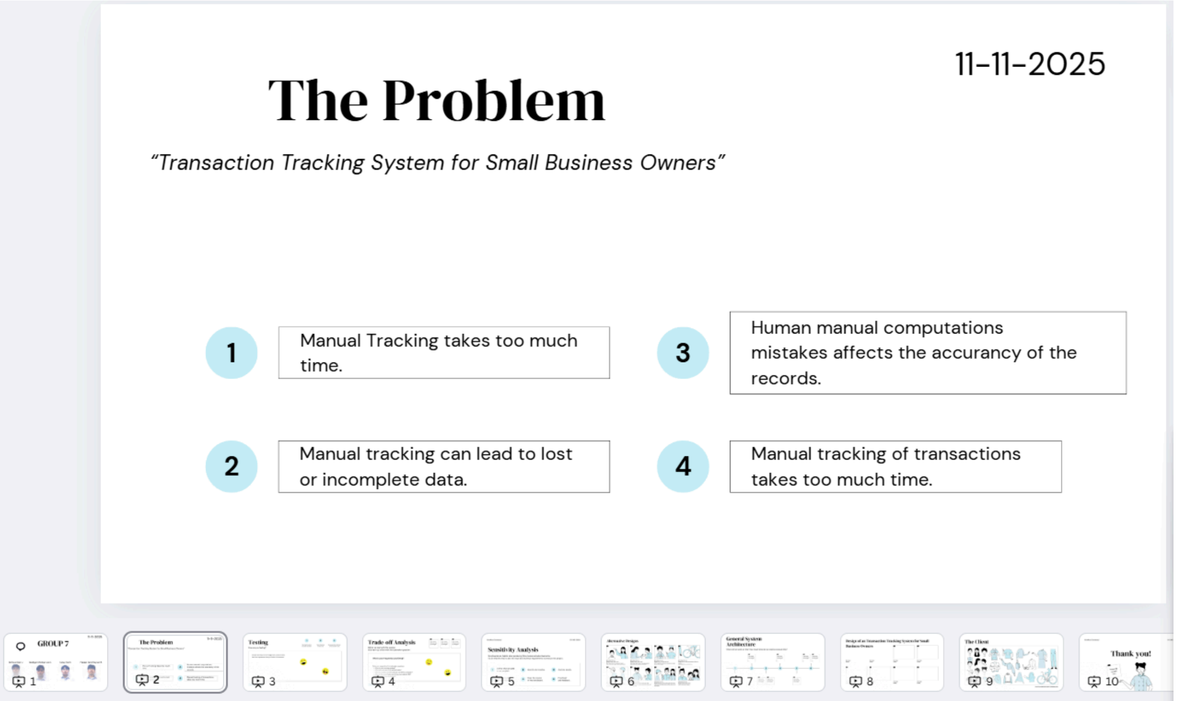
“Transaction Tracking System for Small Business Owners”

Bal, Kevin Ranz J.
Bandigan, Christian Lee C.
Canoy, Hail B.
Pabalan, Hanz Rhayven M.

Work done	Contributions
1. Introduction	ALL
2. Revised Introduction	Bal, Canoy
3. Objectives	Bal, Bandigan, Canoy
4. Brainstorming	ALL
5. Draft of Code	Pabalan,Bandigan
6. Working Inventory tracking code	Bandigan
7. Draft of ppt	ALL
8. Revisions on Introduction	ALL
9. Revision of the code	Pabalan, Bandigan
10. Distribution of works for the weekend	ALL
11. Schedule of google meet	ALL

Draft of PPT :



11-11-2025

The Problem

“Transaction Tracking System for Small Business Owners”

1 Manual Tracking takes too much time.

2 Manual tracking can lead to lost or incomplete data.

3 Human manual computations mistakes affects the accurancy of the records.

4 Manual tracking of transactions takes too much time.

Revisions of Introduction :

Introduction

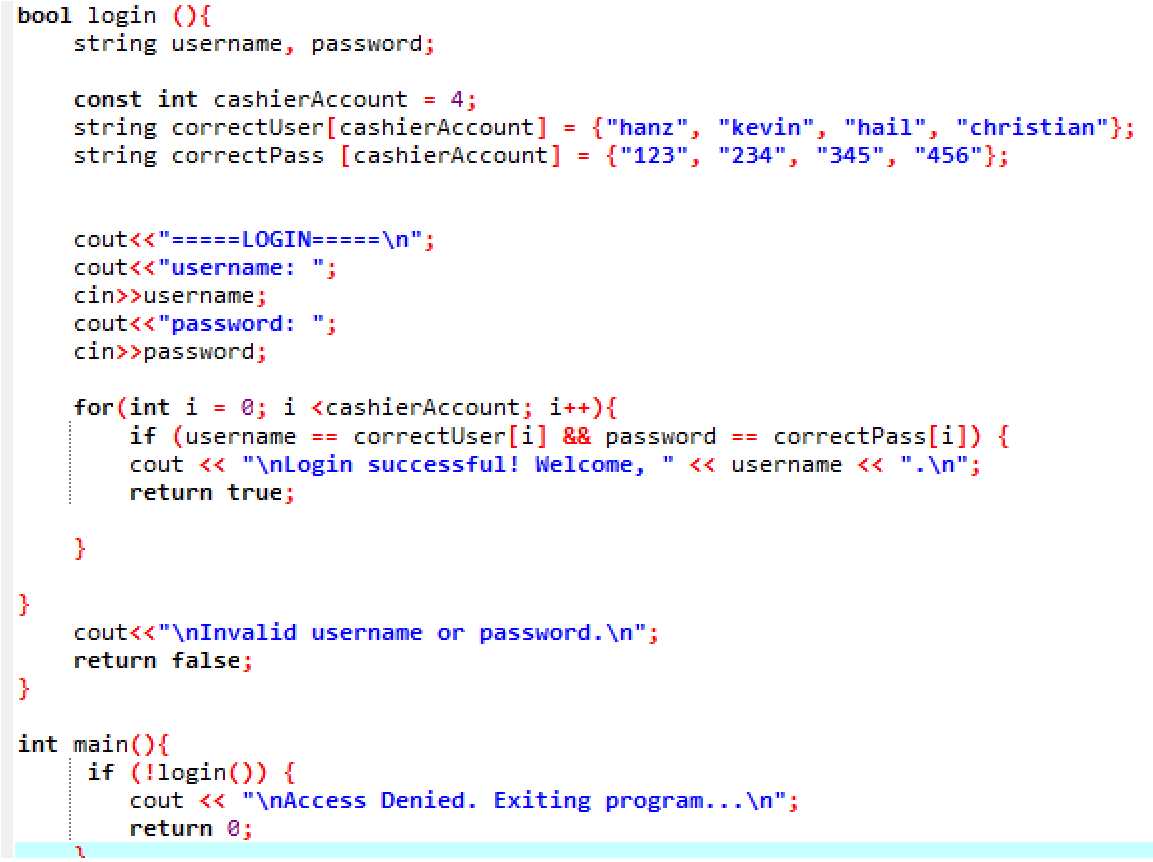
Small business owners mostly rely on notebooks, especially the older generations to record their sales, payments, and inventory. These are the traditional ways but are time consuming and inefficient. As the business grows it will have more sales and payments to track making it more difficult to do *(Giddh Blog, 2025)*.

The plan of action is to deal with the difficulty of tracking the sales, payments, and inventory using the traditional method because they lack access or knowledge to use digital tools. Many small business owners do not use accounting systems because of their high cost and its complex system. Moreover, manual sales computation and inventory tracking is prone to human error and miscalculations leading to product shortages/overstocking and can affect the store's income *(Wynn & Kuhn, 2021; Gestisoft, 2023; Retalon, 2025)*.

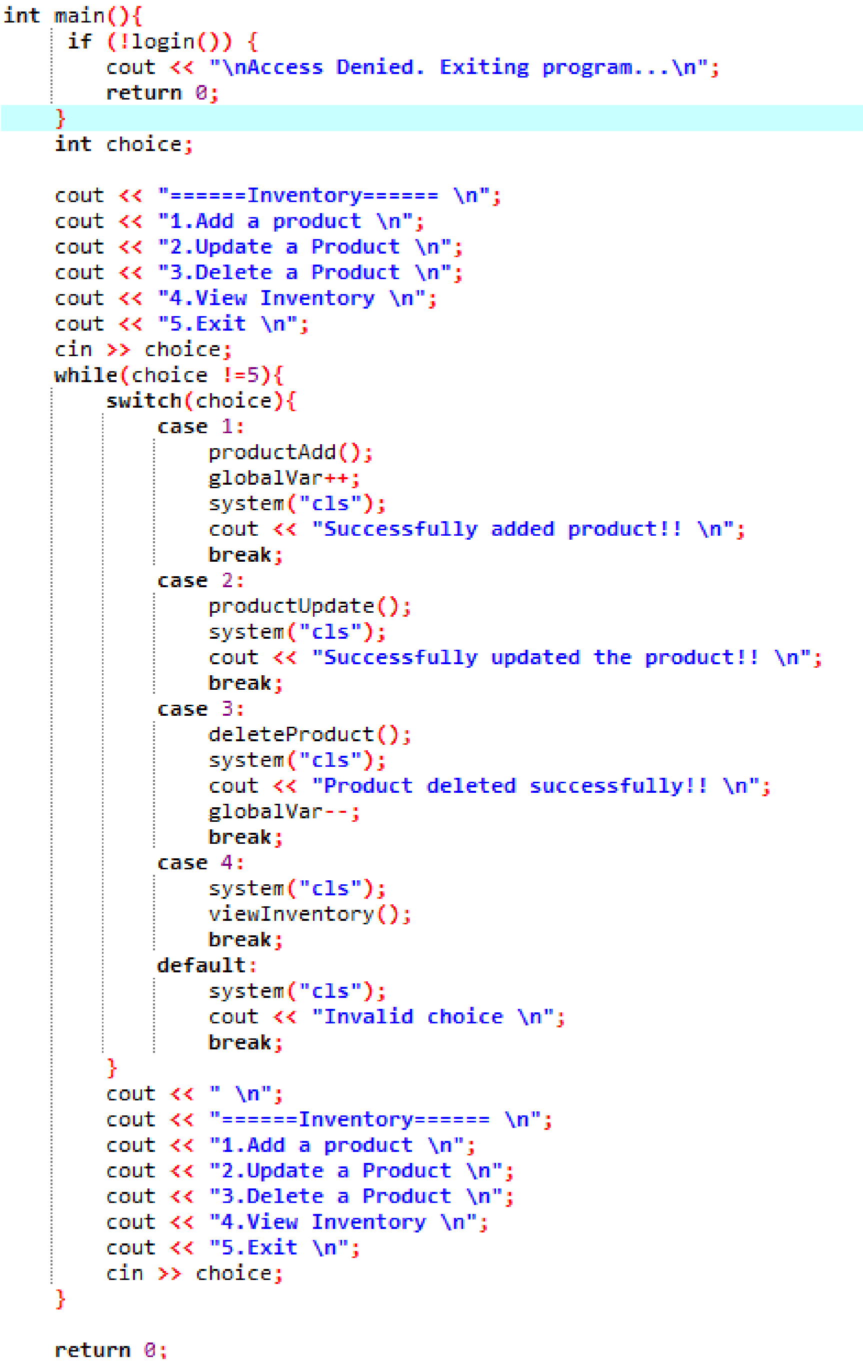
To address this problem, The team would like to propose a Transaction Tracking System and Management System that will simplify sales, payment, and inventory tracking for business owners. Instead of relying on hand written notebooks our system will allow users to input transaction details digitally. And it will automatically generate an organized receipt that can be viewed or printed using notepad.

(Lee et al., 2021) studied the accuracy of inventory management in an organization by replacing manual counting with an automated method. The researchers found that manual counting often ended in problems such as misleading totals and missing items caused by human error. Automation enables the organization to collect information more well and update inventory quickly. This study supports the group's idea by showing that using digital systems instead of handwritten records can reduce errors and improve efficiency for tracking sales, payments, and inventories for small business owners.

Draft of the code :



```
bool login (){
    string username, password;

    const int cashierAccount = 4;
    string correctUser[cashierAccount] = {"hanz", "kevin", "hail", "christian"};
    string correctPass [cashierAccount] = {"123", "234", "345", "456"};


    cout<<"=====LOGIN=====\n";
    cout<<"username: ";
    cin>>username;
    cout<<"password: ";
    cin>>password;

    for(int i = 0; i <cashierAccount; i++){
        if (username == correctUser[i] && password == correctPass[i]) {
        cout << "\nLogin successful! Welcome, " << username << ".\n";
        return true;

    }

}
    cout<<"\nInvalid username or password.\n";
    return false;
}

int main(){
     if (!login()) {
        cout << "\nAccess Denied. Exiting program...\n";
        return 0;
    }
```



```
int main(){
     if (!login()) {
        cout << "\nAccess Denied. Exiting program...\n";
        return 0;
    }
    int choice;

    cout << "======Inventory====== \n";
    cout << "1.Add a product \n";
    cout << "2.Update a Product \n";
    cout << "3.Delete a Product \n";
    cout << "4.View Inventory \n";
    cout << "5.Exit \n";
    cin >> choice;
    while(choice !=5){
        switch(choice){
            case 1:
                productAdd();
                globalVar++;
                system("cls");
                cout << "Successfully added product!! \n";
                break;
            case 2:
                productUpdate();
                system("cls");
                cout << "Successfully updated the product!! \n";
                break;
            case 3:
                deleteProduct();
                system("cls");
                cout << "Product deleted successfully!! \n";
                globalVar--;
                break;
            case 4:
                system("cls");
                viewInventory();
                break;
            default:
                system("cls");
                cout << "Invalid choice \n";
                break;
        }
        cout << " \n";
        cout << "======Inventory====== \n";
        cout << "1.Add a product \n";
        cout << "2.Update a Product \n";
        cout << "3.Delete a Product \n";
        cout << "4.View Inventory \n";
        cout << "5.Exit \n";
        cin >> choice;
    }

    return 0;
```