



## **PSP [20ES104] COURSE PROJECT REPORT**

On

### **“AC INVENTORY MANAGEMENT SYSTEM”**

Developed By:

**H.T.NO:**

**STUDENT NAME:**

**2203A52131**

A. THRISHAL REDDY (TEAM LEAD)

Github = <https://github.com/MrAtr/E-commerce>

**2203A52189**

Y. RITHWIK

**2203A52172**

S. VISHNU TEJA

**2203A52139**

B. SAI VARDHAN REDDY

Under the Guidance of :

Mr. P PRAMOD KUMAR, M.Tech.(Ph.D)

Senior Assistant Professor

Submitted to :

Department Computer Science and Artificial

Intelligence ,SR University.

Ananthasagar(V), Hasanparthy(M), Hanamkonda(Dist.) – 506371

[www.sru.edu.in](http://www.sru.edu.in)

**June 2023**

**Department of Computer Science and Artificial Intelligence.**

**CERTIFICATE**

This is to certify that the PSP course project report entitled “ **AC INVENTORY MANAGEMENT SYSTEM**” is a record of bonafide work carried out by the student(s) A.Thrishal Reddy, Y.Rithwik, S.Vishnu Teja & B.Sai Vardhan Reddy bearing roll number(s) 2203A52131, 2203A52189, 2203A52172 & 2203A52139 of “ **Computer Science and Artificial Intelligence** ” department during the academic year 2022-23.

**Supervisor**

(P Pramod Kumar)

## INDEX

Sl. No	Title	Page No.
1.	Problem statement	1
2.	Module-wise description	2
3.	Knowledge required to develop the project	3
4.	Source code (.c file code followed by .h file code)	4
5.	Results	16

### ***PROBLEM STATEMENT:***

Develop a C Application for reading and storing various Air conditioners (AC's) available in market in terms of (brand,type, capacity, energy rating and price) dynamically and make it available for sorting and searching according to: (brand,type, capacity, energy rating and price) and print the information accordingly.

Provide the functionality for below mentioned:

1. Read details of Air Conditioners dynamically.
2. Display the details of entered AC's According to:

- i) Brand
- ii) Type
- iii) Capacity
- iv) Energy Rating
- v) Price

3. Search According to:

- i) Brand
- ii) Type
- iii) Capacity
- iv) Energy Rating
- v) Price

4. Sort According to:

- i) Brand
- ii) Type
- iii) Capacity
- iv) Energy Rating
- v) Price

## ***MODULES:***

In this application all variables, structures are declared globally so that these variables, and structure members can be accessed throughout the program at any function call. I have created a header file with all the functions prototype and function definition and included it in the main file. We can choose any function by using function calls which are declared in switch-case. In order to repeat the loop control statement (while) is used with condition. I used dot operator to read data from structure and to update it according to the user.

In this application four modules are used.

### **1.Read/Input**

In this module the application asks the person who runs the program to enter Air conditioners details. To give Air conditioners details "for" loop is used.

### **2. Print**

In this module all the stored details of AC's will be displayed on to the screen. In this module printf function and for loop are used.

### **3. Searching**

In this module searching of data is done according to the chosen wise.

In this module there is a sub menu which asks to select

the sorting wise by using switch case. The sorting sub menu will be like press 1. to sort by brand wise press 2. to sort by type wise press 3. to sort capacity wise press 4. to sort by energy-rating wise 5 to sort by price wise.

#### 4. Sorting

In this module sorting of data is done according to the chosen wise.

In this module there is a sub menu which asks to select the sorting wise by using switch case. The sorting sub menu will be like press 1. to sort by brand wise press 2. to sort by type wise press 3. to sort capacity wise press 4. to sort by energy-rating wise 5 to sort by price wise.

### **KNOWLEDGE REQUIRED TO DEVELOP THIS APPLICATION**

- Control Statements (if, if-else, switch)
- Loop Statements (while, for)
- Arrays (1D)
- Strings (Strings and Table of strings) and its functions (strcpy, strcmp)
- Functions (Any type of user defined functions)
- Structure (structure, Dot operator)
- Pointers (pointer to arrays and pointers to structures)
- Macros(#define)

### **SOURCE CODE [.C FILE]:**

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "course_pro1.c"
#define MAX_AC 1000

int main() {
    /*int MAX_AC;
    printf("\n Enter Maximum no. of AC's:");
    scanf("%d",&MAX_AC);*/
    AC acList[MAX_AC];
    int numAC = 0;
    int choice;

    while (1) {
        // Display menu

        printf("\n\nAC Inventory Management System\n");
        printf(" ----- \n");
        printf("1. Add AC\n");
        printf("2. Display ACs\n");
        printf("3. Search AC\n");
        printf("4. Sort AC\n");
        printf("5. Exit\n");
        printf(" ----- \n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                addAC(acList, &numAC);
```

```

        break;
    case 2:
        displayAC(acList, numAC);
        break;
    case 3:
        searchAC(acList, numAC);
        break;
    case 4:
        sortAC(acList, numAC);
        break;
    case 5:
        exit(0);
    default:
        printf("Invalid choice. Try again.\n");
    }
}
return 0;
}

```

### **SOURCE CODE [HEADER FILE]:**

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAX_AC 1000

// Structure for an air conditioner
typedef struct {
    char brand[50];
    char type[50];
    float capacity;
}

```



```

    char energy_rating[10];
    double price;
} AC;

// Function prototypes
void addAC(AC acList[], int *numAC);
void displayAC(AC acList[], int numAC);
void searchAC(AC acList[], int numAC);
void sortAC(AC acList[], int numAC);

// Function to add a new AC to the list
void addAC(AC acList[], int *numAC) {
    if (*numAC == MAX_AC) {
        printf("Error: Maximum number of ACs reached.\n");
        return;
    }

    AC newAC;

    // Get AC details from user
    printf("Enter brand: ");
    scanf("%s", newAC.brand);
    printf("Enter type: ");
    scanf("%s", newAC.type);
    printf("Enter capacity (in tons): ");
    scanf("%f", &newAC.capacity);
    printf("Enter energy rating: ");
    scanf("%s", newAC.energy_rating);
    printf("Enter price: ");

```

```

scanf("%lf", &newAC.price);

// Add AC to the list
acList[*numAC] = newAC;
(*numAC)++;
}

// Function to display all ACs in the list
void displayAC(AC acList[], int numAC) {
    if (numAC == 0) {
        printf("No ACs found.\n");
        return;
    }

    printf("%-20s%-20s%-10s%-15s%-10s\n", "Brand",
    "Type", "Capacity", "Energy Rating", "Price");
    printf("-----\n");
    for (int i = 0; i < numAC; i++) {
        printf("%-20s%-20s%-10f%-15s%-10.2f\n",
        acList[i].brand, acList[i].type, acList[i].capacity,
        acList[i].energy_rating, acList[i].price);
    }
}

// Function to search the AC list by brand, type, capacity,
energy rating, or price
void searchAC(AC acList[], int numAC) {
    if (numAC == 0) {
        printf("No ACs found.\n");
        return;
    }

```

```

}

int choice;
char searchString[100];
int found = 0;
// Display search options
printf("\n\nSearch AC\n");
printf("-----\n");
printf("1. Search by brand\n");
printf("2. Search by type\n");
printf("3. Search by capacity\n");
printf("4. Search by energy rating\n");
printf("5. Search by price\n");
printf("-----\n");
printf("Enter your choice: ");
scanf("%d", &choice);

switch (choice) {
    case 1:
        // Search by brand
        printf("\nEnter the brand name to search for: ");
        scanf("%s", searchString);
        for (int i = 0; i < numAC; i++) {
            if (strcmp(acList[i].brand, searchString) == 0) {
                if (!found) {
                    printf("\nACs matching brand '%s':\n",
searchString);
                    printf("%-20s%-20s%-10s%-15s%-10s\n",
"Brand", "Type", "Capacity", "Energy Rating", "Price");

```

```

        printf("-----\n");
        found = 1;
    }
    printf("%-20s%-20s%-10f%-15s%-10.2f\n",
acList[i].brand, acList[i].type, acList[i].capacity,
acList[i].energy_rating, acList[i].price);
    }
}
break;
case 2:
    // Search by type
    printf("\nEnter the type to search for: ");
    scanf("%s", searchString);
    for (int i = 0; i < numAC; i++) {
        if (strcmp(acList[i].type, searchString) == 0) {
            if (!found) {
                printf("\nACs matching type '%s':\n",
searchString);
                printf("%-20s%-20s%-10s%-15s%-10s\n",
"Brand", "Type", "Capacity", "Energy Rating", "Price");
                printf("-----\n");
                found = 1;
            }
            printf("%-20s%-20s%-10f%-15s%-10.2f\n",
acList[i].brand, acList[i].type, acList[i].capacity,
acList[i].energy_rating, acList[i].price);
        }
    }
}

```

```

        break;
    case 3:
        // Search by capacity
        printf("\nEnter the capacity to search for: ");
        float capacity;
        scanf("%f", &capacity);
        for (int i = 0; i < numAC; i++) {
            if (acList[i].capacity == capacity) {
                if (!found) {
                    printf("\nACs matching capacity '%f':\n",
capacity);
                    printf("%-20s%-20s%-10s%-15s%-10s\n",
"Brand", "Type", "Capacity", "Energy Rating", "Price");
                    printf("-----\n");
                    found = 1;
                }
                printf("%-20s%-20s%-10f%-15s%-10.2f\n",
acList[i].brand, acList[i].type, acList[i].capacity,
acList[i].energy_rating, acList[i].price);
            }
        }
        break;
    case 4:
        // Search by energy rating
        printf("\nEnter the energy rating to search for: ");
        scanf("%s", searchString);
        for (int i = 0; i < numAC; i++) {
            if (strcmp(acList[i].energy_rating, searchString) ==
0) {

```

```

        if (!found) {
            printf("\nACs matching energy rating '%s':\n",
searchString);
            printf("%-20s%-20s%-10s%-15s%-10s\n",
"Brand", "Type", "Capacity", "Energy Rating", "Price");
            printf("-----\n");
            found = 1;
        }
        printf("%-20s%-20s%-10f%-15s%-10.2f\n",
acList[i].brand, acList[i].type, acList[i].capacity,
acList[i].energy_rating, acList[i].price);
    }
}
break;
case 5:
    // Search by price
    printf("\nEnter the price to search for: ");
    float price;
    scanf("%f", &price);
    for (int i = 0; i < numAC; i++) {
        if (acList[i].price == price) {
            if (!found) {
                printf("\nACs matching price '%.2f':\n", price);
                printf("%-20s%-20s%-10s%-15s%-10s\n",
"Brand", "Type", "Capacity", "Energy Rating", "Price");
                printf("-----\n");
                found = 1;
            }
        }
    }
}

```

```

        printf("%-20s%-20s%-10f%-15s%-10.2f\n",
acList[i].brand, acList[i].type, acList[i].capacity,
acList[i].energy_rating, acList[i].price);
    }
}
break;
default:
    printf("\nInvalid choice.\n");
    return;
}

if (!found) {
    printf("\nNo ACs found matching the search
criteria.\n");
}
}

// Function to sort the AC list by brand, type, capacity,
energy rating or price
void sortAC(AC acList[], int numAC) {
    if (numAC == 0) {
        printf("No ACs found.\n");
        return;
    }

    int choice,i,j;
    // Display sort options
    printf("\n\nSort AC\n");
    printf(" ----- \n");
    printf("1. Sort by brand\n");

```

```

printf("2. Sort by type\n");
printf("3. Sort by capacity\n");
printf("4. Sort by energy rating\n");
printf("5. Sort by price\n");
printf(" ----- \n");
printf("Enter your choice: ");
scanf("%d", &choice);
AC temp;
switch (choice) {
    case 1:
        // Sort by brand
        for (int i = 0; i < numAC - 1; i++) {
            for (int j = 0; j < numAC - i - 1; j++) {
                if (strcmp(acList[j].brand, acList[j + 1].brand) >
0) {
                    AC temp = acList[j];
                    acList[j] = acList[j + 1];
                    acList[j + 1] = temp;
                }
            }
        }
        break;
    case 2:
        // Sort by type
        for (int i = 0; i < numAC - 1; i++) {
            for (int j = 0; j < numAC - i - 1; j++) {
                if (strcmp(acList[j].type, acList[j + 1].type) > 0) {
                    AC temp = acList[j];
                    acList[j] = acList[j + 1];

```



```

        acList[j + 1] = temp;
    }
}
}
break;
case 3:
    // Sort by capacity
    for (int i = 0; i < numAC - 1; i++) {
        for (int j = 0; j < numAC - i - 1; j++) {
            if (acList[j].capacity > acList[j + 1].capacity) {
                AC temp = acList[j];
                acList[j] = acList[j + 1];
                acList[j + 1] = temp;
            }
        }
    }
    break;
case 4:
    // Sort by energy rating
    for (int i = 0; i < numAC - 1; i++) {
        for (int j = 0; j < numAC - i - 1; j++) {
            if (strcmp(acList[j].energy_rating, acList[j +
1].energy_rating) > 0) {
                AC temp = acList[j];
                acList[j] = acList[j + 1];
                acList[j + 1] = temp;
            }
        }
    }
}

```

```

        break;
    case 5:
        // Sort by price
        for (int i = 0; i < numAC - 1; i++) {
            for (int j = 0; j < numAC - i - 1; j++) {
                if (acList[j].price > acList[j + 1].price) {
                    AC temp = acList[j];
                    acList[j] = acList[j + 1];
                    acList[j + 1] = temp;
                }
            }
        }
        break;
    default:
        printf("\nInvalid choice.\n");
        return;
}

```

```

// Print the sorted AC list
printf("\nSorted AC List:\n");
printf("%-20s%-20s%-10s%-15s%-10s\n", "Brand",
>Type", "Capacity", "Energy Rating", "Price");
printf(" -----\n");
for (int i = 0; i < numAC; i++) {
    printf("%-20s%-20s%-10f%-15s%-10.2f\n",
acList[i].brand, acList[i].type, acList[i].capacity,
acList[i].energy_rating, acList[i].price);

}
}

```

## RESULTS:

### AC Inventory Management System

-----

1. Add AC
  2. Display ACs
  3. Search AC
  4. Sort AC
  5. Exit
- 

Enter your choice: 1

Enter brand: lg

Enter type: split

Enter capacity (in tons): 1.5

Enter energy rating: 5star

Enter price: 45000

### AC Inventory Management System

-----

1. Add AC
  2. Display ACs
  3. Search AC
  4. Sort AC
  5. Exit
- 

Enter your choice: 1

Enter brand: samsung

Enter type: window

Enter capacity (in tons): 1

Enter energy rating: 3star

Enter price: 30000

## AC Inventory Management System

-----

1. Add AC
  2. Display ACs
  3. Search AC
  4. Sort AC
  5. Exit
- 

Enter your choice: 1

Enter brand: voltas

Enter type: portable

Enter capacity (in tons): 1.3

Enter energy rating: 4star

Enter price: 35000

## AC Inventory Management System

-----

1. Add AC
  2. Display ACs
  3. Search AC
  4. Sort AC
  5. Exit
- 

Enter your choice: 2

Brand	Type	Capacity	Energy Rating	Price
-------	------	----------	---------------	-------

-----

lg	split	1.500000	5star	45000.00
----	-------	----------	-------	----------

samsung	window	1.000000	3star	30000.00
voltas	portable	1.300000	4star	35000.00

## AC Inventory Management System

- 
1. Add AC
  2. Display ACs
  3. Search AC
  4. Sort AC
  5. Exit
- 

Enter your choice: 3

## Search AC

- 
1. Search by brand
  2. Search by type
  3. Search by capacity
  4. Search by energy rating
  5. Search by price
- 

Enter your choice: 1

Enter the brand name to search for: lg

ACs matching brand 'lg':

Brand	Type	Capacity	Energy Rating	Price
lg	split	1.500000	5star	45000.00

## AC Inventory Management System

---

1. Add AC
  2. Display ACs
  3. Search AC
  4. Sort AC
  5. Exit
- 

Enter your choice: 3

### Search AC

---

1. Search by brand
  2. Search by type
  3. Search by capacity
  4. Search by energy rating
  5. Search by price
- 

Enter your choice: 2

Enter the type to search for: window

ACs matching type 'window':

Brand	Type	Capacity	Energy Rating	Price
samsung	window	1.000000	3star	30000.00

## AC Inventory Management System

---

1. Add AC
2. Display ACs
3. Search AC
4. Sort AC
5. Exit

-----  
Enter your choice: 3

Search AC

- 
1. Search by brand
  2. Search by type
  3. Search by capacity
  4. Search by energy rating
  5. Search by price
- 

Enter your choice: 3

Enter the capacity to search for: 1.3

ACs matching capacity '1.300000':

Brand	Type	Capacity	Energy Rating	Price
-----				
voltas	portable	1.300000	4star	35000.00

AC Inventory Management System

- 
1. Add AC
  2. Display ACs
  3. Search AC

4. Sort AC

5. Exit

-----

Enter your choice: 3

Search AC

-----

1. Search by brand

2. Search by type

3. Search by capacity

4. Search by energy rating

5. Search by price

-----

Enter your choice: 4

Enter the energy rating to search for: 5star

ACs matching energy rating '5star':

Brand	Type	Capacity	Energy Rating	Price
-------	------	----------	---------------	-------

-----

lg	split	1.500000	5star	45000.00
----	-------	----------	-------	----------

AC Inventory Management System

-----

1. Add AC

2. Display ACs

3. Search AC

4. Sort AC

5. Exit

-----



Enter your choice: 3

Search AC

- 
1. Search by brand
  2. Search by type
  3. Search by capacity
  4. Search by energy rating
  5. Search by price
- 

Enter your choice: 5

Enter the price to search for: 45000

ACs matching price '45000.00':

Brand	Type	Capacity	Energy Rating	Price
lg	split	1.500000	5star	45000.00

AC Inventory Management System

- 
1. Add AC
  2. Display ACs
  3. Search AC
  4. Sort AC
  5. Exit
- 

Enter your choice: 4

## Sort AC

---

1. Sort by brand
  2. Sort by type
  3. Sort by capacity
  4. Sort by energy rating
  5. Sort by price
- 

Enter your choice: 1

## Sorted AC List:

Brand	Type	Capacity	Energy Rating	Price
lg	split	1.500000	5star	45000.00
samsung	window	1.000000	3star	30000.00
voltas	portable	1.300000	4star	35000.00

## AC Inventory Management System

---

1. Add AC
  2. Display ACs
  3. Search AC
  4. Sort AC
  5. Exit
- 

Enter your choice: 4

## Sort AC

---

1. Sort by brand

2. Sort by type
3. Sort by capacity
4. Sort by energy rating
5. Sort by price

-----  
Enter your choice: 2

Sorted AC List:

Brand	Type	Capacity	Energy Rating	Price
voltas	portable	1.300000	4star	35000.00
lg	split	1.500000	5star	45000.00
samsung	window	1.000000	3star	30000.00

AC Inventory Management System

- 
1. Add AC
  2. Display ACs
  3. Search AC
  4. Sort AC
  5. Exit

-----  
Enter your choice: 4

Sort AC

- 
1. Sort by brand
  2. Sort by type
  3. Sort by capacity
  4. Sort by energy rating

5. Sort by price

-----  
Enter your choice: 3

Sorted AC List:

Brand	Type	Capacity	Energy Rating	Price
samsung	window	1.000000	3star	30000.00
voltas	portable	1.300000	4star	35000.00
lg	split	1.500000	5star	45000.00

AC Inventory Management System

- 
1. Add AC
  2. Display ACs
  3. Search AC
  4. Sort AC
  5. Exit
- 

Enter your choice: 4

Sort AC

- 
1. Sort by brand
  2. Sort by type
  3. Sort by capacity
  4. Sort by energy rating
  5. Sort by price
- 

Enter your choice: 4

Sorted AC List:

Brand	Type	Capacity	Energy Rating	Price
samsung	window	1.000000	3star	30000.00
voltas	portable	1.300000	4star	35000.00
lg	split	1.500000	5star	45000.00

AC Inventory Management System

1. Add AC
2. Display ACs
3. Search AC
4. Sort AC
5. Exit

Enter your choice: 4

Sort AC

1. Sort by brand
2. Sort by type
3. Sort by capacity
4. Sort by energy rating
5. Sort by price

Enter your choice: 5

Sorted AC List:

Brand	Type	Capacity	Energy Rating	Price
-------	------	----------	---------------	-------

```
-----  
samsung      window      1.000000 3star      30000.00  
voltas       portable    1.300000 4star      35000.00  
lg           split       1.500000 5star      45000.00
```

## AC Inventory Management System

```
-----
```

1. Add AC
  2. Display ACs
  3. Search AC
  4. Sort AC
  5. Exit
- ```
-----
```

Enter your choice: 5

```
-----
```

Process exited after 241 seconds with return value 0  
Press any key to continue ...