|  |  |
| --- | --- |
| ***Project*** |  |
| Project Title | ***SUPERSTORE MANAGERMENT*** |
| Registration Number | ***2020-CS-118*** |
| Summary | *This project is about superstore management in which the user has lots of option and can use a lots of function by using this program. In This program there are 4 kinds of user that can login and everyone has their respective departments that are salesman, manager, storekeeper and the admin. By using this program admin can fully control the management and can easily access each and every report of all the departments. The admin have all the access to add any manager salesman and the storekeeper who look after the warehouse and can add and remove the products and can also change the price of products. The salesman can sell the products that are available in the warehouse. The manager looks after the salesman sale and storekeeper products entry and view the Product with greatest & lowest sale and the Salesman with greatest and lower sale and the admin can give bonus to any employee depending on their sale and progress. I think that’s the complete summary of the project.* |
| Some Features | *>= add employee (manager, salesman, storekeeper)*  *>= remove employee*  *>= view the sales of employee*  *>= add the products*  *>= remove the products*  *>= view all the products*  *>= product with highest & lowest sale*  *>= employee with highest & lowest sale*  *>= give bonus to employee*  *>= sell and return products*  *>= store & view all the information of employee*  *>=validate All the information of employee*  *>=give registration number to the employee*  *>=employee can change their username & password*  *>= change the price of products*  *>=view all the login details of all employees* |
| ***Implementation details*** |  |
| Write down the names of all data types used in this code? | *Int, short, string, stringstream.* |
| How many time For, While & Do While Loop is used? | ***For:***  ***While:***  ***Do While:*** |
| Do you have used switch statement if Yes then enter the purpose of each occurrence. | *Yes, I used switch statements on Main Menu and sub Menus To avoid the code to be more complex through if and else statements.* |
| Do you have any runtime error in your code? | *No.* |
| Do you have any compile time error in your code? | *No.* |
| Enter the names of major modules in your system. | *Functions Declaration, Main Function, Loading Function, Validation Functions, Management Functions, Data Functions, Functions Definitions.* |
| Write down all parallel 1D arrays and their purpose. | ***All 1D Arrays of Manager to store Manager Data:***  *ManNamesList[], ManQualList[], ManIDList[], ManMobNoList[] ,ManUsernameList[], ManPasswordList[], ManRegNoList[], ManAgeList[], ManExpList[], ManSalaryList[];*  ***All 1D Arrays of Salesman to store Salesman Data:***  *SalManNamesList[], SalManQualList[], SalManIDList[], SalManMobNoList[] , SalManUsernameList[], SalManPasswordList[], SalManRegNoList[], SalManAgeList[], SalManExpList[], SalManSalaryList[], SalManBonusList[], SalManSales[];*  ***All 1D Arrays of Storekeeper to store Storekeeper Data:***  *StrkprNamesList[], StrkprQualList[], StrkprIDList[], StrkprMobNoList[] , StrkprUsernameList[], StrkprPasswordList[], StrkprRegNoList[], StrkprAgeList[], StrkprExpList[], StrkprSalaryList[], StrkprBonusList[], StrkprSales[];*  ***All 1D Arrays of Products to store Products Data:***  *PdNamesList[], PdIDList[], PdPriceList[], PdQtyList[], PDSale[], PDSaleQty[];* |
| Write down All 2D arrays & their purpose. | ***All 2D Arrays to store Products, Salesman, Storekeeper & Manager Data From their 1D arrays:*** *PdData[][6], SalManData[][12], StrkprData[][11], ManData[][10]* |
| What is the code length Size in KBs and lines of code? | *121KB*  *3420 lines* |

|  |  |
| --- | --- |
| ***Details of Functions*** |  |
| Function Name 1 | ***Hault()*** |
| Function Prototype | *void* |
| Description | *To clear and stop the screen* |
| Function Return Type & Purpose | *No return type* |
| Parameter Names & Purpose Of Each Parameter | *No Parameters* |

|  |  |
| --- | --- |
| Function Name 2 | ***Convert()*** |
| Function Prototype | *Void* |
| Description | *To convert the names to capital letters* |
| Function Return Type & Purpose | *No return type* |
| Parameter Names & Purpose Of Each Parameter | *&name*  *Address of Variable name* |

|  |  |
| --- | --- |
| Function Name 3 | ***MainMenu()*** |
| Function Prototype | *Int* |
| Description | *To display Main Menu* |
| Function Return Type & Purpose | *Integer and return the option of the Menu Displayed* |
| Parameter Names & Purpose Of Each Parameter | *No Parameters* |

|  |  |
| --- | --- |
| Function Name 4 | ***AdminMenu()*** |
| Function Prototype | *Int* |
| Description | *To display Admin Menu* |
| Function Return Type & Purpose | *Integer and return the option of the Menu Displayed* |
| Parameter Names & Purpose Of Each Parameter | *No Parameters* |

|  |  |
| --- | --- |
| Function Name 5 | ***ManagerMenu()*** |
| Function Prototype | *Int* |
| Description | *To display Manager Menu* |
| Function Return Type & Purpose | *Integer and return the option of the Menu Displayed* |
| Parameter Names & Purpose Of Each Parameter | *No Parameters* |

|  |  |
| --- | --- |
| Function Name 6 | ***SalesmanMenu()*** |
| Function Prototype | *Int* |
| Description | *To display Salesman Menu* |
| Function Return Type & Purpose | *Integer and return the option of the Menu Displayed* |
| Parameter Names & Purpose Of Each Parameter | *No Parameters* |

|  |  |
| --- | --- |
| Function Name 7 | ***StorekeeperMenu()*** |
| Function Prototype | *Int* |
| Description | *To display Storekeeper Menu* |
| Function Return Type & Purpose | *Integer and return the option of the Menu Displayed* |
| Parameter Names & Purpose Of Each Parameter | *No Parameters* |

|  |  |
| --- | --- |
| Function Name 8 | ***initializeManager ()*** |
| Function Prototype | *Void* |
| Description | *To initialize all the arrays of Manager* |
| Function Return Type & Purpose | *No Return type* |
| Parameter Names & Purpose Of Each Parameter | *All these Parameters are Arrays of Manager Details which is to be initialized except last parameter which is Size of Maximum Managers in the System****: string Mandata[][10], string ManNamesList[], int ManAgeList[], int ManExpList[], string ManQualList[], string ManMobNoList[] , string ManIDList[], int ManSalaryList[], string ManUsernameList[], string ManPasswordList[], string ManRegNoList[], int MAX\_Manager*** |

|  |  |
| --- | --- |
| Function Name 9 | ***intializeSalesman()*** |
| Function Prototype | *Void* |
| Description | *To initialize all the arrays of Salesman* |
| Function Return Type & Purpose | *No Return type* |
| Parameter Names & Purpose Of Each Parameter | *All these Parameters are Arrays of Salesman Details which is to be initialized except last parameter which is Size of Maximum Salesman in the System:* ***string SalManData[][12], string SalManNamesList[], int SalManAgeList[], int SalManExpList[], string SalManQualList[], string SalMobNoList[] , string SalManIDList[], int SalManSalaryList[], int SalManBonusList[], string SalManUsernameList[], string SalManPasswordList[], string SalManRegNoList[], int SalManSales[], int MAX\_Salesman*** |

|  |  |
| --- | --- |
| Function Name 10 | ***intializeStorekeeper()*** |
| Function Prototype | *Void* |
| Description | *To initialize all the arrays of Storekeeper* |
| Function Return Type & Purpose | *No Return type* |
| Parameter Names & Purpose Of Each Parameter | *All these Parameters are Arrays of Storekeeper Details which is to be initialized except last parameter which is Size of Maximum Storekeeper in the System****:* *string StrkprData[][11], string StrkprNamesList[], int StrkprAgeList[], int StrkprExpList[], string StrkprQualList[], string StrkprMobNoList[] , string StrkprIDList[], int StrkprSalaryList[], int StrkprBonusList[], string StrkprUsernameList[], string StrkprPasswordList[], string StrkprRegNoList[], int StrkprSales[], int MAX\_Storekeeper*** |

|  |  |
| --- | --- |
| Function Name 11 | ***intializeProducts()*** |
| Function Prototype | *Void* |
| Description | *To initialize all the arrays of Products* |
| Function Return Type & Purpose | *No Return type* |
| Parameter Names & Purpose Of Each Parameter | *All these Parameters are Arrays of Products Details which is to be initialized except last parameter which is Size of Maximum Products in the System:* ***string PdData[][6], string PdNamesList[], string PdIDList[], int PdPriceList[], int PdQtyList[], int PDSale[], int PDSaleQty[], int MAX\_Products*** |

|  |  |
| --- | --- |
| Function Name 12 | ***isValidIDCardNumber()*** |
| Function Prototype | *Bool* |
| Description | *To verify the syntax of ID card Number* |
| Function Return Type & Purpose | *Bool to verify ID Card Syntax* |
| Parameter Names & Purpose Of Each Parameter | ***string ID;***  *To pass ID card Number* |

|  |  |
| --- | --- |
| Function Name 13 | ***isValidName()*** |
| Function Prototype | *Bool* |
| Description | *To verify the syntax of Name* |
| Function Return Type & Purpose | *Bool to verify Name Syntax* |
| Parameter Names & Purpose Of Each Parameter | ***string name;***  *To pass Name* |

|  |  |
| --- | --- |
| Function Name 14 | ***isValidMobNo()*** |
| Function Prototype | *Bool* |
| Description | *To verify the syntax of Mobile Number* |
| Function Return Type & Purpose | *Bool to verify Mobile Number Syntax* |
| Parameter Names & Purpose Of Each Parameter | ***string MobNo;***  *To pass Mobile Number* |

|  |  |
| --- | --- |
| Function Name 15 | ***isValidIUsername()*** |
| Function Prototype | *Bool* |
| Description | *To verify the syntax of Username* |
| Function Return Type & Purpose | *Bool to verify username syntax* |
| Parameter Names & Purpose Of Each Parameter | ***string username;***  *To pass username* |

|  |  |
| --- | --- |
| Function Name 16 | ***isValidPassword()*** |
| Function Prototype | *Bool* |
| Description | *To verify the syntax of Password* |
| Function Return Type & Purpose | *Bool to verify Password Syntax* |
| Parameter Names & Purpose Of Each Parameter | ***string password;***  *To pass password* |

|  |  |
| --- | --- |
| Function Name 17 | ***isValidManAge()*** |
| Function Prototype | *Bool* |
| Description | *To verify Manager Age* |
| Function Return Type & Purpose | *Bool to verify Age of Manager between is in 30 & 45* |
| Parameter Names & Purpose Of Each Parameter | ***int Age;***  *To Pass Age* |

|  |  |
| --- | --- |
| Function Name 18 | ***isValidManExp()*** |
| Function Prototype | *Bool* |
| Description | *To verify Manager Experience* |
| Function Return Type & Purpose | *Bool to verify Manager Experience is in between 3 & 8* |
| Parameter Names & Purpose Of Each Parameter | ***int Exp;***  *To pass Experience* |

|  |  |
| --- | --- |
| Function Name 19 | ***isValidManQual()*** |
| Function Prototype | *Bool* |
| Description | *To verify Manager Qualification* |
| Function Return Type & Purpose | *Bool to verify Manager Qualification is higher than or equal to Graduate(GA) & syntax* |
| Parameter Names & Purpose Of Each Parameter | ***string Qual;***  *To pass Qualification Syntax* |

|  |  |
| --- | --- |
| Function Name 20 | ***isValidManRegNo()*** |
| Function Prototype | *Bool* |
| Description | *To verify Manager Registration Number* |
| Function Return Type & Purpose | *Bool to Verify Manager Registration No. starts with MNx (x is an integer)* |
| Parameter Names & Purpose Of Each Parameter | ***string RegNo***  *To pass registration number syntax* |

|  |  |
| --- | --- |
| Function Name 21 | ***validateManData()*** |
| Function Prototype | *Bool* |
| Description | *To validate All Manager data* |
| Function Return Type & Purpose | *Bool to verify all Manager data* |
| Parameter Names & Purpose Of Each Parameter | ***string Name, int Age, int Exp, string Qual, string MobNo, string ID***  *To validate All Data:* |

|  |  |
| --- | --- |
| Function Name 22 | ***isValidSalManAge()*** |
| Function Prototype | *Bool* |
| Description | *To verify Salesman Age* |
| Function Return Type & Purpose | *Bool to verify Age of Salesman between is in 20 & 45* |
| Parameter Names & Purpose Of Each Parameter | ***int Age;***  *To Pass Age* |

|  |  |
| --- | --- |
| Function Name 23 | ***isValidSalManExp()*** |
| Function Prototype | *Bool* |
| Description | *To verify Salesman Experience* |
| Function Return Type & Purpose | *Bool to verify Salesman Experience is in between 2 & 8* |
| Parameter Names & Purpose Of Each Parameter | ***int Exp;***  *To pass Experience* |

|  |  |
| --- | --- |
| Function Name 24 | ***isValidSalManQual()*** |
| Function Prototype | *Bool* |
| Description | *To verify Salesman Qualification* |
| Function Return Type & Purpose | *Bool to verify Salesman Qualification is higher than or Equal to UnderGraduate(UG) & Syntax* |
| Parameter Names & Purpose Of Each Parameter | ***string Qual;***  *To pass Qualification Syntax* |

|  |  |
| --- | --- |
| Function Name 25 | ***isValidSalManRegNo()*** |
| Function Prototype | *Bool* |
| Description | *To verify Salesman Registration Number* |
| Function Return Type & Purpose | *Bool to Verify Salesman Registration No. starts with SLx (x is an integer)* |
| Parameter Names & Purpose Of Each Parameter | ***string RegNo***  *To pass registration number syntax* |

|  |  |
| --- | --- |
| Function Name 26 | ***validateSalManData()*** |
| Function Prototype | *Bool* |
| Description | *To validate All Salesman data* |
| Function Return Type & Purpose | *Bool to verify all Salesman data* |
| Parameter Names & Purpose Of Each Parameter | ***string Name, int Age, int Exp, string Qual, string MobNo, string ID***  *To validate All Data:* |

|  |  |
| --- | --- |
| Function Name 27 | ***isValidStrkprAge()*** |
| Function Prototype | *Bool* |
| Description | *To verify Storekeeper Age* |
| Function Return Type & Purpose | *Bool to verify Age of Storekeeper between is in 18 & 30* |
| Parameter Names & Purpose Of Each Parameter | ***int Age;***  *To Pass Age* |

|  |  |
| --- | --- |
| Function Name 28 | ***isValidStrkprQual()*** |
| Function Prototype | *Bool* |
| Description | *To verify Storekeeper Qualification* |
| Function Return Type & Purpose | *Bool to verify Storekeeper Qualification syntax* |
| Parameter Names & Purpose Of Each Parameter | ***string Qual;***  *To pass Qualification Syntax* |

|  |  |
| --- | --- |
| Function Name 29 | ***isValidStrkprRegNo()*** |
| Function Prototype | *Bool* |
| Description | *To verify Storekeeper Registration Number* |
| Function Return Type & Purpose | *Bool to Verify Storekeeper Registration No. starts with SKx (x is an integer)* |
| Parameter Names & Purpose Of Each Parameter | ***string RegNo***  *To pass registration number syntax* |

|  |  |
| --- | --- |
| Function Name 30 | ***validateStrkprData()*** |
| Function Prototype | *Bool* |
| Description | *To validate All Storekeeper data* |
| Function Return Type & Purpose | *Bool to verify all Storekkeper data* |
| Parameter Names & Purpose Of Each Parameter | ***string Name, int Age, string Qual, string MobNo, string ID***  *To validate All Data:* |

|  |  |
| --- | --- |
| Function Name 31 | ***isValidProductID()*** |
| Function Prototype | *Bool* |
| Description | *To validate Product ID* |
| Function Return Type & Purpose | *Bool to verify Product ID (i.e XXXYY) where x is capital alphabet & y integer* |
| Parameter Names & Purpose Of Each Parameter | ***string PdID;***  *To pass ID of Product* |

|  |  |
| --- | --- |
| Function Name 32 | ***validatePdData()*** |
| Function Prototype | *bool* |
| Description | *To validate product name & ID* |
| Function Return Type & Purpose | *bool to verify name & ID of Product* |
| Parameter Names & Purpose Of Each Parameter | ***string name, string PdID***  *To pass ID & name of Product* |

|  |  |
| --- | --- |
| Function Name 33 | ***UserExist()*** |
| Function Prototype | *bool* |
| Description | *To check any type of user User exists* |
| Function Return Type & Purpose | *bool to check any type of user User exists* |
| Parameter Names & Purpose Of Each Parameter | ***string UsernameList[], string PasswordList[], int size***  *To pass username & password List and Maximum Size of the required user* |

|  |  |
| --- | --- |
| Function Name 34 | ***UserLogin()*** |
| Function Prototype | *Bool* |
| Description | *To check username & password and acess Login* |
| Function Return Type & Purpose | *Bool to check whether user is registered and given Username & password is correct and acess login* |
| Parameter Names & Purpose Of Each Parameter | ***string UsernameList[], string PasswordList[], string Username, string Password, int size***  *To pass username & password and check it exists in the username & passwords arrays & Maximum Size of user* |

|  |  |
| --- | --- |
| Function Name 35 | ***UserIndex()*** |
| Function Prototype | *int* |
| Description | *To find the index of user* |
| Function Return Type & Purpose | *Integer to check the index of the required user to update Sales of user* |
| Parameter Names & Purpose Of Each Parameter | ***string UsernameList[], string PasswordList[], string Username, string Password, int size***  *To pass username & password & list to find index to update sales* |

|  |  |
| --- | --- |
| Function Name 36 | ***ProductExist()*** |
| Function Prototype | *bool* |
| Description | *To check any type of user Product exists* |
| Function Return Type & Purpose | *bool to check any type of user Product exists* |
| Parameter Names & Purpose Of Each Parameter | ***string PdnamesList[], string PdIDList[], int size***  *To pass name & ID List and Maximum Size of the Products* |

|  |  |
| --- | --- |
| Function Name 37 | ***searchID()*** |
| Function Prototype | *int* |
| Description | *To search ID of User and return index corresponding to ID* |
| Function Return Type & Purpose | *integer to get index of corresponding ID to give Registration number* |
| Parameter Names & Purpose Of Each Parameter | ***string IDList[], int size, string ID***  *To pass ID list & ID entered and match And return the corresponding index and size is the maximum size of the required user* |

|  |  |
| --- | --- |
| Function Name 38 | ***searchRegNo()*** |
| Function Prototype | *int* |
| Description | *To search Registration No. of User and return index corresponding to ID* |
| Function Return Type & Purpose | *Integer, to get index of corresponding Reg. No. to add update & delete user* |
| Parameter Names & Purpose Of Each Parameter | ***string RegNoList[], int size, string RegNo***  *To pass Reg. No. list & Reg. No. entered and match And return the corresponding index and size is the maximum size of the required user* |

|  |  |
| --- | --- |
| Function Name 39 | ***StrRegNo()*** |
| Function Prototype | *Void* |
| Description | *To give Store registration number* |
| Function Return Type & Purpose | *No return type* |
| Parameter Names & Purpose Of Each Parameter | ***string IDList[], int size, string ID, string RegNoList[]***  *To pass ID & ID List and size of the req. user and Reg. No. Arrays to give RegNo. Corresponding to ID index* |

|  |  |
| --- | --- |
| Function Name 40 | ***highest\_score*** |
| Function Prototype | *void* |
| Description | *Compare all sales & shows highest Sales* |
| Function Return Type & Purpose | *No return type* |
| Parameter Names & Purpose Of Each Parameter | ***int UserSale[], int size***  *To pass Sales Array to compare all sales & size is the Maximum size of the required user & shows MAX sales* |

|  |  |
| --- | --- |
| Function Name 41 | ***lowest\_score*** |
| Function Prototype | *void* |
| Description | *Compare all sales & shows highest Sales* |
| Function Return Type & Purpose | *No return type* |
| Parameter Names & Purpose Of Each Parameter | ***int UserSale[], int size***  *To pass Sales Array to compare all sales & size is the Maximum size of the required user & shows MIN sales* |

|  |  |
| --- | --- |
| Function Name 42 | ***ManData()*** |
| Function Prototype | *void* |
| Description | *To store all Manager Data from Manager arrays in one 2D array* |
| Function Return Type & Purpose | *No return type* |
| Parameter Names & Purpose Of Each Parameter | ***string ManData[][10],string ManNamesList[], int ManAgeList[], int ManExpList[], string ManQualList[], string ManMobNoList[], int ManSalaryList[], string ManIDList[], string ManRegNoList[], string ManUsernameList[] , string ManPasswordList[], int size***  *To store all data in One 2D array from all these arrays and size is the Maximum size of Manager.* |

|  |  |
| --- | --- |
| Function Name 43 | ***SalManData()*** |
| Function Prototype | *void* |
| Description | *To store all Salesman Data from Salesman arrays in one 2D array* |
| Function Return Type & Purpose | *No return type* |
| Parameter Names & Purpose Of Each Parameter | ***string SalManData[][12] ,string SalManNamesList[], int SalManAgeList[], int SalManExpList[], string SalManQualList[], string SalManMobNoList[], string SalManIDList[], int SalManSalaryList[], string SalManRegNoList[], int SalManBonusList[], string SalManUsernameList[] , string SalManPasswordList[], int SalManSales[],int size***  *To store all data in One 2D array from all these arrays and size is the Maximum size of Salesman* |

|  |  |
| --- | --- |
| Function Name 44 | ***StrkprData()*** |
| Function Prototype | *void* |
| Description | *To store all Manager Data from Manager arrays in one 2D array* |
| Function Return Type & Purpose | *No return type* |
| Parameter Names & Purpose Of Each Parameter | ***string StrkprData[][11] ,string StrkprNamesList[], int StrkprAgeList[], string StrkprQualList[], string StrkprMobNoList[], string StrkprIDList[], int StrkprSalaryList[], string StrkprRegNoList[], int StrkprBonusList[], string StrkprUsernameList[] , string StrkprPasswordList[], int StrkprSales[],int size***  *To store all data in One 2D array from all these arrays and size is the Maximum size of Storekeeper* |

|  |  |
| --- | --- |
| Function Name 45 | ***PdData()*** |
| Function Prototype | *void* |
| Description | *To store all Manager Data from Manager arrays in one 2D array* |
| Function Return Type & Purpose | *No return type* |
| Parameter Names & Purpose Of Each Parameter | ***string PdData[][6],string PdNamesList[], string PdIDList[], int PdPriceList[] ,int PdQtyList[], int PDSale[], int PDSaleQty[], int size***  *To store all data in One 2D array from all these arrays and size is the Maximum size of Products* |

|  |  |
| --- | --- |
| Function Name 46 | ***AddManager()*** |
| Function Prototype | *Bool* |
| Description | *To add Manager* |
| Function Return Type & Purpose | *Bool, To store all Manager Data in corresponding Arrays and Add manager* |
| Parameter Names & Purpose Of Each Parameter | ***string ManNamesList[], int ManAgeList[], int ManExpList[], string ManQualList[], string ManMobNoList[] , string ManIDList[], int ManSalaryList[], string ManUsernameList[] , string ManPasswordList[] , string Name, int Age, int Exp, string Qual, string MobNo , string ID, int Salary, string Username, string Password, int size***  *To pass all the arrays of Managers and Validate Data Entered and add from arrays* |

|  |  |
| --- | --- |
| Function Name 47 | ***RemoveManager()*** |
| Function Prototype | *bool* |
| Description | *To remove Manager* |
| Function Return Type & Purpose | *Bool, To Remove all Manager Data in corresponding Arrays and Remove manager* |
| Parameter Names & Purpose Of Each Parameter | ***string ManData[][10], string ManNamesList[], int ManAgeList[], int ManExpList[], string ManQualList[], string ManMobNoList[], string ManIDList[], int ManSalaryList[], string ManUsernameList[], string ManPasswordList[], string ManRegNoList[], string RegNo, int size***  *To pass all the arrays of Managers and Validate Data Entered and remove from arrays* |

|  |  |
| --- | --- |
| Function Name 48 | ***AddSalesman()*** |
| Function Prototype | *Bool* |
| Description | *To add Salesman* |
| Function Return Type & Purpose | *Bool, To store all Salesman Data in corresponding Arrays and Add salesman* |
| Parameter Names & Purpose Of Each Parameter | ***string SalManNamesList[], int SalManAgeList[], int SalManExpList[], string SalManQualList[], string SalManMobNoList[] , string SalManIDList[], int SalManSalaryList[], string SalManUsernameList[] , string SalManPasswordList[], int SalManBonusList[], int SalManSales[] ,string Name, int Age, int Exp, string Qual, string MobNo , string ID, int Salary, string Username, string Password, int size***  *To pass all the arrays of Salesman and Validate Data Entered and add from arrays* |

|  |  |
| --- | --- |
| Function Name 49 | ***RemoveSalesman()*** |
| Function Prototype | *bool* |
| Description | *To remove Salesman* |
| Function Return Type & Purpose | *Bool, To Remove all Salesman Data in corresponding Arrays and Remove salesman* |
| Parameter Names & Purpose Of Each Parameter | ***string SalManData[][12], string SalManNamesList[], int SalManAgeList[], int SalManExpList[], string SalManQualList[], string SalManMobNoList[], string SalManIDList[], int SalManSalaryList[], string SalManUsernameList[], string SalManPasswordList[], string SalManRegNoList[], int SalManBonusList[], string RegNo, int SalManSales[], int size***  *To pass all the arrays of Salesman and Validate Data Entered and remove from arrays* |

|  |  |
| --- | --- |
| Function Name 50 | ***AddStorekeeper()*** |
| Function Prototype | *Bool* |
| Description | *To add storekeeper* |
| Function Return Type & Purpose | *Bool, To store all Storekeeper Data in corresponding Arrays and Add storekeeper* |
| Parameter Names & Purpose Of Each Parameter | ***string StrkprNamesList[], int StrkprAgeList[], string StrkprQualList[], string StrkprMobNoList[] , string StrkprIDList[], int StrkprSalaryList[], string StrkprUsernameList[] , string StrkprPasswordList[] , int StrkprBonusList[], int StrkprSales[] ,string Name, int Age, string Qual, string MobNo , string ID, int Salary, string Username, string Password, int size***  *To pass all the arrays of Storekeeper and Validate Data Entered and add from arrays* |

|  |  |
| --- | --- |
| Function Name 51 | ***RemoveStorekeeper()*** |
| Function Prototype | *bool* |
| Description | *To remove Storekeeper* |
| Function Return Type & Purpose | *Bool, To Remove all Storekeeper Data in corresponding Arrays and Remove storekeeper* |
| Parameter Names & Purpose Of Each Parameter | ***string StrkprData[][11], string StrkprNamesList[], int StrkprAgeList[], string StrkprQualList[], string StrkprMobNoList[], string StrkprIDList[], int StrkprSalaryList[], string StrkprUsernameList[], string StrkprPasswordList[], string StrkprRegNoList[], int StrkprBonusList[], string RegNo, int StrkprSales[] ,int size***  *To pass all the arrays of Storekeeper and Validate Data Entered and remove from arrays* |

|  |  |
| --- | --- |
| Function Name 52 | ***StrSalaryUpdate()*** |
| Function Prototype | *bool* |
| Description | *To update Storekeeper Salary* |
| Function Return Type & Purpose | *bool, To update Salary of Storekeeper of corresponding Reg. No.* |
| Parameter Names & Purpose Of Each Parameter | ***string Salary[][11], int SalaryList[],string RegNoList[],string RegNo, int size***  *To pass Reg. No. & Salary List to update the Salary list of Respective Reg. No.* |

|  |  |
| --- | --- |
| Function Name 53 | ***ManSalaryUpdate()*** |
| Function Prototype | *bool* |
| Description | *To update Manager Salary* |
| Function Return Type & Purpose | *bool, To update Salary of Manager of corresponding Reg. No.* |
| Parameter Names & Purpose Of Each Parameter | ***string Salary[][11], int SalaryList[],string RegNoList[],string RegNo, int size***  *To pass Reg. No. & Salary List to update the Salary list of Respective Reg. No.* |

|  |  |
| --- | --- |
| Function Name 54 | ***SalSalaryUpdate()*** |
| Function Prototype | *bool* |
| Description | *To update Salesman Salary* |
| Function Return Type & Purpose | *bool, To update Salary of Salesman of corresponding Reg. No.* |
| Parameter Names & Purpose Of Each Parameter | ***string Salary[][11], int SalaryList[],string RegNoList[],string RegNo, int size***  *To pass Reg. No. & Salary List to update the Salary list of Respective Reg. No.* |

|  |  |
| --- | --- |
| Function Name 55 | ***totalSales()*** |
| Function Prototype | *void* |
| Description | *To show Total Sales of All Salesman or Total Products Add by All Storekeeper* |
| Function Return Type & Purpose | *No return Type* |
| Parameter Names & Purpose Of Each Parameter | ***string SalManData[][12],int SalManSales[],int size***  *To pass this & view the total Sales & Products where size is the maximum Size of storekeeper or Salesman* |

|  |  |
| --- | --- |
| Function Name 56 | ***SalemanSale()*** |
| Function Prototype | *void* |
| Description | *To show Individual Sales of All Salesman* |
| Function Return Type & Purpose | *No return Type* |
| Parameter Names & Purpose Of Each Parameter | ***string SalManData[][12], string SalManRegNoList[],int size***  *To pass this 2D array to view All Registered Salesman with their respective sales* |

|  |  |
| --- | --- |
| Function Name 57 | ***SalGiveBonus()*** |
| Function Prototype | *bool* |
| Description | *To give bonus to Salesman* |
| Function Return Type & Purpose | *bool to give bonus to Salesman to respective Reg. No.* |
| Parameter Names & Purpose Of Each Parameter | ***string bonus[][12], int BonusList[], string RegNoList[],string RegNo, int size***  *To pass 2D array & bonus Array & update the Bonus* |

|  |  |
| --- | --- |
| Function Name 58 | ***StrGiveBonus()*** |
| Function Prototype | *bool* |
| Description | *To give bonus to Storekeeper* |
| Function Return Type & Purpose | *bool to give bonus to Storekeeper to respective Reg. No.* |
| Parameter Names & Purpose Of Each Parameter | ***string bonus[][12], int BonusList[], string RegNoList[],string RegNo, int size***  *To pass 2D array & bonus Array & update the Bonus* |

|  |  |
| --- | --- |
| Function Name 59 | ***MaxSale()*** |
| Function Prototype | *Void* |
| Description | *To view Product with maximum sales and the sales Quantity of product And total sales with this product* |
| Function Return Type & Purpose | *No return Type* |
| Parameter Names & Purpose Of Each Parameter | ***string PdData[][6], string PdIDList[], string PdID, int PdPriceList[], int PdQty, int size, int PDSale[], int PDSaleQty[]***  *To pass 2D array & all these parameters to update on each sale* |

|  |  |
| --- | --- |
| Function Name 60 | ***bill()*** |
| Function Prototype | *int* |
| Description | *To get the bill on sales of product* |
| Function Return Type & Purpose | *Integer , To get the bill of the product on sale* |
| Parameter Names & Purpose Of Each Parameter | ***string PdNamesList[],string PdIDList[], string PdID, int PdPriceList[], int PdSaleQty, int size***  *All these are used in the bill* |

|  |  |
| --- | --- |
| Function Name 61 | ***AddProduct()*** |
| Function Prototype | *bool* |
| Description | *To Add Product* |
| Function Return Type & Purpose | *Bool, To Add Product in the corresponding arrays of the products* |
| Parameter Names & Purpose Of Each Parameter | ***string PdNamesList[], string PdIDList[], int PdPriceList[], int PdQtyList[] ,int PdQty, string PdName, string PdID, int PdPrice, int MAX\_Products***  *To store All Data Entered in respective Arrays* |

|  |  |
| --- | --- |
| Function Name 62 | ***RemoveProduct()*** |
| Function Prototype | *bool* |
| Description | *To Remove Product* |
| Function Return Type & Purpose | *Bool, To Remove Product in the corresponding arrays of the products* |
| Parameter Names & Purpose Of Each Parameter | ***string PdNamesList[], string PdIDList[], int PdPriceList[], int PdQtyList[] ,int PdQty, string PdName, string PdID, int PdPrice, int MAX\_Products***  *To remove All Data Entered in respective Arrays* |

|  |  |
| --- | --- |
| Function Name 63 | ***ProductUpdate()*** |
| Function Prototype | *bool* |
| Description | *To update price of product* |
| Function Return Type & Purpose | *bool , To update price of product* |
| Parameter Names & Purpose Of Each Parameter | ***string PdData[][6], int PdPriceList[],string PdIDList[],string PdID, int MAX\_Products***  *To pass Price list and update the price of respective Product ID* |

|  |  |
| --- | --- |
| Function Name 64 | ***SaleProduct()*** |
| Function Prototype | *Bool* |
| Description | *To sale Product* |
| Function Return Type & Purpose | *Bool, To sale product of respective ID and update its Salesman Sales & Product Quantity* |
| Parameter Names & Purpose Of Each Parameter | ***string PdData[][6], string PdNamesList[], string PdIDList[] , int PdQtyList[], string PdID, int PdQty, int size***  *To update the Product Quantity of respective PdID in PdQtyList.* |

|  |  |
| --- | --- |
| Function Name 65 | ***ReturnProduct()*** |
| Function Prototype | *Bool* |
| Description | *To return Product* |
| Function Return Type & Purpose | *Bool, To return product of respective ID and update its Salesman Sales & Product Quantity* |
| Parameter Names & Purpose Of Each Parameter | ***string PdData[][6], string PdNamesList[], string PdIDList[] , int PdQtyList[], string PdID, int PdQty, int size***  *To update the Product Quantity of respective PdID in PdQtyList.* |

|  |  |
| --- | --- |
| Function Name 66 | ***viewAllProducts*** |
| Function Prototype | *Void* |
| Description | *View all products* |
| Function Return Type & Purpose | *Void, To view all products* |
| Parameter Names & Purpose Of Each Parameter | ***string PdData[][6], string PdIDList[], int size***  *Pass this 2D array & IDList to check Product exists and view All products Size is the maximum size of Products* |

|  |  |
| --- | --- |
| Function Name 67 | ***viewManager*** |
| Function Prototype | *Void* |
| Description | *View all Manager* |
| Function Return Type & Purpose | *Void, To view all Manager* |
| Parameter Names & Purpose Of Each Parameter | ***string ManData[][10], string ManRegNoList[], int size***  *Pass this 2D array & IDList to check Manager exists and view All managers Size is the maximum size of Products* |

|  |  |
| --- | --- |
| Function Name 68 | ***viewSalesman*** |
| Function Prototype | *Void* |
| Description | *View all Salesman* |
| Function Return Type & Purpose | *Void, To view all Salesman* |
| Parameter Names & Purpose Of Each Parameter | ***string SalManData[][12], string SalManRegNoList[], int size***  *Pass this 2D array & IDList to check Salesman exists and view All salesman Size is the maximum size of Products* |

|  |  |
| --- | --- |
| Function Name 69 | ***viewStorekeeper*** |
| Function Prototype | *Void* |
| Description | *View all Storekeeper* |
| Function Return Type & Purpose | *Void, To view all Storekeeper* |
| Parameter Names & Purpose Of Each Parameter | ***string StrkprData[][12], string StrkprRegNoList[], int size***  *Pass this 2D array & IDList to check Storekeeper exists and view All storekeeper Size is the maximum size of Products* |

|  |  |
| --- | --- |
| Function Name 70 | ***ViewLoginDetails()*** |
| Function Prototype | *Void* |
| Description | *View all logins Details* |
| Function Return Type & Purpose | *Void, To view all Login Details of all employees* |
| Parameter Names & Purpose Of Each Parameter | ***string ManData[][10],string SalManData[][12],string Strkpr[][11], int size1, int size2, int size3***  *Pass 2D array and Display all login details where size 1,2,3 are Manager,Salesman & Storekeeper maximum size* |

|  |  |
| --- | --- |
| Function Name 71 | ***changeUserPass()*** |
| Function Prototype | *Bool* |
| Description | *To change Username & password of User* |
| Function Return Type & Purpose | *Bool, to change username & password of Respective Reg. No.* |
| Parameter Names & Purpose Of Each Parameter | ***string UsernameList[], string PasswordList[], string RegNoList[], string Username, string Password ,string RegNo, int size***  *To change username & password in these username & passwords list on the respective Reg. No. and size of required category.* |

|  |  |
| --- | --- |
| Function Name 72 | ***saveManager()*** |
| Function Prototype | *Bool* |
| Description | *To save all details of Manager in the file* |
| Function Return Type & Purpose | *Bool, to tell whether file saved or not* |
| Parameter Names & Purpose Of Each Parameter | ***string ManData[][10], string ManRegNoList[], int size***  *ManData contains All Manager Data and RegNo. List to check the saved data is of registered user and size is the maximum size of users* |

|  |  |
| --- | --- |
| Function Name 73 | ***saveSalesman()*** |
| Function Prototype | *Bool* |
| Description | *To save all details of Salesman in the file* |
| Function Return Type & Purpose | *Bool, to tell whether file saved or not* |
| Parameter Names & Purpose Of Each Parameter | ***string SalManData[][12], string SalManRegNoList[], int size***  *SalManData contains All Salesman Data and RegNo. List to check the saved data is of registered user and size is the maximum size of users* |

|  |  |
| --- | --- |
| Function Name 74 | ***saveStorekeeper()*** |
| Function Prototype | *Bool* |
| Description | *To save all details of Storekeeper in the file* |
| Function Return Type & Purpose | *Bool, to tell whether file saved or not* |
| Parameter Names & Purpose Of Each Parameter | ***string StrkprData[][11], string StrkprRegNoList[], int size***  *StrkprData contains All Storekeeper Data and RegNo. List to check the saved data is of registered user and size is the maximum size of users* |

|  |  |
| --- | --- |
| Function Name 75 | ***saveProducts()*** |
| Function Prototype | *Bool* |
| Description | *To save all details of Products in the file* |
| Function Return Type & Purpose | *Bool, to tell whether file saved or not* |
| Parameter Names & Purpose Of Each Parameter | ***string PdData[][6], string PdIDList[], int size***  *PdData contains All Products Data and ID List to check the saved data is of added products and size is the maximum size of Products* |

|  |  |
| --- | --- |
| Function Name 76 | ***loadUser()*** |
| Function Prototype | *Bool* |
| Description | *To load Admin Username & password* |
| Function Return Type & Purpose | *Bool, to check whether user(admin data) loaded or not* |
| Parameter Names & Purpose Of Each Parameter | ***string& auser, string& apass***  *To load data and stores in the auser & apass Parameters Adress* |

|  |  |
| --- | --- |
| Function Name 77 | ***loadManager()*** |
| Function Prototype | *Bool* |
| Description | *Load Managers data in Managers Arrays* |
| Function Return Type & Purpose | *Bool, To check whether all Managers Data is loaded or not* |
| Parameter Names & Purpose Of Each Parameter | ***string ManData[][10],string ManNamesList[], int ManAgeList[], int ManExpList[], string ManQualList[], string ManMobNoList[], int ManSalaryList[], string ManIDList[], string ManRegNoList[], string ManUsernameList[] , string ManPasswordList[], int size***  *Load Managers Data in respective Arrays* |

|  |  |
| --- | --- |
| Function Name 78 | ***loadSalesman()*** |
| Function Prototype | *Bool* |
| Description | *Load Salesman data in Salesman Arrays* |
| Function Return Type & Purpose | *Bool, To check whether all Salesman Data is loaded or not* |
| Parameter Names & Purpose Of Each Parameter | ***string SalManData[][12] ,string SalManNamesList[], int SalManAgeList[], int SalManExpList[], string SalManQualList[], string SalManMobNoList[], string SalManIDList[], int SalManSalaryList[], string SalManRegNoList[], int SalManBonusList[], string SalManUsernameList[] , string SalManPasswordList[], int SalManSales[], int size***  *Load Salesman Data in respective Arrays* |

|  |  |
| --- | --- |
| Function Name 79 | ***loadStorekeeper()*** |
| Function Prototype | *Bool* |
| Description | *Load Storekeeper data in Storekeeper Arrays* |
| Function Return Type & Purpose | *Bool, To check whether all Storekeeper Data is loaded or not* |
| Parameter Names & Purpose Of Each Parameter | ***string StrkprData[][11] ,string StrkprNamesList[], int StrkprAgeList[], string StrkprQualList[], string StrkprMobNoList[], string StrkprIDList[], int StrkprSalaryList[], string StrkprRegNoList[], int StrkprBonusList[], string StrkprUsernameList[] , string StrkprPasswordList[], int StrkprSales[],int size***  *Load Storekeeper Data in respective Arrays* |

|  |  |
| --- | --- |
| Function Name 80 | ***loadProducts()*** |
| Function Prototype | *Bool* |
| Description | *Load Products data in Products Arrays* |
| Function Return Type & Purpose | *Bool, To check whether all Products Data is loaded or not* |
| Parameter Names & Purpose Of Each Parameter | ***string PdData[][6],string PdNamesList[], string PdIDList[], int PdPriceList[] ,int PdQtyList[], int PDSale[], int PDSaleQty[], int size***  *Load Products Data in respective Arrays* |

|  |  |
| --- | --- |
| Function Name 81 | ***main()*** |
| Function Prototype | *int* |
| Description | *For all the inputs & outputs on the console* |
| Function Return Type & Purpose | *Integers For all inputs & outputs* |
| Parameter Names & Purpose Of Each Parameter | ***No Parameters*** |

|  |  |
| --- | --- |
| ***Format of Files*** |  |
| File Name 1 | *Admin.txt* |
| File type | *Input File* |
| File Format | *Username,password* |
|  |  |
| File Name 2 | *ManagerData.txt* |
| File Type | *Input/Output File* |
| File Format | *MUNAWAR ALI ,35,5,GA,0321-1234567,100000,35201-1095530-8,MN1,kokhar,03218858Ar@* |
|  |  |
| File Name 3 | *SalesmanData.txt* |
| File type |  |
| File Format | *AHMED,25,3,GA,0333-0000000,35201-1111111-9,30000,SL1,0,ahmedpd,032188Ar!,0* |
|  |  |
| File Name 4 | *StorekeeperData.txt* |
| File Type | *Input/Output File* |
| File Format | *IFTIKHAR AHMED,22,UG,0321-8888888,35201-1098859-0,25000,SK1,0,iffi,032188Ar@,0* |
|  |  |
| File Name 5 | *ProductData.txt* |
| File Type | *Input/Output File* |
| File Format | *BICYCLE,BKL11,12000,15,0,0* |

|  |  |
| --- | --- |
| ***Details of Interfaces*** | ***MAIN MENU DETAILS*** |
| *Option 1* | *Login As Administrator* |
| *Purpose* | *To login As Administrator* |
| *Input/Output* | *Input: 1*  *output: Admin Menu* |
| *Validation* | *Input other than 0-5 Output will be Invalid Choice* |
| *Test Cases with Sample inputs* | *1:Admin Menu*  *2:Manager Menu*  *3:Salesman Menu*  *4:Storekeeper Menu*  *5:Save All Data*  *0:Exist*  *All other : Invalid Choice* |
| *Which Array is used for Data For Retrieval?* | *No array* |
| *Which Function is Call on this option?* | *AdminMenu()* |
| *ScreenShots* | *On Wrong username & password*  *On correct Username & password* |
|  |  |
| *Option 2* | *Login As Managers* |
| *Purpose* | *To login As Managers* |
| *Input/Output* | *Input: 2*  *output: Manager Menu* |
| *Validation* | *Input other than 0-5 Output will be Invalid Choice* |
| *Test Cases with Sample inputs* | *1:Admin Menu*  *2:Manager Menu*  *3:Salesman Menu*  *4:Storekeeper Menu*  *5:Save All Data*  *0:Exist*  *All other : Invalid Choice* |
| *Which Array is used for Data For Retrieval?* | *No Array* |
| *Which Function is Call on this option?* | *ManagerMenu()* |
| *ScreenShots* | *On wrong inputs*    *On Correct inputs* |

|  |  |
| --- | --- |
| *Option 3* | *Login As Salesman* |
| *Purpose* | *To login As Salesman* |
| *Input/Output* | *Input: 3*  *output: Salesman Menu* |
| *Validation* | *Input other than 0-5 Output will be Invalid Choice* |
| *Test Cases with Sample inputs* | *1:Admin Menu*  *2:Manager Menu*  *3:Salesman Menu*  *4:Storekeeper Menu*  *5:Save All Data*  *0:Exist*  *All other : Invalid Choice* |
| *Which Array is used for Data For Retrieval?* | *No Array* |
| *Which Function is Call on this option?* | *SalesmanMenu()* |
| *ScreenShots* | *on Wrong input*  *On correct input* |

|  |  |
| --- | --- |
| *Option 4* | *Login As Storekeeper* |
| *Purpose* | *To login As Storekeeper* |
| *Input/Output* | *Input: 4*  *output: Storekeeper Menu* |
| *Validation* | *Input other than 0-5 Output will be Invalid Choice* |
| *Test Cases with Sample inputs* | *1:Admin Menu*  *2:Manager Menu*  *3:Salesman Menu*  *4:Storekeeper Menu*  *5:Save All Data*  *0:Exist*  *All other : Invalid Choice* |
| *Which Array is used for Data For Retrieval?* | *No Array* |
| *Which Function is Call on this option?* | *StorekeeperMenu()* |
| *ScreenShots* | *On wrong input*  *on correct input* |

|  |  |
| --- | --- |
| *Option 5* | *Save All Data* |
| *Purpose* | *To save the data* |
| *Input/Output* | *Input: 5*  *output: All Data Saved Sucessfully* |
| *Validation* | *Input other than 0-5 Output will be Invalid Choice* |
| *Test Cases with Sample inputs* | *1:Admin Menu*  *2:Manager Menu*  *3:Salesman Menu*  *4:Storekeeper Menu*  *5:Save All Data*  *0:Exist*  *All other : Invalid Choice* |
| *Which Array is used for Data For Retrieval?* | *All Salesman,Storekeeper,Managers & product Arrays* |
| *Which Function is Call on this option?* | *SaveManager(),SaveSalesman(),SaveStorekeeper(),SaveProducts().* |
| *ScreenShots* |  |

|  |  |
| --- | --- |
| *Option 0* | *Close the Program* |
| *Purpose* | *To close the program* |
| *Input/Output* | *Input: 1*  *output: Exit* |
| *Validation* | *Input other than 0-5 Output will be Invalid Choice* |
| *Test Cases with Sample inputs* | *1:Admin Menu*  *2:Manager Menu*  *3:Salesman Menu*  *4:Storekeeper Menu*  *5:Save All Data*  *0:Exist*  *All other : Invalid Choice* |
| *Which Array is used for Data For Retrieval?* | *No Array* |
| *Which Function is Call on this option?* | *No Function* |
| *ScreenShots* | *Program Closes* |

|  |  |
| --- | --- |
| ***Details of Interfaces*** | ***SubMenu 1*** |
| *Option 1* | *Add Manager* |
| *Purpose* | *To Add Manager* |
| *Input/Output* | *Input:1*  *Output:* |
| *Validation* | *Age btw 30-45*  *Exp btw 3-8*  *Qual atleast GA*  *ID card with dashes & 13 integers*  *Mob No with dashes*  *If all Cases true* |
| *Test Cases with Sample inputs* | *If any information in validation is wrong Manager will not be added*  *Age btw 30-45*  *Exp btw 3-8*  *Qual atleast GA*  *ID card with dashes & 13 integers*  *Mob No with dashes* |
| *Which Array is used for Data For Retrieval?* | *All Manager Arrays* |
| *Which Function is Call on this option?* | *AddManager()* |
| *Screenshots* |  |
|  |  |
| *Option 2* | *Remove Manager* |
| *Purpose* | *To remove Manager* |
| *Input/Output* | *Input: Manager Store registration Number*  *Output: Enter Store Reg No. of manager To remove* |
| *Validation* | *If Registration Number True*  *Manager Removed & starts with “MN”* |
| *Test Cases with Sample inputs* | *SK1:invalid registration no.*  *MN2:Manager Removed* |
| *Which Array is used for Data For Retrieval?* | *All Managers Array* |
| *Which Function is Call on this option?* | *RemoveManager()* |
| *Screenshots* |  |

|  |  |
| --- | --- |
| *Option 3* | *Update Manager Salary* |
| *Purpose* | *To update Manager salary* |
| *Input/Output* | *Input: Manager Reg No.*  *Output: Salary updated* |
| *Validation* | *If Registration Number True*  *Manager Salary Updated & starts with “MN”* |
| *Test Cases with Sample inputs* | *SK1:invalid registration no.*  *MN2:Manager Removed* |
| *Which Array is used for Data For Retrieval?* | *ManSalaryList[],ManRegNoList[],ManData[][]* |
| *Which Function is Call on this option?* | *ManSalaryUpdate()* |
| *Screenshots* |  |

|  |  |
| --- | --- |
| *Option 4* | *Add Salesman* |
| *Purpose* | *To add Salesman* |
| *Input/Output* | *Input:*  *Output:* |
| *Validation* | *Age btw 20-45*  *Exp btw 2-8*  *Qual atleast UG*  *ID card with dashes & 13 integers*  *Mob No with dashes* |
| *Test Cases with Sample inputs* | *If any information in validation is wrong Salesman will not be added*  *Age btw 20-45*  *Exp btw 2-8*  *Qual atleast UG*  *ID card with dashes & 13 integers*  *Mob No with dashes* |
| *Which Array is used for Data For Retrieval?* | *All Salesman Array* |
| *Which Function is Call on this option?* | *AddSalesman()* |
| *Screenshots* |  |

|  |  |
| --- | --- |
| *Option 5* | *Add storekeeper* |
| *Purpose* | *To Add Storekeeper* |
| *Input/Output* | *Input:*  *Output:* |
| *Validation* | *Age btw 18-30*  *Qual at least MAT()*  *ID card with dashes & 13 integers*  *Mob No with dashes* |
| *Test Cases with Sample inputs* | *If any information in validation is wrong Storekeeper will not be added*  *Age btw 18-30*  *Qual atleast MAT*  *ID card with dashes & 13 integers*  *Mob No with dashes* |
| *Which Array is used for Data For Retrieval?* | *All Storekeeper Array* |
| *Which Function is Call on this option?* | *AddStorekeeper()* |
| *Screenshots* |  |

|  |  |
| --- | --- |
| *Option 6* | *View Sales of Salesman* |
| *Purpose* | *To view Sales of All Salesman* |
| *Input/Output* | *Input:6*  *Output: Sales of Salesman* |
| *Validation* | *--* |
| *Test Cases with Sample inputs* | *---* |
| *Which Array is used for Data For Retrieval?* | *SalManData[][], SalManRegNoList[]* |
| *Which Function is Call on this option?* | *SalemanSales()* |
| *Screenshots* |  |

|  |  |
| --- | --- |
| *Option 7* | *Total Sales* |
| *Purpose* | *Total Sales of All Salesman* |
| *Input/Output* | *Input:7*  *Output: Total Sales of Salesman* |
| *Validation* | *-----* |
| *Test Cases with Sample inputs* | *----* |
| *Which Array is used for Data For Retrieval?* | *SalManData[][], SalManSales[]* |
| *Which Function is Call on this option?* | *totalSales[]* |
| *Screenshots* |  |

|  |  |
| --- | --- |
| *Option 8* | *VIEW ALL DETAILS OF SALESMAN WITH SALARY* |
| *Purpose* | *View Details of Salesman* |
| *Input/Output* | *Input:8*  *Output: Details of Salesman* |
| *Validation* | *---* |
| *Test Cases with Sample inputs* | *----* |
| *Which Array is used for Data For Retrieval?* | *SalManUsernameList[],SalManPasswordList[],SalManData[][],SalManRegNoList[]* |
| *Which Function is Call on this option?* | *UserExist() , viewSalesman()* |
| *Screenshots* |  |

|  |  |
| --- | --- |
| *Option 9* | *VIEW ALL DETAILS OF STOREKEEPER WITH SALARY* |
| *Purpose* | *View Details of Storekeeper* |
| *Input/Output* | *Input:9*  *Output: Details of Storekeeper* |
| *Validation* | *----* |
| *Test Cases with Sample inputs* | *----* |
| *Which Array is used for Data For Retrieval?* | *StrkprUsernameList[],StrkprPasswordList[],StrkprData[][],StrkprRegNoList[]* |
| *Which Function is Call on this option?* | *UserExist() , viewSalesman()* |
| *Screenshots* |  |

|  |  |
| --- | --- |
| *Option 10* | *VIEW ALL DETAILS OF MANAGER WITH SALARY* |
| *Purpose* | *View Details of Manager* |
| *Input/Output* | *Input:10*  *Output: Details of Manager* |
| *Validation* | *-----* |
| *Test Cases with Sample inputs* | *-----* |
| *Which Array is used for Data For Retrieval?* | *ManUsernameList[],ManPasswordList[],ManData[][],ManRegNoList[]* |
| *Which Function is Call on this option?* | *UserExist() , viewManager()* |
| *Screenshots* |  |

|  |  |
| --- | --- |
| *Option 11* | *VIEW LOGIN DETAILS OF ALL EMPLOYEES* |
| *Purpose* | *View login Credentials of All users* |
| *Input/Output* | *Input : 11*  *Output: All Login Details* |
| *Validation* | *---* |
| *Test Cases with Sample inputs* | *----* |
| *Which Array is used for Data For Retrieval?* | *ManData[][], SalManData[][],StrkprData[][]* |
| *Which Function is Call on this option?* | *ViewLoginDetais()* |
| *Screenshots* |  |

|  |  |
| --- | --- |
| *Option 12* | *VIEW ALL PRODUCTS* |
| *Purpose* | *View all products added* |
| *Input/Output* | *Input:12*  *output: all products* |
| *Validation* | *----* |
| *Test Cases with Sample inputs* | *----* |
| *Which Array is used for Data For Retrieval?* | *PdNameList[],PdIDList[],PdData[][]* |
| *Which Function is Call on this option?* | *ProductExist() & viewAllProducts()* |
| *Screenshots* |  |

|  |  |
| --- | --- |
| *Option 13* | *VIEW SALESMAN WITH GREATEST & SMALLEST SALES* |
| *Purpose* | *View Salesman with greatest & smallest Sale* |
| *Input/Output* | *Input:13*  *Output: Salesman with greatest & smallest Sale* |
| *Validation* | *----* |
| *Test Cases with Sample inputs* | *----* |
| *Which Array is used for Data For Retrieval?* | *SalManUsernameList[],SalManPasswordList[],SalManSales* |
| *Which Function is Call on this option?* | *UserExist(), highest\_score(), lowest\_score()* |
| *Screenshots* |  |

|  |  |
| --- | --- |
| *Option 14* | *VIEW PRODUCT WITH GREATEST & SMALLEST SALES* |
| *Purpose* | *View Product with greatest & smallest Sale* |
| *Input/Output* | *Input:13*  *Output: Product with greatest & smallest Sale* |
| *Validation* | *----* |
| *Test Cases with Sample inputs* | *----* |
| *Which Array is used for Data For Retrieval?* | *PdNameList[],PdIDList[],PdSale* |
| *Which Function is Call on this option?* | *ProductExist(), highest\_score(), lowest\_score()* |
| *Screenshots* |  |

|  |  |
| --- | --- |
| *Option 15* | *GO BACK TO MAIN MENU* |
| *Purpose* | *Go back to MainMenu* |
| *Input/Output* | *Input :15*  *Output: Go back to MainMenu* |
| *Validation* | *---* |
| *Test Cases with Sample inputs* | *---* |
| *Which Array is used for Data For Retrieval?* | *---* |
| *Which Function is Call on this option?* | *---* |
| *Screenshots* |  |

|  |  |
| --- | --- |
| *Option 16* | *TO EXIT THE PROGRAM* |
| *Purpose* | *To close the program* |
| *Input/Output* | *Input :16*  *Output: EXIT* |
| *Validation* | *---* |
| *Test Cases with Sample inputs* | *---* |
| *Which Array is used for Data For Retrieval?* | *---* |
| *Which Function is Call on this option?* | *---* |
| *Screenshots* | *---* |

|  |  |
| --- | --- |
| ***Details of Interfaces*** | ***SubMenu 2*** |
| *Option 1* | *GIVE SALESMAN BONUS* |
| *Purpose* | *To give Salesman Bonus* |
| *Input/Output* | *Input: Salesman Reg No.*  *Output: Bonus Added* |
| *Validation* | *If Salesman Registration Number True*  *Salesman Bonus Updated & starts with “SL”* |
| *Test Cases with Sample inputs* | *MN1:invalid registration no.*  *SL1: Bonus Added* |
| *Which Array is used for Data For Retrieval?* | *SalManData[][], SalManBonusList[], SalManRegNoList[]* |
| *Which Function is Call on this option?* | *SalGiveBonus()* |
| *Screenshots* |  |

|  |  |
| --- | --- |
| *Option 2* | *REMOVE SALESMAN* |
| *Purpose* | *To remove Salesman* |
| *Input/Output* | *Input: Salesman Reg No.*  *Output: Salesman Removed* |
| *Validation* | *If Salesman Registration Number True*  *Salesman Removed & starts with “SL”* |
| *Test Cases with Sample inputs* | *MN1:invalid registration no.*  *SL1: Salesman Removed* |
| *Which Array is used for Data For Retrieval?* | *All Salesman Arrays* |
| *Which Function is Call on this option?* | *RemoveSalesman()* |
| *Screenshots* |  |

|  |  |
| --- | --- |
| *Option 3* | *GIVE STOREKEEPER BONUS* |
| *Purpose* | *To give Storekeeper Bonus* |
| *Input/Output* | *Input: Storekeeper Reg No.*  *Output: Bonus Added* |
| *Validation* | *If Storekeeper Registration Number True*  *Storekeeper Bonus Updated & starts with “SK”* |
| *Test Cases with Sample inputs* | *MN1:invalid registration no.*  *SK1: Bonus Added* |
| *Which Array is used for Data For Retrieval?* | *StrkprData[][], StrkprBonusList[], StrkprRegNoList[]* |
| *Which Function is Call on this option?* | *StrGiveBonus()* |
| *Screenshots* |  |

|  |  |
| --- | --- |
| *Option 4* | *REMOVE STOREKEEPER* |
| *Purpose* | *To remove Storekeeper* |
| *Input/Output* | *Input: Storekeeper Reg No.*  *Output: Storekeeper Removed* |
| *Validation* | *If Salesman Registration Number True*  *Salesman Removed & starts with “SL”* |
| *Test Cases with Sample inputs* | *MN1:invalid registration no.*  *SL1: Salesman Removed* |
| *Which Array is used for Data For Retrieval?* | *All Storekeeper Arrays* |
| *Which Function is Call on this option?* | *RemoveStorekeeper* |
| *Screenshots* |  |

|  |  |
| --- | --- |
| *Option 5* | *Update Salesman Salary* |
| *Purpose* | *To update Salesman salary* |
| *Input/Output* | *Input: Salesman Reg No.*  *Output: Salary updated* |
| *Validation* | *If Registration Number True*  *Salesman Salary Updated & starts with “SL”* |
| *Test Cases with Sample inputs* | *SK1:invalid registration no.*  *SL2:Salesman Salary Updated* |
| *Which Array is used for Data For Retrieval?* | *SalManSalaryList[], SalManRegNoList[], SalManData[][]* |
| *Which Function is Call on this option?* | *SalManSalaryUpdate()* |
| *Screenshots* |  |

|  |  |
| --- | --- |
| *Option 6* | *Update Manager Salary* |
| *Purpose* | *To update Manager salary* |
| *Input/Output* | *Input: Manager Reg No.*  *Output: Salary updated* |
| *Validation* | *If Registration Number True*  *Manager Salary Updated & starts with “MN”* |
| *Test Cases with Sample inputs* | *SL1:invalid registration no.*  *SK1:Storekeeper Salary Updated* |
| *Which Array is used for Data For Retrieval?* | *StrkprSalaryList[], StrkprRegNoList[], StrkprData[][]* |
| *Which Function is Call on this option?* | *StrkprSalaryUpdate()* |
| *Screenshots* |  |

|  |  |
| --- | --- |
| *Option 7* | *View Sales of Salesman* |
| *Purpose* | *To view Sales of All Salesman* |
| *Input/Output* | *Input:6*  *Output: Sales of Salesman* |
| *Validation* | *--* |
| *Test Cases with Sample inputs* | *---* |
| *Which Array is used for Data For Retrieval?* | *SalManData[][], SalManRegNoList[]* |
| *Which Function is Call on this option?* | *SalemanSales()* |
| *Screenshots* |  |

|  |  |
| --- | --- |
| *Option 8* | *Total Sales* |
| *Purpose* | *Total Sales of All Salesman* |
| *Input/Output* | *Input:7*  *Output: Total Sales of Salesman* |
| *Validation* | *-----* |
| *Test Cases with Sample inputs* | *----* |
| *Which Array is used for Data For Retrieval?* | *SalManData[][], SalManSales[]* |
| *Which Function is Call on this option?* | *totalSales[]* |
| *Screenshots* |  |

|  |  |
| --- | --- |
| *Option 9* | *VIEW ALL DETAILS OF SALESMAN WITH SALARY* |
| *Purpose* | *View Details of Salesman* |
| *Input/Output* | *Input:8*  *Output: Details of Salesman* |
| *Validation* | *---* |
| *Test Cases with Sample inputs* | *----* |
| *Which Array is used for Data For Retrieval?* | *SalManUsernameList[],SalManPasswordList[],SalManData[][],SalManRegNoList[]* |
| *Which Function is Call on this option?* | *UserExist() , viewSalesman()* |
| *Screenshots* |  |

|  |  |
| --- | --- |
| *Option 10* | *VIEW ALL DETAILS OF STOREKEEPER WITH SALARY* |
| *Purpose* | *View Details of Storekeeper* |
| *Input/Output* | *Input:9*  *Output: Details of Storekeeper* |
| *Validation* | *----* |
| *Test Cases with Sample inputs* | *----* |
| *Which Array is used for Data For Retrieval?* | *StrkprUsernameList[],StrkprPasswordList[],StrkprData[][],StrkprRegNoList[]* |
| *Which Function is Call on this option?* | *UserExist() , viewSalesman()* |
| *Screenshots* |  |

|  |  |
| --- | --- |
| *Option 11* | *VIEW ALL PRODUCTS* |
| *Purpose* | *View all products added* |
| *Input/Output* | *Input:11*  *output: all products* |
| *Validation* | *----* |
| *Test Cases with Sample inputs* | *----* |
| *Which Array is used for Data For Retrieval?* | *PdNameList[],PdIDList[],PdData[][]* |
| *Which Function is Call on this option?* | *ProductExist() & viewAllProducts()* |
| *Screenshots* |  |

|  |  |
| --- | --- |
| *Option 12* | *VIEW SALESMAN WITH GREATEST & SMALLEST SALES* |
| *Purpose* | *View Salesman with greatest & smallest Sale* |
| *Input/Output* | *Input:13*  *Output: Salesman with greatest & smallest Sale* |
| *Validation* | *----* |
| *Test Cases with Sample inputs* | *----* |
| *Which Array is used for Data For Retrieval?* | *SalManUsernameList[],SalManPasswordList[],SalManSales* |
| *Which Function is Call on this option?* | *UserExist(), highest\_score(), lowest\_score()* |
| *Screenshots* |  |

|  |  |  |
| --- | --- | --- |
| *Option 13* | *VIEW PRODUCT WITH GREATEST & SMALLEST SALES* |  |
| *Purpose* | *View Product with greatest & smallest Sale* |  |
| *Input/Output* | *Input:13*  *Output: Product with greatest & smallest Sale* |  |
| *Validation* | *----* |  |
| *Test Cases with Sample inputs* | *----* |  |
| *Which Array is used for Data For Retrieval?* | *PdNameList[],PdIDList[],PdSale* |  |
| *Which Function is Call on this option?* | *ProductExist(), highest\_score(), lowest\_score()* |  |
| *Screenshots* |  |  |

|  |  |
| --- | --- |
| *Option 14* | *CHANGE YOUR USERNAME OR PASSWORD* |
| *Purpose* | *To change Username & Password* |
| *Input/Output* | *Input: Registration Number of Manager & old username & password*  *Output: Username & password updated* |
| *Validation* | *Username without space*  *Password 1 special character,1 capital letter,1 integer & 1 small char* |
| *Test Cases with Sample inputs* | *SK1: invalid RegNo*  *MN1: Enter old username & password* |
| *Which Array is used for Data For Retrieval?* | *ManUsernameList[], ManPasswordList[], ManRegNoList[]* |
| *Which Function is Call on this option?* | *changeUserPass()* |
| *Screenshots* |  |

|  |  |
| --- | --- |
| *Option 15* | *GO BACK TO MAIN MENU* |
| *Purpose* | *Go back to MainMenu* |
| *Input/Output* | *Input :15*  *Output: Go back to MainMenu* |
| *Validation* | *---* |
| *Test Cases with Sample inputs* | *---* |
| *Which Array is used for Data For Retrieval?* | *---* |
| *Which Function is Call on this option?* | *---* |
| *Screenshots* |  |

|  |  |
| --- | --- |
| *Option 16* | *TO EXIT THE PROGRAM* |
| *Purpose* | *To close the program* |
| *Input/Output* | *Input :16*  *Output: EXIT* |
| *Validation* | *---* |
| *Test Cases with Sample inputs* | *---* |
| *Which Array is used for Data For Retrieval?* | *---* |
| *Which Function is Call on this option?* | *---* |
| *Screenshots* | *---* |

|  |  |
| --- | --- |
| ***Details of Interfaces*** | ***SubMenu 3*** |
| *Option 1* | *SALE PRODUCT* |
| *Purpose* | *To sale added Products* |
| *Input/Output* | *Input: Product ID of Product & quantity*  *Output: Remaining Qty & Bill* |
| *Validation* | *Product ID (xxxyy) x:Capital Letter y:integer* |
| *Test Cases with Sample inputs* | *BK111: Invalid Product ID* |
| *Which Array is used for Data For Retrieval?* | *PdData[][],PdNamesList[], PdIDList[], PdQtyList[]* |
| *Which Function is Call on this option?* | *SaleProduct()* |
| *Screenshots* |  |
|  |  |
| *Option 2* | *RETURN PRODUCT* |
| *Purpose* | *To return product* |
| *Input/Output* | *Input: Product ID of Product & quantity*  *Output: Remaining Qty & Bill* |
| *Validation* | *Product ID (xxxyy) x:Capital Letter y:integer* |
| *Test Cases with Sample inputs* | *BK111: Invalid Product ID* |
| *Which Array is used for Data For Retrieval?* | *PdData[][],PdNamesList[], PdIDList[], PdQtyList[]* |
| *Which Function is Call on this option?* | *ReturnProduct()* |
| *Screenshots* |  |

|  |  |
| --- | --- |
| *Option 3* | *VIEW ALL PRODUCTS* |
| *Purpose* | *View all products added* |
| *Input/Output* | *Input:11*  *output: all products* |
| *Validation* | *----* |
| *Test Cases with Sample inputs* | *----* |
| *Which Array is used for Data For Retrieval?* | *PdNameList[],PdIDList[],PdData[][]* |
| *Which Function is Call on this option?* | *ProductExist() & viewAllProducts()* |
| *Screenshots* |  |

|  |  |
| --- | --- |
| *Option 4* | *VIEW YOUR SALES* |
| *Purpose* | *To view Logged in Saleman Sale* |
| *Input/Output* | *---* |
| *Validation* | *---* |
| *Test Cases with Sample inputs* | *---* |
| *Which Array is used for Data For Retrieval?* | *SalManSales[]* |
| *Which Function is Call on this option?* | *---* |
| *Screenshots* |  |

|  |  |
| --- | --- |
| *Option 5* | *VIEW YOUR BONUS* |
| *Purpose* | *To view Logged in Saleman Bonus* |
| *Input/Output* | *---* |
| *Validation* | *---* |
| *Test Cases with Sample inputs* | *---* |
| *Which Array is used for Data For Retrieval?* | *SalManBonusList[]* |
| *Which Function is Call on this option?* | *---* |
| *Screenshots* |  |

|  |  |
| --- | --- |
| *Option 6* | *CHANGE YOUR USERNAME OR PASSWORD* |
| *Purpose* | *To change Username & Password* |
| *Input/Output* | *Input: Registration Number of Salesman & old username & password*  *Output: Username & password updated* |
| *Validation* | *Username without space*  *Password 1 special character,1 capital letter,1 integer & 1 small char* |
| *Test Cases with Sample inputs* | *SK1: invalid RegNo*  *SL1: Enter old username & password* |
| *Which Array is used for Data For Retrieval?* | *SalManUsernameList[], SalManPasswordList[], SalManRegNoList[]* |
| *Which Function is Call on this option?* | *changeUserPass()* |
| *Screenshots* |  |

|  |  |
| --- | --- |
| *Option 7* | *GO BACK TO MAIN MENU* |
| *Purpose* | *Go back to MainMenu* |
| *Input/Output* | *Input :7*  *Output: Go back to MainMenu* |
| *Validation* | *---* |
| *Test Cases with Sample inputs* | *---* |
| *Which Array is used for Data For Retrieval?* | *---* |
| *Which Function is Call on this option?* | *---* |
| *Screenshots* |  |

|  |  |
| --- | --- |
| *Option 8* | *TO EXIT THE PROGRAM* |
| *Purpose* | *To close the program* |
| *Input/Output* | *Input :8*  *Output: EXIT* |
| *Validation* | *---* |
| *Test Cases with Sample inputs* | *---* |
| *Which Array is used for Data For Retrieval?* | *---* |
| *Which Function is Call on this option?* | *---* |
| *Screenshots* | *---* |

|  |  |
| --- | --- |
| ***Details of Interfaces*** | ***SubMenu 4*** |
| *Option 1* | *ADD PRODUCT* |
| *Purpose* | *To add Product* |
| *Input/Output* | *Product Name,ID,Qty,Price* |
| *Validation* | *Product ID first 3 capital letter 2 integer* |
| *Test Cases with Sample inputs* | *BK111: Invalid Product ID* |
| *Which Array is used for Data For Retrieval?* | *PdNamesList[], PdIDList[], PdPriceList[], PdQtyList[]* |
| *Which Function is Call on this option?* | *AddProduct()* |
| *Screenshots* |  |

|  |  |
| --- | --- |
| *Option 2* | *REMOVE PRODUCT* |
| *Purpose* | *To remove Product* |
| *Input/Output* | *Input: GNT11*  *Output: Product Removed Sucessfully* |
| *Validation* | *Product ID first 3 capital letter 2 integer* |
| *Test Cases with Sample inputs* | *BK111: Invalid Product ID* |
| *Which Array is used for Data For Retrieval?* | *PdNamesList[], PdIDList[], PdPriceList[], PdQtyList[]* |
| *Which Function is Call on this option?* | *RemoveProduct()* |
| *Screenshots* |  |

|  |  |
| --- | --- |
| *Option 3* | *VIEW ALL PRODUCTS* |
| *Purpose* | *View all products added* |
| *Input/Output* | *Input:11*  *output: all products* |
| *Validation* | *----* |
| *Test Cases with Sample inputs* | *----* |
| *Which Array is used for Data For Retrieval?* | *PdNameList[],PdIDList[],PdData[][]* |
| *Which Function is Call on this option?* | *ProductExist() & viewAllProducts()* |
| *Screenshots* |  |

|  |  |
| --- | --- |
| *Option 4* | *VIEW THE NO. OF PRODUUCTS YOU ENTERED* |
| *Purpose* | *To view Logged In Storekeeper Products Entry* |
| *Input/Output* | *---* |
| *Validation* | *----* |
| *Test Cases with Sample inputs* | *----* |
| *Which Array is used for Data For Retrieval?* | *StrkprSales[]* |
| *Which Function is Call on this option?* | *----* |
| *Screenshots* |  |

|  |  |
| --- | --- |
| *Option 5* | *VIEW YOUR BONUS* |
| *Purpose* | *To view Logged in Storekeeper Bonus* |
| *Input/Output* | *---* |
| *Validation* | *---* |
| *Test Cases with Sample inputs* | *---* |
| *Which Array is used for Data For Retrieval?* | *StrkprBonusList[]* |
| *Which Function is Call on this option?* | *---* |
| *Screenshots* |  |

|  |  |
| --- | --- |
| *Option 6* | *UPDATE THE PRICE OF PRODUCTS* |
| *Purpose* | *To update the price of product* |
| *Input/Output* | *Input Product ID, New Price*  *output: Price updated Sucessfully* |
| *Validation* | *Product ID first 3 capital letter 2 integer* |
| *Test Cases with Sample inputs* | *BK111: Invalid Product ID* |
| *Which Array is used for Data For Retrieval?* | *PdData[][], PdPriceList[], PdIDList[]* |
| *Which Function is Call on this option?* | *ProductUpdate()* |
| *Screenshots* |  |

|  |  |
| --- | --- |
| *Option 7* | *CHANGE YOUR USERNAME OR PASSWORD* |
| *Purpose* | *To change Username & Password* |
| *Input/Output* | *Input: Registration Number of Storekeeper & old username & password*  *Output: Username & password updated* |
| *Validation* | *Username without space*  *Password 1 special character,1 capital letter,1 integer & 1 small char* |
| *Test Cases with Sample inputs* | *SK1: invalid RegNo*  *SL1: Enter old username & password* |
| *Which Array is used for Data For Retrieval?* | *StrkprUsernameList[], StrkprPasswordList[], StrkprRegNoList[]* |
| *Which Function is Call on this option?* | *changeUserPass()* |
| *Screenshots* |  |

|  |  |
| --- | --- |
| *Option 8* | *GO BACK TO MAIN MENU* |
| *Purpose* | *Go back to MainMenu* |
| *Input/Output* | *Input :7*  *Output: Go back to MainMenu* |
| *Validation* | *---* |
| *Test Cases with Sample inputs* | *---* |
| *Which Array is used for Data For Retrieval?* | *---* |
| *Which Function is Call on this option?* | *---* |
| *Screenshots* |  |

|  |  |
| --- | --- |
| *Option 9* | *TO EXIT THE PROGRAM* |
| *Purpose* | *To close the program* |
| *Input/Output* | *Input :8*  *Output: EXIT* |
| *Validation* | *---* |
| *Test Cases with Sample inputs* | *---* |
| *Which Array is used for Data For Retrieval?* | *---* |
| *Which Function is Call on this option?* | *---* |
| *Screenshots* | *---* |