Lab #4

Task #2 :

#include<iostream>

#include<string>

using namespace std;

class node { // MAKING CLASS NODE FOR DOUBLE LINKED LIST

public:

node\* next;

node\* prev;

int data;

string employe\_name;

int employee\_id;

string product\_name;

int product\_price, product\_id;

node()

{ // CONSTRUCTOR TO INITIALIZE POINTERS WITH NULL

next = 0;

prev = 0;

}

void input() {

cout << "\_\*\*\*\*\*\*\*\*\*INPUT EMPLOYEE\*\*\*\*\*\*\*\* \n";

cout << "enter employee name : ";

cin.ignore();

getline(cin, employe\_name);

cout << "enter employe id : ";

cin >> employee\_id;

}

};

class list

{ // MAKE A CLASS LIST

node\* head;

public:

static int money;

list()

{ // CONSTRUCTOR TO INITIALIZE HEAD WITH NULL

head = 0;

}

void insert\_product() { // FUNCTION TO INSERT PRODUCT

node\* newnode = new node();

cout << "enter product name : " << endl;

cin >> newnode->product\_name;

cout << "enter product price : " << endl;

cin >> newnode->product\_price;

cout << "enter product id : " << endl;

cin >> newnode->product\_id;

if (head == 0) {

head = newnode;

}

else {

newnode->next = head;

head->prev = newnode;

head = newnode;

}

}

void display() { // FUNCTION TO DISPLAY

node\* temp = head;

while (temp != 0) {

cout << "product name : " << temp->product\_name << endl;

cout << "product price : " << temp->product\_price << endl;

cout << "product ID : " << temp->product\_id << endl;

temp = temp->next;

}

cout << "\n";

}

void search() {

cout << "\_\_\_\_\_\_\_PURCHASING\_\_\_\_\_\_\_\_ \n";

int count = 0;

cout << "enter product id you want to purchase : ";

int id;

cin >> id;

bool flag = false;

node\* temp = head;

while (temp != 0) {

count++;

if (id == temp->product\_id) {

flag = true;

break;

}

temp = temp->next;

}

if (flag == true) {

cout << "your product is available \n";

cout << "your quantity of product : ";

int q;

cin >> q;

//temp->pp = temp->pp \* q;

money += temp->product\_price;

DELETE(count);

}

else {

cout << "your product is not available !!!!!!! \n";

}

}

void Delete\_first() { // FUNCTION TO DELETE FIRST

node\* temp = head;

head = head->next;

delete temp;

}

void DELETE(int n) { // FUNCTION TO DELETE ATA NTH POSITION

node\* temp = head;

node\* pr = head;

if (n == 1) {

Delete\_first();

}

else {

for (int i = 1; i < n; i++) {

pr = temp;

temp = temp->next;

}

pr->next = temp->next;

temp->next->prev = pr;

delete temp;

}

}

void billing() {

cout << "total bill : " << money << endl;

}

};

int list::money = 0;

int main() {

node n;

list l;

int choice;

int num;

string username;

string password;

cout << "enter username : ";

cin >> username;

cout << "enter password : ";

cin >> password;

if (username == "admin" && password == "123456")

{

do

{

cout << "1) INSERT EMPLOYEE RECORD \n";

cout << "2) INSERT PRODUCT RECORD \n";

cout << "3) BUY PRODUCT \n";

cout << "4) BILL GENERATE \n";

cout << "5) EXIT \n";

cout << "enter your choice \n";

cin >> choice;

switch (choice)

{

case 1:

n.input();

break;

case 2:

cout << "how many products you want to enter : ";

cin >> num;

for (int i = 0; i < num; i++) {

l.insert\_product();

}

break;

case 3:

l.search();

break;

case 4:

l.billing();

break;

case 5:

cout << "program is ended \n";

break;

default:

cout << "wrong choice!!!!!!!!!!! \n";

break;

}

} while (choice != 5);

}

else {

cout << "wrong password !!!!!!!!!!!!!!!! \n";

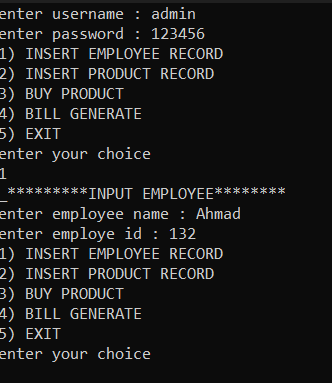
}

system("pause");

return 0;

}

**Output:**

****