQUIREMENT

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Organizations and businesses which are based on food products can predict the requirements for a specific time.
NFR-2	**Security**	Data about their requirements about food or about the organization are authenticated.
NFR-3	**Reliability**	Since it's a secured web application its more reliable.
NFR-4	**Performance**	Our application is designed in such a way that any number of users can access it at once, it will work perfectly without lagging.
NFR-5	**Availability**	No specific constraints for accessing our application, so it's available in any platform irrespective of OS etc.
NFR-6	**Scalability**	Since our application is hosted on IBM cloud, its scalable.

5. PROJECT DESIGN

5.1 DATA FLOW DIAGRAM

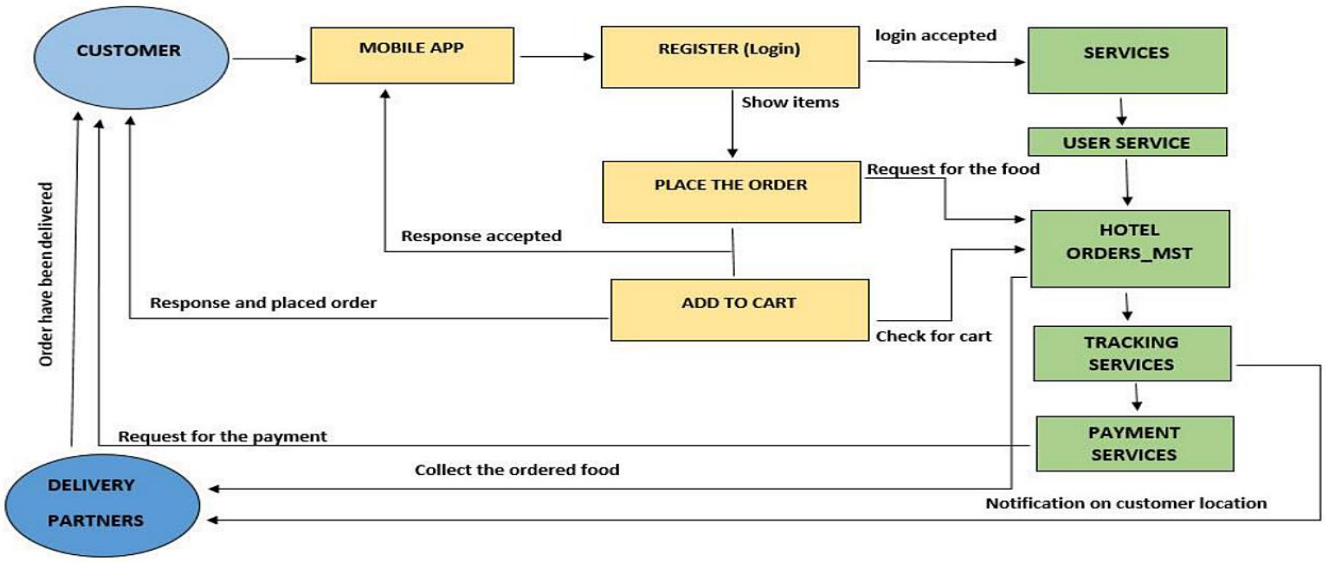

Figure - 5

5.2 SOLUTION & TECHNICAL ARCHITECTURE

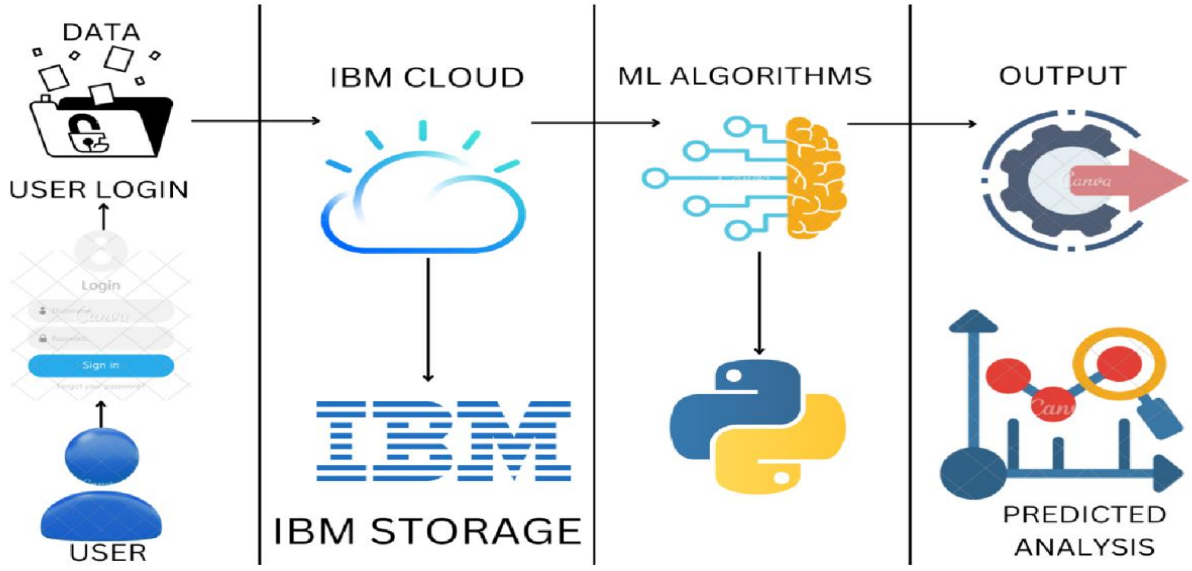

Figure - 6

Components and Technologies

S.No	Component	Description	Technology
1.	Customer	By using Mobile App and Through online registration.	HTML, CSS, JavaScript
2.	Restaurant	It includes all the goods and services that the restaurants meals.	Online transactions
3.	Geolocation	Used to reach the destination.	Google map, user address.
4.	Platform owner	Wait for the delivery of food.	Mobile phones and online websites.
5.	Database Analytics	Data Type, Configurations etc.	MySQL, NoSQL, etc
6.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant etc.
7.	File Storage	User information.	IBM Block Storage or Other Storage Service or Local Filesystem
8.	Amazon s3 bucket	Storage with data availability.	HTTP interface
9.	Cloud watch alarm	Purpose of External API used in the application	Notification services.

Application Chracteristics

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Google chrome, online websites	Technology of Opensource framework
2.	Security Implementations	Authentications through OTP	Through mobile phones.
3.	Scalable Architecture	Based on quality. Based on taste.	Quality assurance Quality control.
4.	Availability	Through online	Online system
5.	Performance	Provide qualitative food Encourage customer loyalty. Boost sales.	Testing shows preference for mistakes. Detecting the defect within a software.

5.3 USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail	I can receive confirmation Gmail & click confirm	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password	I can receive confirmation Gmail & click confirm	High	Sprint-1
	Dashboard	USN-6	As a user I can check the facilities of the Dashboard	I can receive and help to more support in the Dashboard	High	Sprint-2
Customer (Web user)	Web user	USN-7	As a user I can use this website to learn more about this quantity method	I can receive confirmation & click to check the quantity	Low	Sprint-2
Customer Care Executive	Customer order delivery	USN-8	Doorstep delivery. easy process to get the order.	Rating on delivery partner and food quality.	Medium	Sprint-2
Administrator	Hotel management, website holders	USN-9	Choosing the restaurant. Multiple choice for restaurant profile.	Advertising through websites	Low	Sprint-1
Customer(order)	Order	USN-10	As a user I can order to deliver a product of items	I can receive your order to verify the delivery of the product	High	Sprint-1

6. PROJECT PLANNING & SCHEDULING

6.1 SPRINT PLANNING & ESTIMATION

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my username and password.	5	High	Ragasudha.S
Sprint-1	Login	USN-2	As a user, I can log into the application by entering email & password	5	High	Ragasudha.S
Sprint -1	Explore	USN-3	As a registered user, I can explore the various options available on the home page.	8	Medium	Ramya.S
Sprint-2	User Manual	USN-4	As a registered user, I can take a tour over the user manual and can understand the functionalities.	5	Low	Sangeetha.R
Sprint-2	Predict	USN-5	As a registered user, I can pay and make predictions on the website.	13	Medium	Ragasudha.S
Sprint-3	Premium membership	USN-6	As a premium user, I can deposit money on the wallet and make use of many discounts available.	5	High	Sridevi.J
Sprint-3	Survey	USN-7	As an administrator , I conduct periodic surveys to keep track of food demands.	13	Medium	Sridevi.J
Sprint-4	Inventory	USN-8	As an administrator , I should be able to alter or delete food options in the list.	13	Medium	Sangeetha.R
Sprint-4	Maintenance	USN-9	As an administrator, I can edit the user's details and premium valet management.	5	High	Ragasudha.S

6.2 SPRINT DELIVERY SCHEDULE

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	18	6 Days	24 Oct 2022	29 Oct 2022	18	29 Oct 2022
Sprint-2	18	6 Days	31 Oct 2022	05 Nov 2022	18	31 Oct 2022
Sprint-3	18	6 Days	07 Nov 2022	12 Nov 2022	18	07 Nov 2022
Sprint-4	18	6 Days	14 Nov 2022	19 Nov 2022	18	14 Nov 2022

6.3 REPORTS FROM JIRA

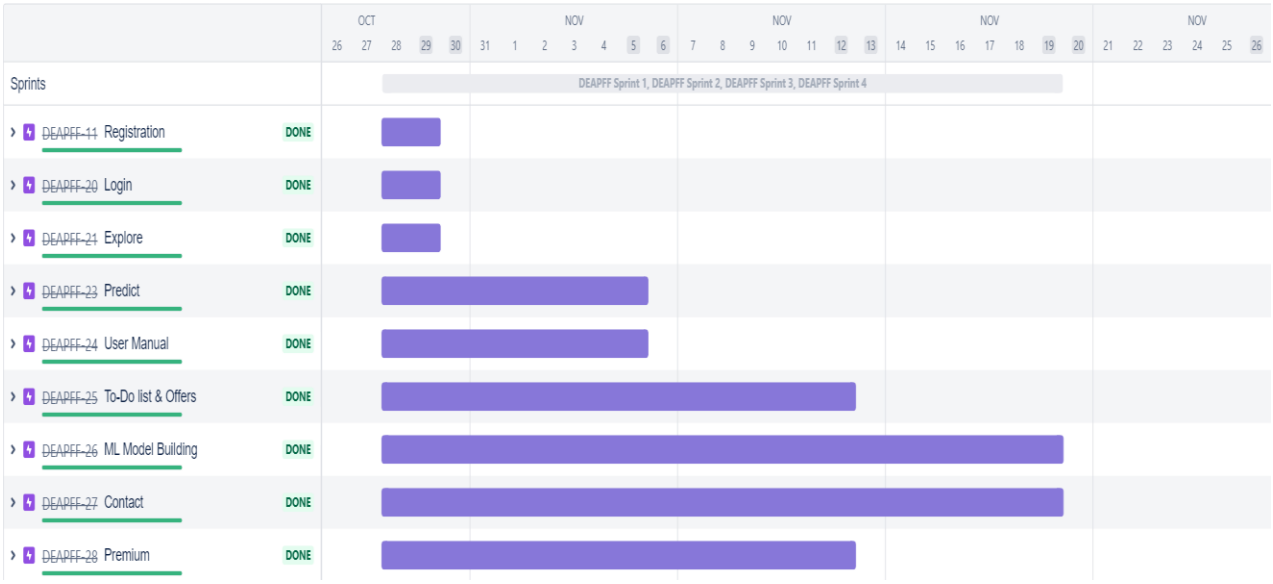

Figure - 7

SPRINT -1

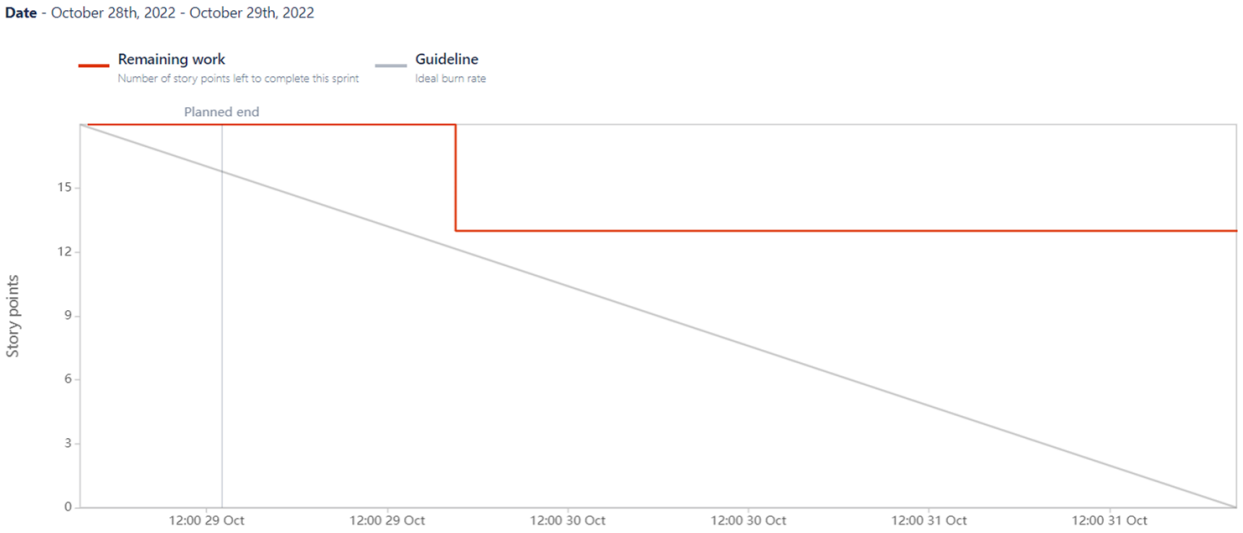

Figure – 8

SPRINT - 2

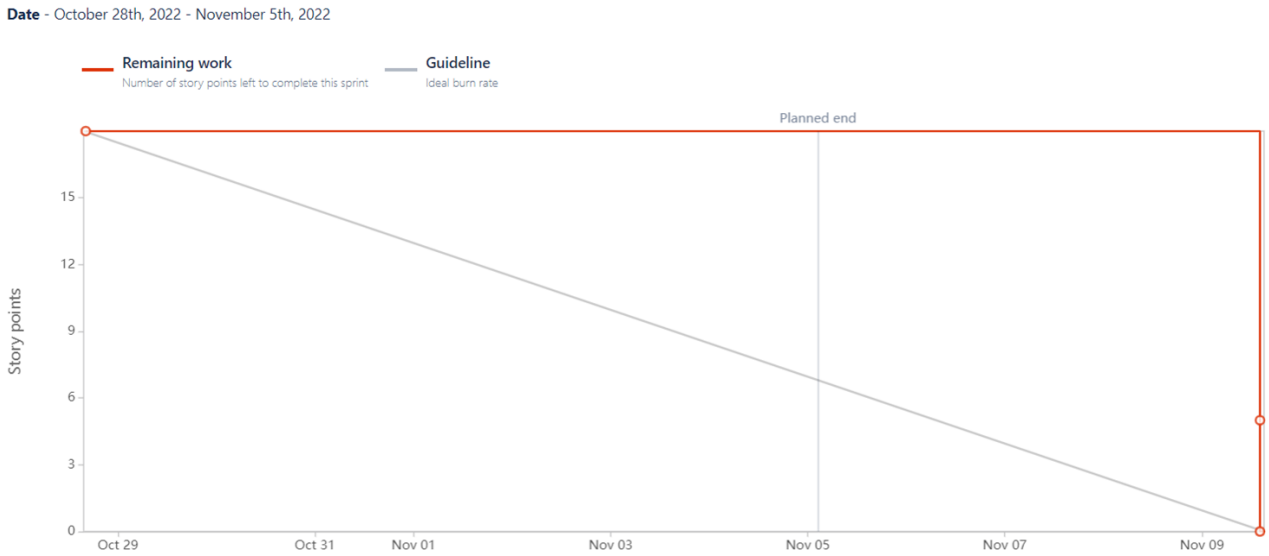

Figure - 9

SPRINT - 3

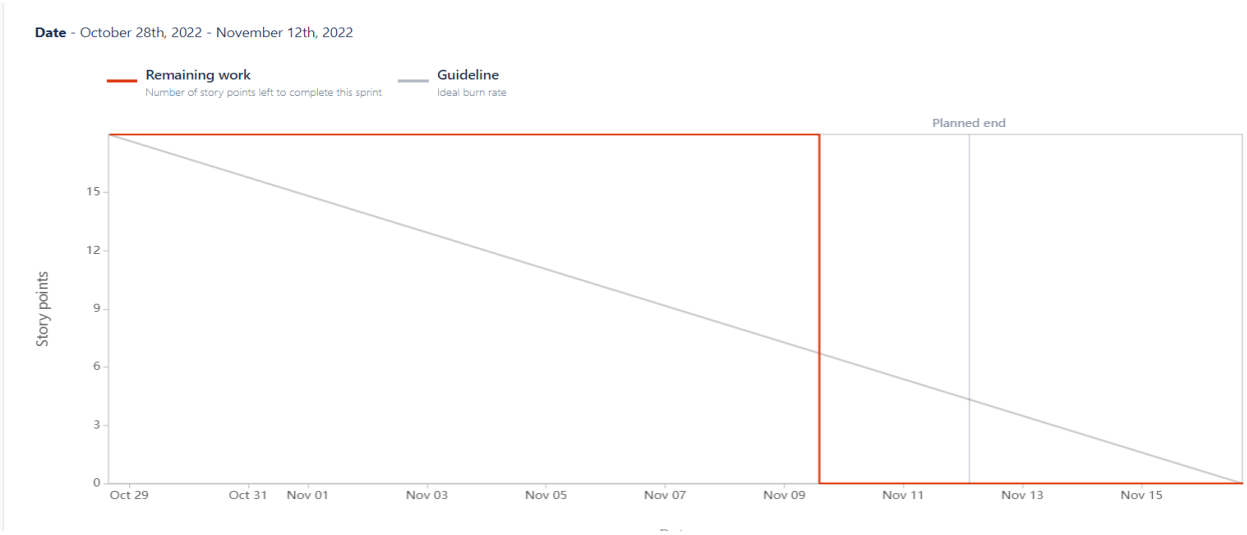

Figure - 10

SPRINT - 4

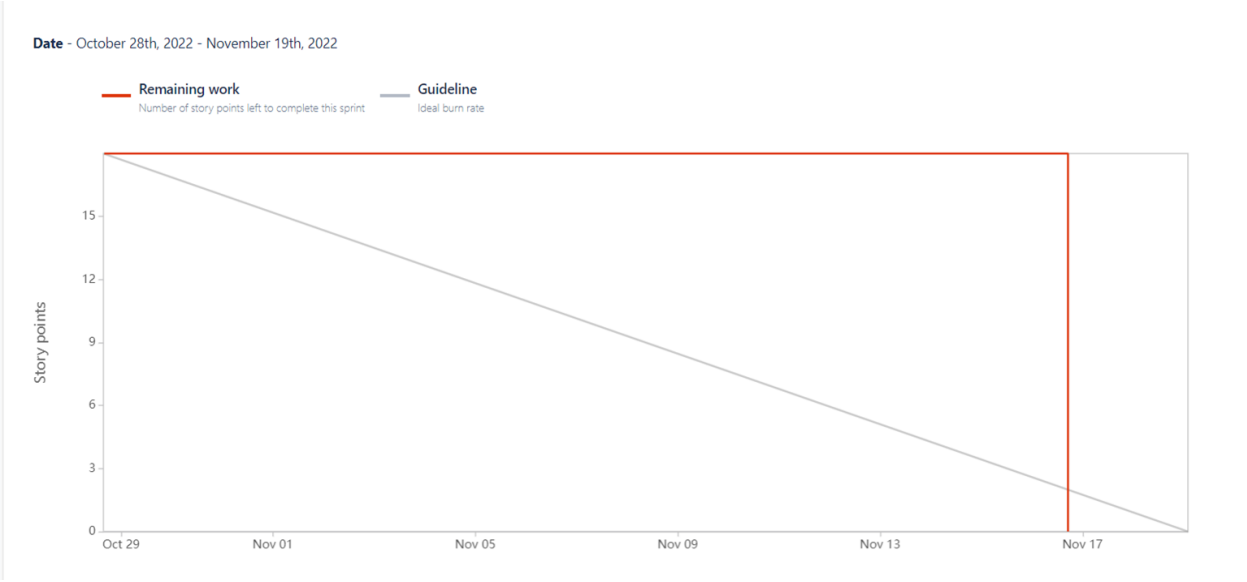

Figure - 11

7. CODING & SOLUTIONING

7.1 FEATURE 1 :

PREMIUM FEATURES :

Demand Est provides a series of exciting ingredients for the premium clients. The user can avail and claim the premium subscription provided in the application. Through the subscription, they're given special access to many features.

The unique features include:

1. Festive Offers and coupon codes
2. Weekly,Monthly,Quarterly,Yearly schemes
3. To-Do list
4. Blog

CODE :

Offers.html :

```
<!DOCTYPE html>
<html>
<head>
   <title>Offers</title>
<style>
   .cards {
   margin-right:25%;
   margin-left:25%;
   margin-top: 5%;
   max-width: 1000px;
   display: grid;
   grid-template-columns: repeat(auto-fill, minmax(225px, 1fr));
   grid-auto-rows: auto;
   gap: 20px;
```

```
    font-family: sans-serif;
    padding-top: 30px;
}
body{
    background: #5ccef4;
    background: -moz-linear-gradient(90deg, #5ccef4 0%, rgba(199,74,233,1) 100%);
    background: -webkit-linear-gradient(90deg, #5ccef4 0%, rgba(199,74,233,1) 100%);
    background: linear-gradient(90deg, #5ccef4 0%, rgba(199,74,233,1) 100%);
    filter:
progid:DXImageTransform.Microsoft.gradient(startColorstr="#e94ca1",endColorstr="#c74ae9",
GradientType=1);
    background-image:url({{url_for('static',filename='register.jpg')}});
}
.cards * {
    box-sizing: border-box;
}
.card__image {
    width: 100%;
    height: 150px;
    object-fit: cover;
    display: block;
    border-top: 2px solid #333333;
    border-right: 2px solid #333333;
    border-left: 2px solid #333333;
}
.card__content {
    line-height: 1.5;
    font-size: 0.9em;
    padding: 15px;
```

```
    background: #fafafa;
    border-right: 2px solid #333333;
    border-left: 2px solid #333333;
}
.card__content > p:first-of-type {
    margin-top: 0;
}

.card__content > p:last-of-type {
    margin-bottom: 0;
}
.card__info {
    padding: 15px;
    display: flex;
    justify-content: space-between;
    align-items: center;
    color: black;
    background: #5ccef4;
    font-size: 0.8em;
    border-bottom: 2px solid #333333;
    border-right: 2px solid #333333;
    border-left: 2px solid #333333;
}
.card__info i {
    font-size: 0.9em;
    margin-right: 8px;
}
.card__link {
```

```
    color: black;
    text-decoration: none;
}
.card__link:hover {
    text-decoration: underline;
}
button{
    width: 100px;
    height: 40px;
    border-radius: 10px;
    background-color: yellow;
    color: black;
    margin-left: 320px;
    font-weight: bolder;
    cursor: pointer;
    margin-top: 50px;
}
a{
    border: #5ccef4;
    color: black;
    font-size: medium;
}
</style>
</head>
<body>
    <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">
    <div class="cards">
      <div class="card">
```

```
      <img class="card__image" src="{{url_for('static', filename='offer.jpeg')}}" height="400px"
width="300px" alt="">
      <div class="card__content">
        <p>
          Upto 30% discount on your predictions !<br>
          Only limited coupons available...Hurry up to  grab yours' soon
        </p>
        <p>
          Use the below code to avail this offer...
        </p>
      </div>
      <div class="card__info">
        <div>
          <i class="material-icons">thumb_up</i>310
        </div>
        <div>
          <a href="./" class="card__link">Offer Code-3446</a>
        </div>
      </div>
    </div>
    <div class="card">
      <img class="card__image" src="{{url_for('static', filename='offer1.jpeg')}}"
height="400px" width="300px" alt="">
      <div class="card__content">
        <p>
          Excited for upcoming festive seasons ??
        </p>
        <p>
          DemandEst provides you exciting offers for this festive period.. <br>
```

```
        Claim your offers using the code below :)
      </p>
    </div>
    <div class="card__info">
      <div>
        <i class="material-icons">thumb_up</i>1,993
      </div>
      <div>
        <a href="./" class="card__link">Offer Code-3447</a>
      </div>
    </div>
  </div>
  <div class="card">
    <img class="card__image" src="{{url_for('static', filename='offer2.png')}}"
height="400px" width="300px" alt="">
    <div class="card__content">
      <p>
        Lucky Cashbacks and redeem vouchers are on your way!
      </p>
      <p>
        Apply and See If you are the lucky winner of the day and luck awaits :)<br>
    Redeem using the code given below …
  </p>
    </div>
    <div class="card__info">
      <div>
        <i class="material-icons">thumb_up</i>887
      </div>
```

```
        <div>
          <a href="./" class="card__link">Offer Code-3448</a>
        </div>
       </div>
      </div>
      <a href="pay">REDEEM NOW !</a>
    </body>
    </html>
```

Pricing.html:

```
<!DOCTYPE html>
<html>
<head>
 <title>Get Your subscriptions here</title>
</head>
<style>
  @import url('https://fonts.googleapis.com/css?family=Montserrat|Open+Sans|Roboto');
*{
 margin:0;
 padding: 0;
 outline: 0;
}
.filter{
 position: absolute;
 left: 0;
 top: 0;
 bottom: 0;
 right: 0;
```

```
 z-index: 1;
background: #5ccef4;
background: -moz-linear-gradient(90deg, #5ccef4 0%, rgba(199,74,233,1) 100%);
background: -webkit-linear-gradient(90deg, #5ccef4 0%, rgba(199,74,233,1) 100%);
background: linear-gradient(90deg, #5ccef4 0%, rgba(199,74,233,1) 100%);
filter:
progid:DXImageTransform.Microsoft.gradient(startColorstr="#e94ca1",endColorstr="#c74ae9",
GradientType=1);
opacity: .7;
}
table{
 position: absolute;
 z-index: 2;
 left: 50%;
 top: 50%;
 transform: translate(-50%,-50%);
 width: 60%;
 border-collapse: collapse;
 border-spacing: 0;
 box-shadow: 0 2px 15px rgba(64,64,64,.7);
 border-radius: 12px 12px 0 0;
 overflow: hidden;
}
td , th{
 padding: 15px 20px;
 text-align: center;
}
th{
 background-color: rgba(233,76,161,1);
```

```
 color: #fafafa;
 font-family: 'Open Sans',Sans-serif;
 font-weight: 200;
 text-transform: uppercase;
}
tr{
 width: 100%;
 background-color: #fafafa;
 font-family: 'Montserrat', sans-serif;
}
tr:nth-child(even){
 background-color: #eeeeee;
}
</style>
<body>
  <div class="filter">
 </div>
 <table >
 <tr>
 <th>S.no</th>
 <th>Scheme</th>
 <th>Cost</th>
 <th>Purchase</th>
 </tr>
  <tr>
 <td>1</td>
 <td>Weekly</td>
 <td>Rs.149</td>
```

```
 <td><a href="pay">BUY</a></td>
</tr>
<tr>
    <td>2</td>
    <td>Mothly</td>
    <td>Rs.549</td>
    <td><a href="pay">BUY</a></td>
   </tr>
   <tr>
    <td>3</td>
    <td>Quarterly</td>
    <td>Rs.749</td>
    <td><a href="pay">BUY</a></td>
   </tr>
   <tr>
    <td>4</td>
    <td>Yearly</td>
    <td>Rs.999</td>
    <td><a href="pay">BUY</a></td>
   </tr>
   <tr></tr>
    <td>5</td>
    <td>Festival scheme</td>
    <td>Rs.499</td>
    <td><a href="pay">BUY</a></td>
   </tr>
  </table>
 </body>
```

```
</html>
```

Todo.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <title>TO-DO LIST</title>
    <style>
        body{
            content:'';
            position:static;
            width:100vw;
            height:100vh;
            background-image:url({{url_for('static',filename='todo.jpg')}});
        }
        header{
            height:60px;
        }
        * {
            margin:0;
            padding:0;
            box-sizing:border-box;
            font-family: sans-serif;
        }
        #head{
            padding:20px;
```

```
    text-align:center;
    color: azure;
    background-color: rgb(14, 15, 15);
    border-radius:50px ;
    font-size: 30px;
    margin-left: 500px;
    margin-top: 50px;
}
main{
    width:500px;
    height: 500px;
    border: 5px rgba(19, 17, 19, 0.85) solid;
    border-radius: 60px;
    margin:40px auto;
    padding: 30px;
    display: flex;
    flex-direction: column;
    align-items: center;
    background: rgba(17, 141, 207, 0.6);
}
ul{
   list-style: none;
   width:100%;
   background-color:rgb(119, 78, 216);
}
li{
    padding: 10px;
    position:relative;
```

```
}
#mytext{
    height:30px;
    margin-right: 10px;
    padding-left: 10px;
    padding-right: 10px;
}
form{
    margin:20px;
}
#addBtn{
    background-color:rgb(129, 81, 241);
    padding:5px;
}
#addBtn:hover{
    background-color: rgb(13, 153, 223);
    cursor:pointer;
}
.close{
    position: absolute;
    top:0;
    right:0;
    padding:10px;
    cursor:pointer;
 }
.close:hover{
    background-color: rgb(13, 153, 223);
 }
```

```
    </style>
</head>
<body>
    <header>
    </header>
    <main>
        <form>
            <input type="text" placeholder="Enter the tasks" id="mytext">
            <span onclick="mylist()" id="addBtn">+</span>
        </form>
        <ul id="myul">

            <li>Go to gym</li>
            <li>Buy Groceries</li>
        </ul>
    </main>
    <script>
        var list= document.getElementsByTagName('li');
        for(var i=0;i<list.length;i++){
            var span=document.createElement("span");
            span.className="close";
            var close=document.createTextNode("-");
            span.appendChild(close);
            list[i].appendChild(span);
        }
        var close = document.getElementsByClassName("close");
        for(i=0;i < close.length;i++){
            close[i].onclick=function(){
```

```
        var d=this.parentElement;
        d.style.display="none";
    }
}
function mylist(){
    var li = document.createElement("li");
    var mytxt = document.getElementById("mytext").value;
    var t=document.createTextNode(mytxt);
    li.appendChild(t);
    if(mytxt==""){
        alert("Input Field is Empty");
    }
    else{
      document.getElementById("myul").appendChild(li);
    }
    var list= document.getElementsByTagName('li');
    for(var i=0;i<list.length;i++){
        var span=document.createElement("span");
        span.className="close";
        var close=document.createTextNode("-");
        span.appendChild(close);
        list[i].appendChild(span);
    }
    var close = document.getElementsByClassName("close")
    for(i=0;i<close.length;i++){
        close[i].onclick=function(){
        var d=this.parentElement;
        d.style.display="none";
```

```
        }
      }
   }
   </script>
</body>
</html>
```

7.2 FEATURE 2:

USER MANUAL :

1. During registration, the application asks for the city code of the user for predicting needs.
2. Since every user may not be aware of their respective city codes , they can have a look on the user manual page .

CODE :

Cities.html :

```
<!DOCTYPE html>
<html>
<head>
   <title>User Manual</title>
</head>
<style>
   *{
   margin: 0;
   padding: 0;
   font-family: arial;
   box-sizing: border-box;
}
body{
   height: 100vh;
```

```
    display: grid;
    place-items: center;
    background: linear-gradient(rgba(0,0,0,0),rgba(0,0,0,0.6));
    background-image:url({{url_for('static',filename='city.png')}});
    background-size: 100% 100%;
    background-repeat: no-repeat;
}
table{
    width: 550px;
    margin-left:600px;
    box-shadow: -1px 12px 12px -6px rgba(0,0,0,0.5);
    border-radius: 10px;
}
table, td, th{
    padding: 10px;
    border: 10px solid lightgray;
    border-radius: 10px;
    border-collapse: collapse;
    text-align: center;
    cursor: pointer;
}
td{
    font-size: 18px;
}
th{
    background-color: blue;
    color: white;
}
tr:nth-child(odd){
```

```
    background-color: lightblue;
}
tr:nth-child(odd):hover{
    background-color: dodgerblue;
    color: white;
    transform: scale(1.5);
    transition: transform 300ms ease-in;
}
tr:nth-child(even){
    background-color:plum;
}
tr:nth-child(even):hover{
    background-color:pink;
    transform: scale(1.5);
    transition: transform 300ms ease-in;
}
a{
    text-decoration: none;
    color:black;
    font-size:19px;
    border:10px solid yellowgreen;
    background-color: yellowgreen;
    border-radius: 10px;
    font-weight: bolder;
}
</style>
<body>
<table>
    <tr>
```

```
    <th>S.No</th>
    <th>Cities</th>
    <th>Region Code</th>
    <th>PIN CODE</th>
    <th>Centre Type</th>
</tr>
<tr>
    <td>1</td>
    <td>Srirangam</td>
    <td>77</td>
    <td>620005</td>
    <td>Type A</td>
</tr>
<tr>
    <td>2</td>
    <td>Manaparai</td>
    <td>34</td>
    <td>621306</td>
    <td>Type B</td>
</tr>
<tr>
    <td>3</td>
    <td>Thiruverumbur</td>
    <td>56</td>
    <td>620011</td>
    <td>Type B</td>
</tr>
<tr>
    <td>4</td>
```

```
    <td>Panjappur</td>
    <td>71</td>
    <td>620012</td>
    <td>Type C</td>
</tr>
<tr>
    <td>5</td>
    <td>Lalgudi</td>
    <td>85</td>
    <td>621601</td>
    <td>Type B</td>
</tr>
<tr>
    <td>6</td>
    <td>Musiri</td>
    <td>56</td>
    <td>621211</td>
    <td>Type B</td>
</tr>
<tr>
    <td>7</td>
    <td>Mannachanallur</td>
    <td>34</td>
    <td>639112</td>
    <td>Type C</td>
</tr>
<tr>
    <td>8</td>
    <td>Kattur</td>
```

```
    <td>77</td>
    <td>610104</td>
    <td>Type A</td>
</tr>
<tr>
    <td>9</td>
    <td>Samayapuram</td>
    <td>56</td>
    <td>621112</td>
    <td>Type A</td>
</tr>
<tr>
    <td>10</td>
    <td>Uraiyur</td>
    <td>34</td>
    <td>620003</td>
    <td>Type B</td>
</tr>
</table>
<p><a href="register">Click here to Go back ..</a></p>
</body>
</html>
```

8. TESTING

8.1 TEST CASES

LoginPage_TC_OO5	Functional	Login page	Verify user is able to log into application with InValid credentials		1.Enter URL(https://shopenzer.com/) and click go 2.Click on My Account dropdown button 3.Enter InValid username/email in Email text box 4.Enter Invalid password in password text box 5.Click on login button	Username: suba@gmail.com password: suba	Application should show 'Incorrect email or password ' validation message.
LoginPage_TC_OO6	Functional	Login Page	Verify user is able to log into application with InValid credentials		1.Enter URL(https://shopenzer.com/) and click go 2.Click on My Account dropdown button 3.Enter InValid username/email in Email text box 4.Enter Invalid password in password text box 5.Click on login button	Username: ram@gmail.com password: ram	Application should show 'Incorrect email or password ' validation message.
LoginPage_TC_OO7	Functional	Login Page	Verify user is able to log into application with InValid credentials		1.Enter URL(https://shopenzer.com/) and click go 2.Click on My Account dropdown button 3.Enter InValid username/email in Email text box 4.Enter Invalid password in password text box 5.Click on login button	Username: ram@gmail.com password: ram	Application should show 'Incorrect email or password ' validation message.

Date	03-Nov-22
Team ID	PNT2022TMID32677
Project Name	Project - DemandEst – AI powere
Maximum Marks	4 marks

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	C
LoginPage_TC_OO1	Functional	Home Page	Verify user is able to see the Login/Signup popup when user clicked on My account button		1.Enter URL and click go 2.Click on My Account dropdown button 3.Verify login/Singup popup displayed or not	http://127.0.0.1:8000	Login/Signup popup should display	Working as expected	Pass	
LoginPage_TC_OO2	UI	Home Page	Verify the UI elements in Login/Signup popup		1.Enter URL and click go 2.Click on My Account dropdown button 3.Verify login/Singup popup with below UI elements: a.email text box b.password text box c.Login button d.New customer? Create account link e.Last password? Recovery password link	http://127.0.0.1:8000	Application should show below UI elements: a.email text box b.password text box c.Login button with orange colour d.New customer? Create account link e.Last password? Recovery password link	Working as expected	Pass	
LoginPage_TC_OO3	Functional	Home page	Verify user is able to log into application with Valid credentials		1.Enter URL(https://shopenzer.com/) and click go 2.Click on My Account dropdown button 3.Enter Valid username/email in Email text box 4.Enter valid password in password text box 5.Click on login button	Username: sujit@gmail.com password: sujit	User should navigate to user account homepage			
LoginPage_TC_OO4	Functional	Login page	Verify user is able to log into application with InValid credentials		1.Enter URL(https://shopenzer.com/) and click go 2.Click on My Account dropdown button 3.Enter InValid username/email in Email text box 4.Enter valid password in password text box 5.Click on login button	Username: Sriidevi4@gmail.com password: sri	Application should show 'Incorrect email or password ' validation message.			
LoginPage_TC_OO4	Functional	Login page	Verify user is able to log into application with InValid credentials		1.Enter URL(https://shopenzer.com/) and click go 2.Click on My Account dropdown button 3.Enter Valid username/email in Email text box 4.Enter Invalid password in password text box 5.Click on login button	Username: vijay@gmail.com password: vijay	Application should show 'Incorrect email or password ' validation message.			
LoginPage_TC_OO5	Functional	Login page	Verify user is able to log into application with InValid credentials		1.Enter URL(https://shopenzer.com/) and click go 2.Click on My Account dropdown button 3.Enter InValid username/email	Username: suba@gmail.com password: suba	Application should show 'Incorrect email or password ' validation message.			

	Test Scenarios
1	Verify user is able to see login page
2	Verify user is able to loginto application or not?
3	Verify user is able to navigate to create your account page?
4	Verify user is able to recovery password
5	Veriify login page elements
	Search
1	Verify user is able to search by entering keywords in search box
2	Verify user is able to see suggestions based on keyword entered in search box
3	Verify user is able to see related auto suggestions displaying based on keyword entered in search box
4	Verify user is able to see no matches found message when no results are matching with entered keyword
5	Verify user is able to see seach detailed page when nothing entered in textbox

8.2 USER ACCEPTING TESTING

1. PurposeofDocument

The purpose of this document is to briefly explain thetestcoverageandopenissuesofthe [ProductName] project at the time of the release to User Acceptance Testing (UAT).

2. DefectAnalysis

Thisreportshowsthenumberofresolvedor closed bugs at each severity level, and how they were resolved

Resolution	Severity1	Severity2	Severity3	Severity4	Subtotal
By Design	10	4	2	3	20
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	37
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won'tFix	0	5	2	1	8
Totals	24	14	13	26	77

3. TestCaseAnalysis

Thisreportshowsthenumberoftestcasesthathavepassed,failed,anduntested

Section	TotalCases	Not Tested	Fail	Pass
Login	7	0	0	7
Final Report Output	4	0	0	4
Premium user	6	0	0	6
Prediction	12	0	0	12

`

9. RESULTS

Thus, We have trained our machine learning model in IBM Watson studio and have successfully built an web application for predicting the food demand for some next 10 weeks. The app can be utilized both by short scale and large scale predictions and thus prevent maximum amount of food wastage also helping the vendors to minimize their expenditures.

10. ADVANTAGES & DISADVANTAGES

Advantages:

1. Avoids exploitation of perishable food.
2. Easy to use.
3. Can decrease cost.

Disadvantages:

The output obtained may not be precise.

Resource intensive.

Sudden increase in demand cannot be predicted.

11. CONCLUSION

The main objective behind this project is to reduce food wastage. The availability of the food items makes the society better. Demand forecasting is the activity of estimating the quantity of a product or service that consumers will purchase and is performed through this project. The replenishment of majority of raw materials is done on weekly basis and since the raw material is perishable, the procurement planning is of utmost importance.

Secondly, staffing of the centers is also one area wherein accurate demand forecasts are helpful. The task is to predict the demand for the next 10 weeks, and is achieved. Demand forecasts is, with no doubt, the basis for developing an efficient supply chain. The supply chain planning and control depends of accurate estimates of the volume of products and services to be processed to satisfy customer's needs.

12. FUTURE SCOPE

1. Working on the frontend to make the framework more dynamic.
2. In the future, we also plan to improve forecasting accuracy and research on the efficiency of store management.
3. In order to ensure the correctness of the forecasts, we have planned to conduct periodic surveys.

13. APPENDIX

SOURCE CODE

main.py:

```
import pandas as pd

import numpy as np

import pickle

import os

import ibm

from flask import Flask,request, render_template,session,redirect,flash

import requests

import ibm_db

app=Flask(__name__,template_folder="templates")

app.secret_key = "super secret key"

#for db connection

dsn_hostname = "ba99a9e6-d59e-4883-8fc0-
d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud" # e.g.: "54a2f15b-5c0f-46df-
8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud"

dsn_uid = "ljx83321"        # e.g. "abc12345"

dsn_pwd = "u7Mi2uux6vXQaKhT"      # e.g. "7dBZ3wWt9XN6$o0J"

dsn_driver = "{IBMDB2CL1}"

dsn_database = "bludb"           # e.g. "BLUDB"

dsn_port = "31321"               # e.g. "32733"

dsn_protocol = "TCPIP"           # i.e. "TCPIP"

dsn_security = "SSL"             #i.e. "SSL

dsn_cert="DigiCertGlobalRootCA.crt"

dsn = (

    "DRIVER={0};"

    "DATABASE={1};"
```

```
    "HOSTNAME={2};"
    "PORT={3};"
    "PROTOCOL={4};"
    "UID={5};"
    "PWD={6};"
    "SECURITY={7};"
    "SSLServerCertificate={8};").format(dsn_driver, dsn_database, dsn_hostname, dsn_port,
dsn_protocol, dsn_uid, dsn_pwd,dsn_security,dsn_cert)
conn = 0
try:
    conn =ibm_db.connect(dsn, "", "")
    print("Connected to database: ", dsn_database, "as user: ", dsn_uid, "on host: ", dsn_hostname)
except:
    print("Unable to connect: ", ibm_db.conn_errormsg())
print(dsn)

#ibm cloud api
API_KEY = "RQRG9kMruxQGPGPQmzRtxPBBK_6hO91hjt6lDcahLizz"
token_response = requests.post('https://iam.cloud.ibm.com/identity/token',
                               data={"apikey": API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-
type:apikey'})
mltoken = token_response.json()["access_token"]

header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}

#for login db table
@app.route('/', methods=['GET'])
def index():
    return render_template('index.html')
```

```
@app.route('/login',methods=['GET','POST'])
def login():
    if request.method=='POST':
        email=request.form['email']
        password=request.form['password']
        qry="select * from  CUSTOMER where email=? AND password=?"
        stmt=ibm_db.prepare(conn,qry)
        ibm_db.bind_param(stmt,1,email)
        ibm_db.bind_param(stmt,2,password)
        ibm_db.execute(stmt)
        resp=ibm_db.fetch_assoc(stmt)
        print("resp - ",resp)
        if resp:
            return render_template("page.html")
        else:
            return render_template('index.html')
    return render_template('index.html')


#for reg db table
@app.route('/register',methods=['GET','POST'])
def register():
    if request.method=='POST':
        user=request.form['name']
        password=request.form['password']
        email=request.form['email']
        shopid=request.form['shopid']
```

```
        phone=request.form['phone']
        cityid=request.form['cityid']
        qry="insert into
CUSTOMER(user,password,email,shop_id,phone,city_id)values(?,?,?,?,?,?)"
        stmt=ibm_db.prepare(conn,qry)
        ibm_db.bind_param(stmt,1,user)
        ibm_db.bind_param(stmt,2,password)
        ibm_db.bind_param(stmt,3,email)
        ibm_db.bind_param(stmt,4,shopid)
        ibm_db.bind_param(stmt,5,phone)
        ibm_db.bind_param(stmt,6,cityid)
        resp=ibm_db.execute(stmt)
        print(resp)
        return render_template('index.html')
    return render_template('register.html')
#routings
@app.route('/home',methods=['GET','POST'])
def home():
    return render_template('home.html')

@app.route('/payment',methods=['GET','POST'])
def payment():
    return render_template('payment.html')

@app.route('/cities',methods=['GET','POST'])
def cities():
    return render_template('cities.html')
```

```
@app.route('/prem',methods=['GET','POST'])
def prem():
    return render_template('prem.html')

@app.route('/page',methods=['GET','POST'])
def page():
    return render_template('page.html')

@app.route('/pricing',methods=['GET','POST'])
def pricing():
    return render_template('pricing.html')

@app.route('/todo',methods=['GET','POST'])
def todo():
    return render_template('todo.html')

@app.route('/offer',methods=['GET','POST'])
def offer ():
    return render_template('offer.html')

@app.route('/contact',methods=['GET','POST'])
def contact ():
    return render_template('contact.html')
@app.route('/pay',methods=['GET','POST'])
def pay ():
    return render_template('payment.html')
def page():
    return render_template('predict.html')
```

```
@app.route('/predict', methods=['GET', 'POST'])
def predict():
    if request.method=='GET':
        return render_template('predict.html')
    print("[INFO] loading model...")
    model = pickle.load(open('fdemand.pkl', 'rb'))
    input_features = [float(x) for x in request.form.values()]
    features_value = [np.array(input_features)]
    print(features_value)
    features_name = ['homepage_featured', 'emailer_for_promotion', 'op_area', 'cuisine',
      'city_code', 'region_code', 'category']
    prediction = model.predict(features_value)
    output=(prediction[0])//2
    print(output)
    return render_template('predict.html', prediction_text=output)
if __name__ == '__main__':
    app.run(host='0.0.0.0',port=8000,debug=False)
```

ibm.py:

```
import requests
# NOTE: you must manually set API_KEY below using information retrieved from your IBM Cloud account.
API_KEY = "RQRG9kMruxQGPGPQmzRtxPBBK_6hO91hjt6lDcahLizz"
token_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={"apikey": API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})
mltoken = token_response.json()["access_token"]
header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}

# NOTE: manually define and pass the array(s) of values to be scored in the next line
```

```
payload_scoring = {"input_data":
 [{"field": [['homepage_featured', 'emailer_for_promotion', 'op_area', 'cuisine',
      'city_code', 'region_code', 'category']],
"values": [[1.,1.,3.,0.,480.,56.,1.]]}]}
response_scoring = requests.post('https://us-
south.ml.cloud.ibm.com/ml/v4/deployments/38e81a58-4f9b-4e2b-a15c-
9fe565df5548/predictions?version=2022-11-17', json=payload_scoring,
headers={'Authorization': 'Bearer ' + mltoken})
print("Scoring response")
print(response_scoring.json())
predictions =response_scoring.json()
print(predictions)
print('Final Prediction Result',predictions['predictions'][0]['values'][0][0])
```

index.html :

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Login</title>
   <style>
   h1{
       color:white;
       opacity: 0.5;
       font-size:70px;
       font-family:'Lucida Sans', 'Lucida Sans Regular', 'Lucida Grande', 'Lucida Sans Unicode',
Geneva, Verdana, sans-serif ;
```

```
    font-style:bold ,italic;
    padding-right: 10 %;
    down:20%
}
    body{
        margin:50px;
        padding:20px;
        background-size:100%;
        background-repeat: no-repeat;
        font-family: sans-serif;
        background-image:url({{url_for('static',filename='food1.png')}});
    }
    .loginbox{
        position: absolute;
        top:60%;
        right:280px;
        transform:translate(-50%,-50%);
        width:350px;
        height:420px;
        padding:80px 40px;
        box-sizing:border-box;
        background:lightseagreen;
        color:#fff;

    }
    h2{
        margin:0;
        padding:0 0 15px;
```

```
    color:yellow;
    font-size:32px;
    text-align: center;
    font-family: 'Gill Sans', 'Gill Sans MT', Calibri, 'Trebuchet MS', sans-serif;
}
.loginbox p{
    padding:0;
    margin:0;
    font-weight: bold;
    color:#fff;
}
.loginbox input{
    width:100%;
    margin-bottom: 20px;

}
.loginbox input[type="text"], .loginbox input[type="password"]
{
    border:none;
    border-bottom: 1px solid #fff;
    background: transparent;
    outline: none;
    height:40px;
    color:#fff;
    font-size:16px;
}
#sign
{
```

```
    border:none;
    outline: none;
    height: 30px;
    color:#fff;
    font-size: 16px;
    background: rgb(6, 85, 66);
    cursor:pointer;
    border-radius: 20px;
    position: absolute;
    top:55%;
    left:10%;
    right: 10%;
    padding:20px 80px;
    padding-left:70px;
    margin-top:90px;
    padding-bottom:30px;
    box-sizing:border-box;
}
#sign1
{
    color:yellow;
    font-size: 16px;
    cursor:pointer;
    font-size:20px;
    font-family: 'Gill Sans', 'Gill Sans MT', Calibri, 'Trebuchet MS', sans-serif;
    position: absolute;
    padding-left:40px;
    margin-top:80px;
```

```
            padding:10px 50px;
            box-sizing:border-box;
        }
        a{
            color: black;
            text-decoration: none;
        }
        .user{
            width:100px;
            height: 100px;
            overflow: hidden;
            position: absolute;
            top:calc(-100px/2);
            left: calc(50% - 50px);
            border-radius: 50px;
        }
        #inp{
            color:rgb(236, 75, 75);
        }
       #sign_btn{
        transition-duration: 0.4s;
        color:white;
        }
    </style>
</head>
<body>
    <h1>FOOD <br> DEMAND <br> FORECASTING</h1>
    <!-- <div class="col">
```

```
        <h3 class="title"></h3>
        <img src="{{url_for('static', filename='user.png')}}" width="100px" height="100px"/>
    </div> -->
<div class="loginbox" >
    <h2>WELCOME</h2>
    <form action="/login" method="post">
        <p>EMAIL</p>
        <input type="text" id="inp" name="email" placeholder="Enter the email"
onclick="checkemail()">
        <div id="d1"></div>
        <p>PASSWORD</p>
        <input type="password" name="password" placeholder="Enter the password">
        <input type="submit" value=" SIGN IN  " name="sign_btn"></input>
        <div>
            <a href="register"> NEW USER ?</a>
        </div>
        <br>
    </form>
</div>
</body>
</html>
```

home.html :

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Document</title>
<style>
   body{
      margin:0;
      padding:0;
      min-height:20vh;
      padding-top: 100px;
      background-image:url({{url_for('static',filename='image.jpg')}});
      background-size:100%;
      background-position: center;
      display:flex;
      justify-content:center;
      font-family:consolas;
   }
   .container{
      width:1000px;
      position: relative;
      display: flex;
      justify-content: space-between;
   }
   .container .card{
      position: relative;
      cursor:pointer;
   }
   .container .card .face{
      width:300px;
      height:200px;
```

```
    transition: 0.5s;
}
.container .card .face.face1{
    position: relative;
    background:darkolivegreen;
    display:flex;
    justify-content: center;
    align-items: center;
    z-index:1;
    transform: translateY(100px);
}
.container .card:hover .face.face1{
    background:darkkhaki;
    transform: translateY(0);
}
.container .card .face.face1 .content{
    opacity: 0.2;
    transition: 0.5s;
}
.container .card.hover .face.face1 .content{
    opacity: 1;
}
.container .card .face.face1 .content img{
    max-width: 100px;
}
.container .card .face.face1 .content h2{
    margin:10px 0 0;
    padding: 0;
```

```
    color:rgb(255, 255, 255);
    text-align: center;

}
.container .card .face.face2{
    position: relative;
    background:#fff;
    display: flex;
    justify-content: center;
    align-items: center;
    padding: 20px;
    box-sizing: border-box;
    box-shadow: 0 20px 50px rgba(0,0,0,0.8);
    transform: translateY(-100px);
}
.container .card:hover .face.face2{
    transform: translateY(0);
}
.container .card .face.face2 .content p{
    margin: 0;
    padding: 0;
}
.container .card .face.face2 .content a{
    margin: 15px 0 0;
    display: inline-block;
    text-decoration: none;
    font-weight: 900;
    color:black;
```

```
        padding: 5px;
        border: 1px solid blueviolet;
    }
    .container .card .face.face2 .content a:hover{
        background: #333;
        color:burlywood;
    }
    h3{
        background-color: beige;
    }
  </style>
</head>
<body>
  <div class="container">
    <div class="card">
      <div class="face face1">
        <div class="content">
          <h2>PREDICT</h2>
        </div>
      </div>
      <div class="face face2">
        <div class="content">
          <p>
            Start your predictions here ! You can discover new predictions and cool schemes over here.
          </p>
          <a href="predict">PREDICT</a>
        </div>
```

```
        </div>
          <div class="card">
      <div class="face face1">
        <div class="content">
          <h2>PREMIUM USER</h2>
        </div>
      </div>
      <div class="face face2">
        <div class="content">
          <p>
            Welcome to our premium site ! Make predictions easy on the go .
          </p>
          <a href="payment">SUBSCRIBE</a>
        </div>
      </div>
    </div>
    <div class="card">
      <div class="face face1">
        <div class="content">
          <h2>USER MANUAL</h2>
        </div>
      </div>
      <div class="face face2">
        <div class="content">
          <p>
            Contact details and customer reviews are displayed over here. Just for future references!
          </p>
```

```
            <a href="cities">GO TO MANUAL</a>
        </div>
    </div>
    </div>
</body>
</html>
```

predict.html :

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<style>
    @import
url('https://fonts.googleapis.com/css2?family=Poppins:wght@100;300;400;500;600&display=sw
ap');
*{
  font-family: 'Poppins', sans-serif;
  margin:0; padding:0;
  box-sizing: border-box;
  outline: none; border:none;
  text-transform: capitalize;
  transition: all .2s linear;
}
.container{
  display: flex;
  justify-content: center;
```

```
  align-items: center;
  padding:25px;
  min-height: 100vh;
  background: -moz-linear-gradient(90deg, #5ccef4 0%, rgba(199,74,233,1) 100%);
background: -webkit-linear-gradient(90deg, #5ccef4 0%, rgba(199,74,233,1) 100%);
background: linear-gradient(90deg, #5ccef4 0%, rgba(199,74,233,1) 100%);
filter:
progid:DXImageTransform.Microsoft.gradient(startColorstr="#e94ca1",endColorstr="#c74a9",
GradientType=1);
}
img {
  margin-left: 60%;
  margin-top: -1500px;
}
.container form{
  padding:20px;
  width:700px;
  background: #fff;
  box-shadow: 0 5px 10px rgba(0,0,0,.1);
}
.container form .row{
  display: flex;
  flex-wrap: wrap;
  gap:15px;
}
.container form .row .col{
  flex:1 1 250px;
}
.container form .row .col .title{
```

```
  font-size: 20px;
  color:#333;
  padding-bottom: 5px;
  text-transform: uppercase;
}
.container form .row .col .inputBox{
  margin:15px 0;
}
.container form .row .col .inputBox span{
  margin-bottom: 10px;
  display: block;
}

.container form .row .col .inputBox input{
  width: 100%;
  border:1px solid #ccc;
  padding:10px 15px;
  font-size: 15px;
  text-transform: none;
}
.container form .row .col .inputBox input:focus{
  border:1px solid #000;
}
.container form .row .col .flex{
  display: flex;
  gap:15px;
}
.container form .row .col .flex .inputBox{
```

```
  margin-top: 5px;
}
.container form .row .col .inputBox img{
  height: 34px;
  margin-top: 5px;
  filter: drop-shadow(0 0 1px #000);
}
.container form .submit-btn{
  width: 100%;
  padding:12px;
  font-size: 17px;
  background: #2ecc71;
  color:black;
  margin-top: 5px;
  cursor: pointer;
}
.container form .submit-btn:hover{
  background: #2ecc71;
}
.container form .row .col .inputBox1 select{
  border:1px solid #000;
}
</style>
<body>
<div class="container">

    <form action="{{ url_for('predict') }}" method="POST">
        <div class="row">
```

```
<div class="col">
   <h3 class="title">PREDICTION FORM</h3>
            <div class="inputBox">
      <span>Home_Page Featured </span>
      <select id="homepage_featured" name="homepage_featured">
         <option value="">homepage_featured</option>
           <option value="0">No</option>
           <option value="1">Yes</option>
          </select>
   </div>
   <div class="inputBox">
      <span>Email For Promotion </span>
      <select id="emailer_for_promotion" name="emailer_for_promotion">
         <option value="">emailer_for_promotion</option>
           <option value="0">No</option>
           <option value="1">Yes</option>
          </select>
   </div>
   <div class="inputBox1">
      <span>Select Cuisine</span><br>
      <select name="cuisine" id="cuisine">
         <option value="#">Select</option>
         <option value="0">INDIAN</option>
         <option value="1">CHINESE</option>
         <option value="2">CONTINENTAL</option>
         <option value="3">ITALIAN</option>
      </select><br>
   </div>
```

```
            <div class="col">
              <h3 class="title"></h3>
              <img src="{{url_for('static', filename='man.png')}}" width="400px"
height="230px"/>
            </div>
              <div class="inputBox">
                 <span> Operational Area</span>
                 <input type="number" placeholder="(2-7)">
              </div>
              <div class="inputBox">
                 <span> Region Code :</span>
                 <input type="number" placeholder="123">
              </div>
              <div class="inputBox">
                 <span> City Code :</span>
                 <input type="number" placeholder="77">
              </div>
              </div>
              <div class="inputBox">
                 <span>Category</span>
                 <select id="category" name="category">
                    <option value="">Choose Category</option>
                       <option value="0">Beverages</option>
                       <option value="1">Biryani</option>
                       <option value="2">Desert</option>
                       <option value="3">Extras</option>
                       <option value="4">Fish</option>
                       <option value="5">Other Snacks</option>
```

```
                    <option value="6">Pasta</option>
                    <option value="7">Pizza</option>
                    <option value="8">Rice Bowl</option>
                    <option value="9">Salad</option>
                    <option value="10">Sandwich</option>
                    <option value="11">Seafood</option>
                    <option value="12">Soup</option>
                    <option value="13">Starters</option>
                  </select>
            </div>
            <input type="submit" value="PREDICT" class="submit-btn">
            <h1 class="predict">Number of orders: {{ prediction_text }}</h1>
    </form>
</div>
</body>
</html>
```

GITHUB & PROJECT DEMO LINK

https://github.com/IBM-EPBL/IBM-Project-16682-1664017135

https://www.youtube.com/watch?v=tah63yvoFQ4