

LINUX PROJECT REPORT

NAME-AYUSHMAN JOSHI

SAPID-590025854

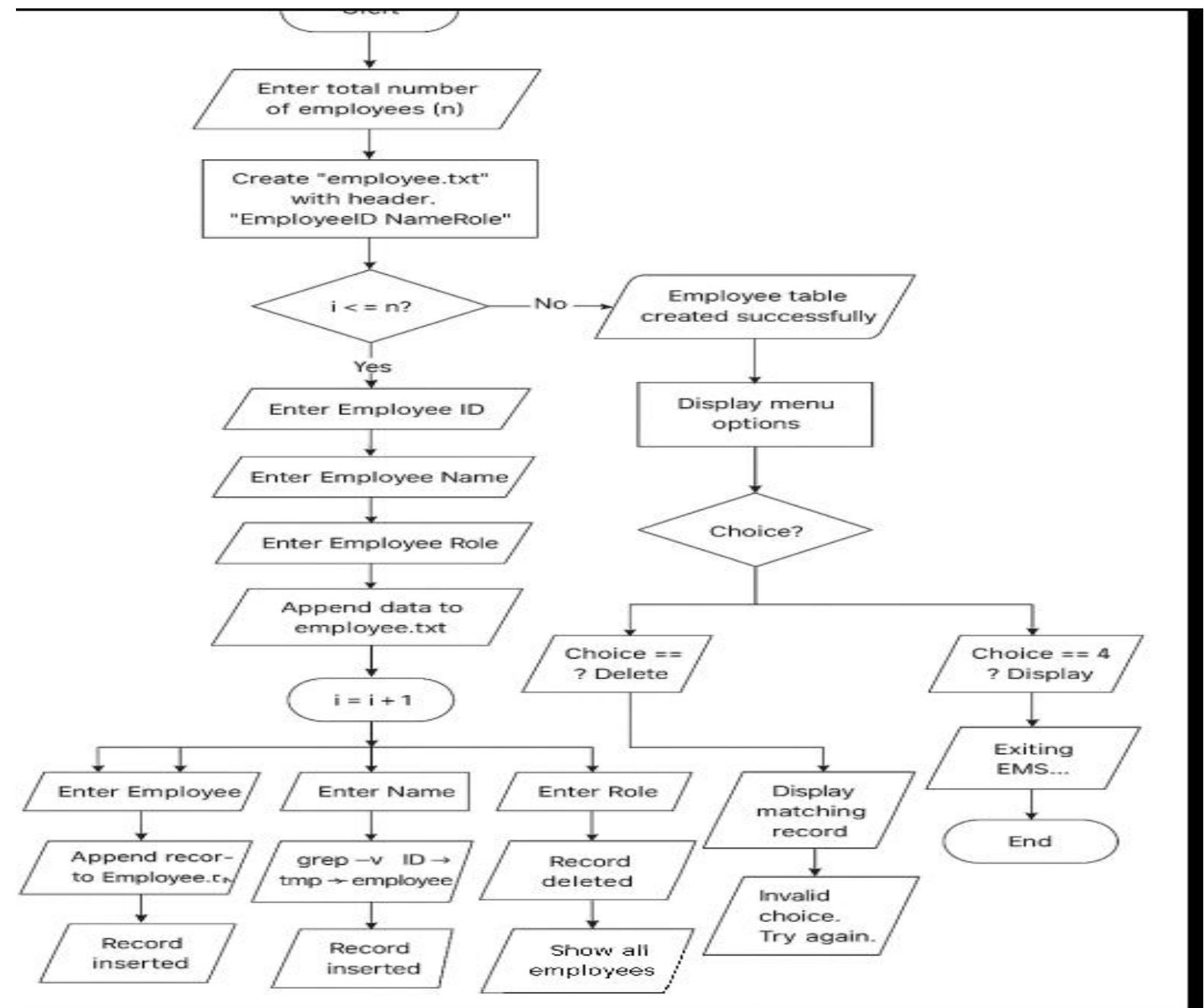
NAME-PRAGATI

SAP ID-590028725

PROBLEM STATEMENT

- Efficient employee data management is vital for any organization. Each employee must be uniquely identified and linked to roles and other details. Manual record-keeping often leads to errors, duplication, and inefficiency, making it difficult to update or retrieve information quickly.
- To overcome these challenges, the project aims to develop an **Employee Management System (EMS) on a Linux environment**. The EMS will provide the following core functionalities:
- **Build the Employee Table:** Create and maintain a structured repository of employee records.
- **Insert New Entries:** Add new employees with unique IDs and associated details.
- **Delete Entries:** Remove outdated or unnecessary employee records.
- **Search a Record:** Retrieve employee information efficiently using unique identifiers or attributes.

FLOWCHART



ALGORITHM

1. Start
2. Set FILE = "employee.txt".
3. Define '**check_duplicate(id)**'

Search file for a line starting with id.

If found- return duplicate exists.

Else- return no duplicate.

4. Define '**check_exists(id)**'

Search file for a line starting with id.

If found- return exists.

Else- return not found.

Main Program Loop

Repeat forever:

5. Display EMS menu:

- i. Create Table
- ii. Insert Entry
- iii. Delete Entry
- iv. Search Entry
- v. Display Table
- vi. Exit

6. Ask user to enter choice – choice.

Case 1: Create Employee Table

7. Ask for number of entries n.

8. Create new file with header:

EmployeeID Name Role

9. Repeat for each employee I =1 to n:

- i. Loop until a **unique Employee ID** is entered:
- ii. Read id

iii. If check_duplicate(id) - show error – ask again.

iv. Else – continue.

Read **name**

Read **role**

Append record to file.

10. Display:" Employee table created successfully!"

Case 2: Insert New Entry

11. Loop until a unique Employee ID is entered:

 Read id

 If duplicate – show error – ask again.

 Else – continue.

12. Read **name**

13. Read **role**

14. Append entry (id name role) to file.

15. Display: "Record inserted successfully."

Case 3: Delete Entry

16. Ask user for id to delete.
17. Check if ID exists using check_exists(id):
 - If exists – remove line containing that ID from file – show "Record deleted successfully."
 - Else – show "Error: Employee ID not found."

Case 4: Search Entry

18. Ask user for id.
19. Check if ID exists:
 - If exists – print matching record.
 - Else – print "Error: Employee ID not found."

Case 5: Display Table

20. Print entire contents of file.

Case 6: Exit

21. Display "Exiting... Thank you!"
22. Terminate program.

Default Case

23. Display "Invalid Choice! Try Again."

Repeat main loop

END.

Problems Our Group Faced?

Problems while checking for same employee id.

Initially the code was not able to differentiate between the same employee id. It was allowing the insertion of different employees with same id.

Problems while executing the code.

While executing the code several errors occurred which were later corrected.

Symbolic errors, errors while putting spaces at the correct places.

It was solved by breaking the code part by part and then reviewing it.

Unclear understanding of the structure of the code initially.

Initially it was not easy to prepare a correct structure of the program. It was then solved by creating the same program in C language.

• Snips Of Code

```
#!/bin/bash

FILE="employee.txt"

check_duplicate() {
    if grep -q "^$1" "$FILE"; then
        return 0      # duplicate exists
    else
        return 1      # no duplicate
    fi
}

check_exists() {
    if grep -q "^$1" "$FILE"; then
        return 0
    else
        return 1
    fi
}

while true
do
    echo ""
    echo "===== Employee Management System ====="
    echo "1. Create Employee Table"
    echo "2. Insert New Entry"
    echo "3. Delete Entry"
    echo "4. Search Entry"
    echo "5. Display Table"
```

```
check_duplicate() {
    if grep -q "^$1 " "$FILE"; then
        return 0      # duplicate exists
    else
        return 1      # no duplicate
    fi
}

check_exists() {
    if grep -q "^$1 " "$FILE"; then
        return 0
    else
        return 1
    fi
}

while true
do
    echo ""
    echo "===== Employee Management System ====="
    echo "1. Create Employee Table"
    echo "2. Insert New Entry"
    echo "3. Delete Entry"
    echo "4. Search Entry"
    echo "5. Display Table"
    echo "6. Exit"
    echo "Enter your choice: "
    read choice

    case $choice in
```

```
case $choice in
  1)
    echo "Enter number of employee entries to create: "
    read n

    echo "Building Employee Table..."
    echo "EmployeeID Name Role" > $FILE

    for (( i=1; i<=n; i++ ))
    do
      echo "Enter details for Employee $i"

      while true
      do
        echo -n "Employee ID: "
        read id
        if check_duplicate "$id"; then
          echo "Error: Employee ID already exists! Enter another."
        else
          break
        fi
      done

      echo -n "Employee Name: "
      read name
      echo -n "Employee Role: "
      read role
      echo "$id $name $role" >> $FILE
      echo "Entry added."
    done
    echo "Employee table created successfully!"
  ::
```

```
;;
2)

while true
do
    echo -n "Enter Employee ID: "
    read id
    if check_duplicate "$id"; then
        echo "Error: Employee ID already exists! Try again."
    else
        break
    fi
done
```

```
echo -n "Enter Employee Name: "
read name
echo -n "Enter Employee Role: "
read role
echo "$id $name $role" >> $FILE
echo "Record inserted successfully."
;;
```

```
3)
echo -n "Enter Employee ID to delete: "
read id

if check_exists "$id"; then
    grep -v "^$id" $FILE > tmp.txt && mv tmp.txt $FILE
    echo "Record deleted successfully."
else
    echo "Error: Employee ID not found."
fi
```

```
fi
;;
4)
    echo -n "Enter Employee ID to search: "
    read id

    if check_exists "$id"; then
        grep "^$id " $FILE
    else
        echo "Error: Employee ID not found."
    fi
;;
5)
    echo "----- Employee Records -----"
    cat $FILE
    echo "-----"
;;
6)
    echo "Exiting... Thank You!"
    exit
;;
*)
    echo "Invalid Choice! Try Again."
;;
esac
done
```

INPUT/OUTPUT SCREENSHOTS

```
AYUSHMAN@AYUSHMAN:~$ chmod +x EMS.sh
AYUSHMAN@AYUSHMAN:~$ ./EMS.sh

===== Employee Management System =====
1. Create Employee Table
2. Insert New Entry
3. Delete Entry
4. Search Entry
5. Display Table
6. Exit
Enter your choice:
1
Enter number of employee entries to create:
3
Building Employee Table...
Enter details for Employee 1
Employee ID: 120034
Employee Name: Rohit
Employee Role: Head of Department
Entry added.
Enter details for Employee 2
Employee ID: 120035
Employee Name: Rohan
Employee Role: Manager
Entry added.
Enter details for Employee 3
Employee ID: 120036
Employee Name: Rahul
Employee Role: Assistant Manager
Entry added.
```

- 2. Insert New Entry
- 3. Delete Entry
- 4. Search Entry
- 5. Display Table
- 6. Exit

Enter your choice:

1

Enter number of employee entries to create:

3

Building Employee Table...

Enter details for Employee 1

Employee ID: 120034

Employee Name: Rohit

Employee Role: Head of Department

Entry added.

Enter details for Employee 2

Employee ID: 120035

Employee Name: Rohan

Employee Role: Manager

Entry added.

Enter details for Employee 3

Employee ID: 120036

Employee Name: Rahul

Employee Role: Assistant Manager

Entry added.

Employee table created successfully!

===== Employee Management System =====

- 1. Create Employee Table
- 2. Insert New Entry
- 3. Delete Entry
- 4. Search Entry
- 5. Display Table

1. Create Employee Table
2. Insert New Entry
3. Delete Entry
4. Search Entry
5. Display Table
6. Exit

Enter your choice:

2

Enter Employee ID: 120033

Enter Employee Name: Raman

Enter Employee Role: Clerk

Record inserted successfully.

===== Employee Management System =====

1. Create Employee Table
2. Insert New Entry
3. Delete Entry
4. Search Entry
5. Display Table
6. Exit

Enter your choice:

3

Enter Employee ID to delete: 120035

Record deleted successfully.

===== Employee Management System =====

1. Create Employee Table
2. Insert New Entry
3. Delete Entry
4. Search Entry
5. Display Table
6. Exit

Enter your choice:

Enter your choice:

4

Enter Employee ID to search: 120036

120036 Rahul Assistant Manager

=====
App Center Employee Management System =====

- 1. Create Employee Table
- 2. Insert New Entry
- 3. Delete Entry
- 4. Search Entry
- 5. Display Table
- 6. Exit

Enter your choice:

5

----- Employee Records -----

EmployeeID Name Role

120034 Rohit Head of Department

120036 Rahul Assistant Manager

120033 Raman Clerk

===== Employee Management System =====

- 1. Create Employee Table
- 2. Insert New Entry
- 3. Delete Entry
- 4. Search Entry
- 5. Display Table
- 6. Exit

Enter your choice:

6

Exiting... Thank You!