**SKILL DEVELOPMENT PROJECT**

**BATCH-26**

**MP-1:**

**Algorithm implementation of the optimization component in the design phase:**

**Introduction:**

Our project **THE MEALMATE** is an e-commerce website which delivers original, step-by-step recipes and fresh ingredients to customers. Each delivery you learn to cook inventive new dishes with seasonal ingredients. By letting us source these hard-to-find ingredients for you, you'll get food that is fresher and cheaper than you can get at your local supermarket, and there's no waste because we only send you what you need for each recipe.

So, the main component of optimization for an e-commerce website is **PRICING** of the product we are selling. The cost of the product must be in such a way that it would provide an optimal profit and also the consumer price considering all the pricing aspects from the making of the product to delivering it to the consumer.

**Below is the complete algorithm including all the steps taken to optimize our prices :**

**ALGORITHM:**

**STEP 1:** Form two different types of sheets in an excel for implementation of the unit test cases of every recipe.

**STEP 2:** Give two different names to the sheets being:

**RECIPE COST CALCULATION SHEET:**

**STEP 2.1:** Put the recipe name as the heading and start the optimized cost calculation.

**STEP 2.2:** Put up columns named

* Ingredient
* Total purchase quantity
* Cost of each ingredient
* Quantity needed for each recipe

**STEP 2.3:** FOR CALCULATING cost of each ingredient

Here, We use the concept of linear programming by considering the all the ingredients a constrains. We need to calculate the cost for each ingredient.

So, we use the formula:

**(Quantity needed for recipe(g))/(Purchase quantity\*cost of the ingredient)**

**STEP 2.4:** FOR CALCULATING Total optimized cost of the recipe by adding up all the constrains.

**Total = sum(all calculated cost of the ingredient)**

**STEP 2.5:** FOR CALCULATING according to the number of servings:

**Servings = Total/n**

Here, n is the number of servings the customer asks for.

**STEP 2.6:** FOR CALCULATING cost per 10

**Cost per 10 = Total\*10**

**STEP 3:** Whenever a new recipe is added the step 2 repeats on whole.

**STEP 4:** Now, after making the recipe cost sheets, we make the **PROFIT MARGIN AND FINAL COAST SHEET** for optimized final maximum profit ad optimized cost of all the recipes.

STEP 4.1: Put up columns named

* Menu item
* Quantity sold
* Food cost
* Menu price
* Food cost percentage
* Profit margin

**STEP 4.2:** The quantity sold column is filled by the number of dishes sold.

**STEP 4.3:** Get the food cost from the “total” constrain in the recipe cost sheet.

**STEP 4.4:** Set up the menu price by adding revenue of 18% as GST and also add the shipping charges.

**Menu Price= Food cost + 18% gst + shipping charges**

**STEP 4.5:** CALCULATING FOOD COST PERCENTAGE:

**Food cost percentage = Food cost/revenue cost(18% GST)**

**STEP 4.6**: CALCULATING **THE OPTIMIZED PROFIT MARGIN:**

**Profit margin** gauges the degree to which a company or a business activity makes money, essentially by dividing income by revenues. Expressed as a percentage, **profit margin** indicates how many cents of **profit** has been generated for each dollar of sale.

**Profit margin = (Menu price \* Quantity sold) – (Quantity sold \* Menu price \***

**Food cost percentage)**

**STEP 5:** Repeat the STEP 4 for each new recipe which is added.

**STEP 6:** By the above overall process the **OPTIMIZED PRICE** of the recipe id obtained.

**STEP 7:** Now according to whatever price obtained, we implement the price seen by the website users.

**Algorithm implementation for the SEARCH BAR:**

In our project we implemented the search bar in the “Kitchen tools part”. If the customer wants to buy a kitchen tool he/she can type the name and get the result if the product is available or not. This search bar uses the following algorithm:

The type script for this is:

Search(){

if(this.name != ""){

this.filterList = this.productsList.filter(res=>{

return res.name.toLocaleLowerCase().match(this.name.toLocaleLowerCase());

});

this.OnChange();

}

else if(this.name == ""){

this.ngOnInit();

}

}

Here the **ALGORITHM** goes as,

**STEP 1:** If the filter list i.e the name typed by the costumer is equal to a name present in the Product list then

**STEP 1.1:** Match the lower case alphabets and the uppercase

**STEP 1.2:** Return the product as the result.

**STEP 2:** If a costumer type “k” everything optimally related to k, irrelevant of the case, is shown as the result.

For example: Typing “k” turns out the optimal results as

* Knives
* Kitchen wear etc.

**STEP 3:** Else if the product is not found after the name typed by the customer then it returns and empty screen which is the null value.

**STEP 4:** Else it returns all the products available in the website as a result.

**STEP 5:** By following the above four steps the costumers can navigate the necessary product.