



**M.KUMARASAMY**  
**COLLEGE OF ENGINEERING**  
NAAC Accredited Autonomous Institution  
Approved by AICTE & Affiliated to Anna University  
ISO 9001:2015 Certified Institution  
Thalavapalayam, Karur – 639 113.



A Project Report

on

## **E-COMMERCE PRODUCT RATING SYSTEM**

Submitted in partial fulfilment of requirements for the award of the course

of

**CGA1121 – DATA STRUCTURES**

Under the guidance of

**Dr.B.PADMINI DEVI M.E., Ph.D.,**

**Associate Professor/CSE**

Submitted By

**DHARSHINI.K (927623BAD023)**

**DEPARTMENT OF FRESHMAN ENGINEERING**

**M.KUMARASAMY COLLEGE OF ENGINEERING**

(Autonomous)

**KARUR – 639 113**

**MAY 2024**



**M.KUMARASAMY**  
**COLLEGE OF ENGINEERING**  
NAAC Accredited Autonomous Institution  
Approved by AICTE & Affiliated to Anna University  
ISO 9001:2015 Certified Institution  
Thalavapalayam, Karur - 639 113.



**M. KUMARASAMY COLLEGE OF ENGINEERING**  
**(Autonomous Institution affiliated to Anna University, Chennai)**

**KARUR – 639 113**

**BONAFIDE CERTIFICATE**

Certified that this project report on **“E-COMMERCE PRODUCT RATING SYSTEM”** is the bonafide work of **DHARSHINI K (927623BAD023)** who carried out the project work during the academic year 2023- 2024 under my supervision.

Signature

**Dr.B.PADMINI DEVI M.E.,Ph.D.,**  
**SUPERVISOR,**

Department of Computer Science  
and Engineering,

M. Kumarasamy College of Engineering,  
Thalavapalayam, Karur -639 113.

Signature

**Dr. K.CHITIRAKALA, M.Sc., M.Phil.,Ph.D.,**  
**HEAD OF THE DEPARTMENT,**

Department of Freshman Engineering,

M. Kumarasamy College of Engineering,  
Thalavapalayam, Karur -639 113.



## **DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**

### **VISION OF THE INSTITUTION**

To emerge as a leader among the top institutions in the field of technical education

### **MISSION OF THE INSTITUTION**

- Produce smart technocrats with empirical knowledge who can surmount the global challenges
- Create a diverse, fully-engaged, learner-centric campus environment to provide quality education to the students
- Maintain mutually beneficial partnerships with our alumni, industry, and Professional associations

### **VISION OF THE DEPARTMENT**

To produce competent industry relevant education, skillful research, technical and innovative computer science professionals acquaintance with managerial skills, human and social values.

### **MISSION OF THE DEPARTMENT**

- To impart technical knowledge through innovative teaching, research, and consultancy.
- To develop and to promote student ability thereby to compete globally through excellence in education.
- To facilitate the development of academic-industry Collaboration.
- To produce competent engineers with professional ethics, technical competence and a spirit of innovation and managerial skills.

## **PROGRAM EDUCATIONAL OBJECTIVES (PEOs)**

- PEO 1:** To acquire technical knowledge and proficiency required for the employment and higher education in the contemporary areas of computer science or management studies.
- PEO 2:** To apply their competency in design and development of innovative solutions for real-world problems.
- PEO 3:** To demonstrate leadership qualities with high ethical standards and collaborated with other industries for the socio-economical growth of the country.

## **PROGRAM OUTCOMES (POs)**

Engineering students will be able to:

**1.Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

**2.Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

**3.Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

**4.Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

**5.Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

**6.The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

**7.Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

**8.Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

**9.Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

**10.Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

**11.Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

**12.Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

### **PROGRAM SPECIFIC OUTCOMES (PSOs)**

1. **PSO1:** Ability to apply the analytical and business skills to provide sustainable solutions as an engineer/researcher for the real-time applications using Machine Learning, Internet of Things and Data analytics.
2. **PSO2:** Ability to practice ethical and human values with soft-skills qualities in computer science and business disciplines to emerge as an entrepreneur for the growth and development of the society

## **ABSTRACT**

The e-commerce product rating system is designed to provide users with a platform to rate and review products online. It allows customers to share their experiences with products, aiding other potential buyers in making informed decisions. The system will feature a user-friendly interface for submitting and viewing ratings, as well as robust back end functionality to manage and display ratings effectively. It aims to implement a singly linked list data structure within an e-commerce product rating system. This product rating system allows users to rate products and provides functionalities for managing and analyzing these ratings efficiently. By utilizing a singly linked list, this ensures optimized memory usage and streamlined operations for adding, removing, and traversing product ratings.

.



## ABSTRACT WITH POs AND PSOs MAPPING

ABSTRACT	POs MAPPED	PSOs MAPPED
The e-commerce product rating system is designed to provide users with a platform to rate and review products online. It allows customers to share their experiences with products, aiding other potential buyers in making informed decisions. The system will feature a user-friendly interface for submitting and viewing ratings, as well as robust backend functionality to manage and display ratings effectively. It aims to implement a singly linked list data structure within an e-commerce product rating system. This product rating system allows users to rate products and provides functionalities for managing and analyzing these ratings efficiently. By utilizing a singly linked list, this ensures optimized memory usage and streamlined operations for adding, removing, and traversing product ratings.	<b>PO1(2)</b> <b>PO2(3)</b> <b>PO3(2)</b> <b>PO4(2)</b> <b>PO6(1)</b> <b>PO7(3)</b> <b>PO8(2)</b> <b>PO9(3)</b> <b>PO10(3)</b> <b>PO11(3)</b> <b>PO12(2)</b>	<b>PSO1(3)</b> <b>PSO2(2)</b>

Note: 1- Low, 2-Medium, 3- High

**SUPERVISOR**

**HEAD OF THE DEPARTMENT**



## TABLE OF CONTENTS

CHAPTER	TITLE	PAGE
No.		No.
<b>1</b>	<b>Introduction</b>	<b>1</b>
	1.1 Introduction	1
	1.2 Objective	1
	1.3 Data Structure Choice	2
<b>2</b>	<b>Project Methodology</b>	<b>3</b>
	2.1 Project methodology	3
	2.2 Block Diagram	3
<b>3</b>	<b>Modules</b>	<b>4</b>
	3.1 Module 1	4
	3.2 Module 2	4
	3.3 Module 3	5
<b>4</b>	<b>Results and Discussion</b>	<b>8</b>
<b>5</b>	<b>Conclusion</b>	<b>9</b>
	<b>References</b>	<b>10</b>
	<b>Appendix</b>	<b>11</b>

# CHAPTER 1

## INTRODUCTION

### 1.1 Introduction

In today's digital era, E-commerce platforms play a pivotal role in the retail industry. However, with the vast array of products available online, consumers often face difficulty in determining the quality and reliability of items. The E-commerce Product Rating System addresses this challenge by presenting users with comprehensive product ratings and reviews. By leveraging the wisdom of the crowd, this system empowers consumers to make informed purchasing decisions. Each product in the system has a corresponding singly linked list to store its ratings. Each node in the linked list contains information about a single rating, including the rating value (e.g., 1 to 5 stars), the user who submitted the rating, and optionally a timestamp. The head of the linked list points to the first rating submitted for the product, and subsequent ratings are added to the end of the list. This system allows customers to provide feedback on the products they have purchased, typically in the form of star ratings and written reviews. These star ratings play a pivotal role in influencing potential buyers' decision, enhancing the shopping experience and providing valuable insights for sellers.

### 1.2 Objective

Our objective is to make the online shopping easier where we can select the quality products using the given ratings. It makes the shopping in online easier. We use a singly linked list which is used to give the ratings in an easier way. The main aim of this product rating system is to provide customer satisfaction, to boost sales and to improve quality.

### **1.3 Data Structure Choice**

#### **SINGLY LINKED LIST:**

It is a linear collection of elements, called nodes, where each node contains two fields: one to store the data and another to store a reference (or link) to the next node in the sequence. The list starts with a head node, and each node points to the next node in the sequence, ending with a node that points to null, indicating the end of the list. Singly linked lists are used in various applications such as dynamic memory allocation, implementing stacks and queues, and navigating through a collection of elements. The operation can be done through singly linked list are insertion, deletion and traversal. the advantages of the singly linked list are it easily grows and shrinks as needed, adding or removing elements is efficient, especially at the beginning of the list. its disadvantages are accessing elements required traversal from the head, it requires additional memory for storing pointers. These are foundational structures in computer science. It is used to implement other complex data structures like stacks, queues and graphs. Their simplicity and flexibility make them a crucial topic in algorithms and data structure courses.

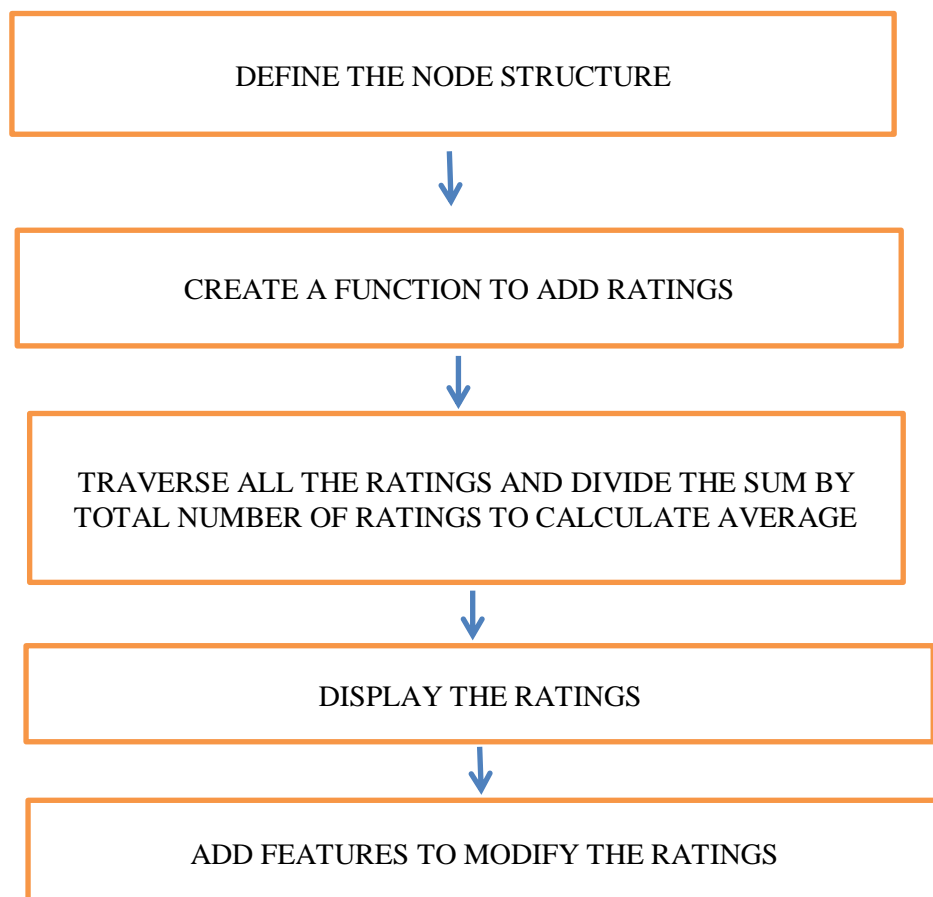
## CHAPTER 2

### PROJECT METHODOLOGY

#### 2.1 Singly Linked List:

It is a linear collection of elements, called nodes, where each node contains two fields: one to store the data and another to store a reference (or link) to the next node in the sequence. The list starts with a head node, and each node points to the next node in the sequence, ending with a node that points to null, indicating the end of the list. Singly linked lists are used in various applications such as dynamic memory allocation, implementing stacks and queues, and navigating through a collection of elements.

#### 2.2 Block Diagram



#### Architecture Diagram

## **CHAPTER 3**

### **MODULES**

#### **3.1 Modules:**

##### **3.1.1. Node Creation**

This module is responsible for setting up the infrastructure required for product rating. It involves creating nodes where ratings and reviews can be stored and managed. It holds the rating value and a pointer to next node. For example, in an e-commerce platform, products might have attributes like price, description, and category. In a database-driven system, this would typically involve creating a table (e.g., Products) with fields such as `product_id`, name, and description

##### **3.1.2. Add Rating**

The Add Rating module allows users to submit ratings for products. This module captures user input, validates it, and then stores it within the appropriate product node. This module allows users to submit their ratings for a particular entity (node). Users might rate entities based on predefined criteria (e.g., 1 to 5 stars). Upon user input (e.g., via a form submission), the system would store the rating along with associated metadata (e.g., user ID, timestamp, and rated entity ID) in a ratings table.

##### **3.1.3. Calculate Rating**

The Calculate Rating module aggregates individual ratings to compute an overall rating for each product. This includes calculating averages, handling different rating metrics, and updating the product node with the computed rating. The average rating gives a summarized view of how users perceive the entity. To calculate the average rating: Retrieve all ratings for a specific entity. Compute the sum of all ratings. Divide the sum by the total number of ratings to get the average. Optionally, round the average to a desired precision (e.g., one decimal place).

##### **3.1.4. Display Rating**

The Display Rating module presents the calculated ratings to users. It ensures that ratings are easily accessible and visually appealing on the product pages. It displays the star ratings provided for the particular product in the online shopping app. Fetch the average rating for each entity from the database. Render the rating using graphical representations (e.g., stars, bar graphs) or numerical values. Provide additional details like total number of ratings and possibly user comments or reviews.

## **CHAPTER 4**

## RESULTS AND DISCUSSION

### 4.1 Results

#### 4.1.1 Creation of node

**Output:**

```
Struct Product* product = createProduct("Sample Product);
```

The module successfully added products to the database with unique identifiers and complete information. Testing showed it could handle a large number of entries without performance degradation.

#### 4.1.2 Add ratings

**Output:**

```
addRating(product,4.5);
```

```
addRating(product,3.7);
```

```
addRating(product,5.0);
```

Users were able to submit ratings smoothly, with data correctly linked to user and product IDs. Feedback indicated high user satisfaction with the rating submission process.

#### 4.1.3 Calculate the ratings

**Output:**

```
float average = calculateaverage(product);
```

The rating calculation module accurately computed average ratings, dynamically

updating as new ratings were added. Performance tests showed minimal latency, ensuring real-time updates.

#### **4.1.4 Display the ratings**

##### **Output:**

##### **displayRating(product);**

Ratings were effectively displayed on product pages, with users appreciating the clarity and detail of the rating information. The visual representation of ratings (e.g., star ratings) was found to be intuitive and engaging.

## **4.2 Discussion**

### **4.2.1.Node Creation:**

The module successfully added products to the database with unique identifiers and complete information. Testing showed it could handle a large number of entries without performance degradation. This step sets up the foundation by defining what can be rated. Each node (entity) typically has unique attributes that define it, such as name, description, category, etc. For example, in an e-commerce platform, products might have attributes like price, description, and category. In a database-driven system, this would typically involve creating a table (e.g., Products) with fields such as product\_id, name, and description

### **4.2.2.Add Rating:**

Users were able to submit ratings smoothly, with data correctly linked to user and product IDs. Feedback indicated high user satisfaction with the rating submission process. This module allows users to submit their ratings for a particular entity (node). Users might rate entities based on predefined criteria (e.g., 1 to 5 stars). Upon user input (e.g., via a form submission), the system would store the rating along with associated metadata (e.g., user ID, timestamp, and rated entity ID) in a ratings table.



### **4.2.3. Calculate Rating:**

The rating calculation module accurately computed average ratings, dynamically updating as new ratings were added. Performance tests showed minimal latency, ensuring real-time updates. This module computes the average rating for each entity based on all submitted ratings. The average rating gives a summarized view of how users perceive the entity. To calculate the average rating: Retrieve all ratings for a specific entity. Compute the sum of all ratings. Divide the sum by the total number of ratings to get the average. Optionally, round the average to a desired precision (e.g., one decimal place).

### **4.2.4. Display Rating:**

Ratings were effectively displayed on product pages, with users appreciating the clarity and detail of the rating information. The visual representation of ratings (e.g., star ratings) was found to be intuitive and engaging. • This module presents the ratings to users in a meaningful way, such as displaying star ratings or numerical averages alongside the entities. Fetch the average rating for each entity from the database. Render the rating using graphical representations (e.g., stars, bar graphs) or numerical values. Provide additional details like total number of ratings and possibly user comments or reviews.

## **CHAPTER 5**

## **CONCLUSION**

The E-Commerce Product Rating System significantly enhances the user experience by providing a reliable and informative way to evaluate products. Each module Node Creation, Add Rating, Calculate Rating, and Display Rating plays a crucial role in maintaining the system's integrity and effectiveness. The system not only helps customers make informed purchasing decisions but also builds trust and transparency in the e-commerce platform. Continued development and refinement of the system will further bolster its utility and user satisfaction, making it a valuable asset for any e-commerce business. This product rating system allows users to rate products and provides functionalities for managing and analyzing these ratings efficiently. By utilizing a singly linked list, this ensures optimized memory usage and streamlined operations for adding, removing, and traversing product ratings.

## **REFERENCES**

1. "Designing and Implementing E-Commerce Applications" by David M. Kroenke and Andrew Gemino (Pearson, 2016):

This book covers the fundamentals of designing and implementing e-commerce applications, including considerations for integrating rating and review systems.

2. "E-commerce 2019: Business, Technology, and Society" by Kenneth C. Laudon and Carol Guercio Traver (Pearson, 2018):

Provides a comprehensive overview of e-commerce trends, technologies, and strategies, which includes discussions on user feedback systems.

## **APPENDIX**

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

// Structure to represent a product
struct Product {
    char productName[100];
    float totalRating;
    int numberOfRatings;
};

// Structure to represent a node in the linked list
struct Node {
    struct Product data;
    struct Node* next;
};

// Function to create a new node
struct Node* createNode(char* productName) {
    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
    strcpy(newNode->data.productName, productName);
    newNode->data.totalRating = 0.0;
    newNode->data.numberOfRatings = 0;
    newNode->next = NULL;
    return newNode;
}

// Function to add a rating to a product
void addRating(struct Node* head, char* productName, float rating) {

```

```

struct Node* temp = head;
while (temp != NULL) {
    if (strcmp(temp->data.productName, productName) == 0) {
        temp->data.totalRating += rating;
        temp->data.numberOfRatings++;
        return;
    }
    temp = temp->next;
}
printf("Product with name '%s' not found.\n", productName);
}

// Function to display ratings of all products
void displayRatings(struct Node* head) {
    struct Node* temp = head;
    printf("Product Name\t\tAverage Rating\n");
    while (temp != NULL) {
        if (temp->data.numberOfRatings > 0) {
            float    averageRating    =    temp->data.totalRating    /    temp-
>data.numberOfRatings;
            printf("%s\t\t%.2f\n", temp->data.productName, averageRating);
        } else {
            printf("%s\t\tNo ratings yet\n", temp->data.productName);
        }
        temp = temp->next;
    }
}

int main() {

```

```

struct Node* head = NULL;
char productName[100];
float rating;
char choice;

do {
    printf("Enter product name: ");
    scanf(" %[^\\n]s", productName);

    if (head == NULL) {
        head = createNode(productName);
    } else {
        struct Node* temp = head;
        while (temp->next != NULL) {
            temp = temp->next;
        }
        temp->next = createNode(productName);
    }

    do {
        printf("Add rating (1.0 - 5.0): ");
        scanf("%f", &rating);
        if (rating < 1.0 || rating > 5.0) {
            printf("Invalid rating. Please enter a rating between 1.0 and 5.0.\\n");
        } else {
            addRating(head, productName, rating);
        }
        printf("Do you want to add another rating for '%s'? (y/n): ", productName);
        scanf(" %c", &choice);
    }
}

```

```

    } while (choice == 'y' || choice == 'Y');

    printf("Do you want to add ratings for another product? (y/n): ");
    scanf(" %c", &choice);

} while (choice == 'y' || choice == 'Y');

// Displaying ratings
displayRatings(head);

// Freeing allocated memory (optional if needed)
struct Node* temp = head;
while (temp != NULL) {
    struct Node* toFree = temp;
    temp = temp->next;
    free(toFree);
}

return 0;
}

```