Vehicle Management System

Team Members:

Mustafa Zahid
(20K-1045)

Dehya Khurraim
(20K-0128)

Hussain
(20K-0185)

Accounting and Finance Project Report

Description

Our software is developed for a vehicle company named Apollo, to manage the financial record. It would consist of all the sub-elements of the basic five pillars of accounting namely asset, liability, revenue, expense and capital.

It will be used by the Accountants hired by the Apollo.

- The user will be provided various options along the way and based on the options selected; we would categorize the sub-elements and build a general journal for a given month of the year.
- Cash inflow and outflow, parts/supplies bought by cash or on account/note are some examples.
- The company will also check how many shares are bought and based on that will pay cash/dividend payable to the stock owners of the company.
- Trial balance will also be made in an external file for user to observe the balances of all ledgers are compiled into debit and credit account.

Code Snippets

```
#include<iostream>
#include<string>
#include<fstream>
#include<iomanip>
#include<conio.h>
#include<Windows.h>
#include<ctime>
using namespace std;
```

These are the build-in libraries we have used in our program.

```
class journal
{
private:
    string from;
    string to;
    string date;
    long long int amount;
    ofstream ir;
    int count;
```

These are the data member of journal Class.

```
long long int trial_debit_sum=0, trial_credit_sum=0;
class ledger_trial
{

private:
    //static int count;
    ofstream out_file,out_file_trial;
    ifstream in_file;
    string filename, account_name;
    long long int debit_sum, credit_sum ;
    string entery, date, entry;
    long long int amount, amount1, balance_cc;
    bool flag=true;
```

These are the data member of ledger_trial Class.

```
void get_entries()
{
    ir.open("journal.txt");
    display_logo();            //funtion call for design first name etc for journal
    int i = 0;
    while (i == 0)
    {
        ++count;
        in << endl;
        system("cls");
        debit();


        credit();


        string disc;           //variable for taking discription of journal entry

        cout << "\n\n\tENTER DESCRIPTION:   ";
        cin.ignore();
        getline(cin, disc);
    //  cin.ignore();
        in <<setw(11)<<"|(" <<setw(50) << disc<<")" << endl;

            for (int j = 125; j > 0; j--)
            {
                in << "-";
            }

            cout << "\n\n\n\n              PRESS 1 TO EXIT 0 TO CONTINUE:  ";
            cin >> i;
            system("cls");
    }
    ir.close();

}
```

The function get_entries ask the data from the user and create an external file of journal

```
void credit()
{
    in << setw(10) << "|";
    cout << "\n\n\tENTER CREDIT ENTRY:   ";
    cin >> tc;

    in << setw(40) <<"to "<< tc;
    cout << "\n\n\tENTER CREDIT AMOUNT:    ";
    cin >> amount;

    in << setw(50) << amount << endl;
}
void debit()
{
    cout << "\n\n\tENTER DATE OF ENTRY:    ";
    cin.clear();
    //cin.ignore();
    cin.clear();
    //cin.fill();

    getline(cin, date);
    if (count > 1)
    {
        getline(cin, date);
    }

    in << date;
    cout << "\n\n\tENTER DEBIT ENTRY:    ";
    cin.clear();
//  cin.ignore();
    cin.clear();

    getline(cin, from);
    in << setw(20) << from;
    cout << "\n\n\tENTER DEBIT AMOUNT:   ";
    cin >> amount;
    in << setw(70) << amount << endl;
}
```

The function credit is called by the get_enteries and in this system asks user data regarding credit entries.

The function debit is called by the get_enteries and in this system asks user data regarding debit entries.

```
void display_logic()
{
    string name;
    cout << "\n\n\tENTER NAME FOR JOURNAL:   ";

    cin.ignore();
    cin.clear();
    getline(cin, name);
    cout << "\n\n\tENTER DATE OF JOURNAL:   ";

    //cin.ignore();
    getline(cin, date);

    in << setw(50) << name << endl;
    in << setw(50) << "JOURNAL" << endl;
    in << setw(50) << date << endl;
        for (int j = 125; j > 0; j--)
        {
            in << "-";
        }

        in << endl;
        in << setw(10) << "DATE|" << setw(25) << "DESCRIPTION" << setw(50) << "|DEBIT" << setw(20) << "|CREDIT" << endl;

        for (int j = 125; j > 0; j--)
        {
            in << "-";
        }

}
```

This function display_logic ask user the name of journal and date the journal is created.

It also makes so styling the external journal file.

```
void start()
{
    string ledger_name; //for concantination
    debit_sum = credit_sum = 0;
    balance_cd = 0;                 //for delete the data of previous ledger
    filename = account_name =entry_date=entry= "";
    cout << "\n\n\n\n                ENTER LEDGER NAME    ";
    cin >> filename;

    ledger_name = filename;
    ledger_name = ledger_name + ".txt";          //concentinate the string with txt

    ledgernames[index++]=ledger_name;

    out_file.open(ledger_name.c_str());          //create file for ledger
    in_file.open("journal.txt");                 //open file for searching and catching data

    //taking data from journal and design the ledger
    getline(in_file, account_name);
    out_file << account_name << endl;
    out_file << setw(50) <<filename<<endl;
    out_file << setw(50) << "LEDGER" << endl;

    //for line
        for (int j = 80; j > 0; j--)
        {
            out_file << "-";
        }

    out_file << endl;
    out_file  << "Date" << setw(20) << "Particular" ;
    out_file << setw(15) << "Amount" << setw(10) << "Date" << setw(20) << "Particular" << setw(15) << "Amount" << endl;

    //for line
        for (int j = 80; j > 0; j--)
        {
            out_file << "-";
        }

    out_file << endl;
    get_entries();                  //new function call

}
```

This function start asks user the name of ledger

It also makes so styling the external ledger file.

```
void get_entries()
{
    string waste;                //for waste lines
    for (int i = 0; i < 5;i++)
        getline(in_file, waste);        //remove waste lines that is catching from journal file
    //cout << waste << endl;
    do
    {
        in_file >> date >> entery >> amount;            //taking debit entries from journal file
    //  cout << date << " " << entery << amount << endl;
        for (int i = 0; i < 2; i++)
            in_file >> waste;

        in_file >> entry >> amount1;                    //taking credit entries from journal file
        //cout << entry << amount1<<endl;
        //checking filename with jounral entries

        if (entery == filename)
        {
            out_file << setw(10) << date << setw(20) << entery << setw(10) << amount1<<"|" << endl;     //
            debit_sum += amount;            //it will sum of ledger entries from debit side
        }
        else if (entry == filename)
        {
            out_file << setw(41)<<"|"<<setw(5) << date << setw(20) << entry << setw(15) << amount << endl;
            credit_sum += amount1;          //it will sum of ledger entries from credit side
        }


        for (int i = 0; i < 3; i++)
        {
            getline(in_file, waste);
        }
    } while (!in_file.eof());
    total();            //function call
    trial();

}
```

This get_enteries is extracting the entries of ledger from the external journal file and adding them in external ledger file.

```
void total()
{
    //this function is doing last step that is total and finish ledger with balance carry down
    if (debit_sum > credit_sum)
    {
        flag = true;
        balance_cd = debit_sum - credit_sum;

        for (int j = 80; j > 0; j--)
        {
            out_file << "-";
        }
        out_file << endl;
        out_file << setw(10) << "Total" << setw(10) << debit_sum - credit_sum << endl;

        for (int j = 80; j > 0; j--)
        {
            out_file << "-";
        }

        out_file << endl;
    }
    else
    {
        flag = false;
        balance_cd = credit_sum - debit_sum;

            for (int j = 80; j > 0; j--)
            {
                out_file << "-";
            }

        out_file << endl;
        out_file << setw(55) << "Total " << setw(15) <<credit_sum - debit_sum << endl;

            for (int j = 80; j > 0; j--)
            {
                out_file << "-";
            }

        out_file << endl;
        out_file.close();       //ledger file is done and now it is closed
        in_file.close();        //journal file is also close we take all wanted material from journal files
    }
}
```

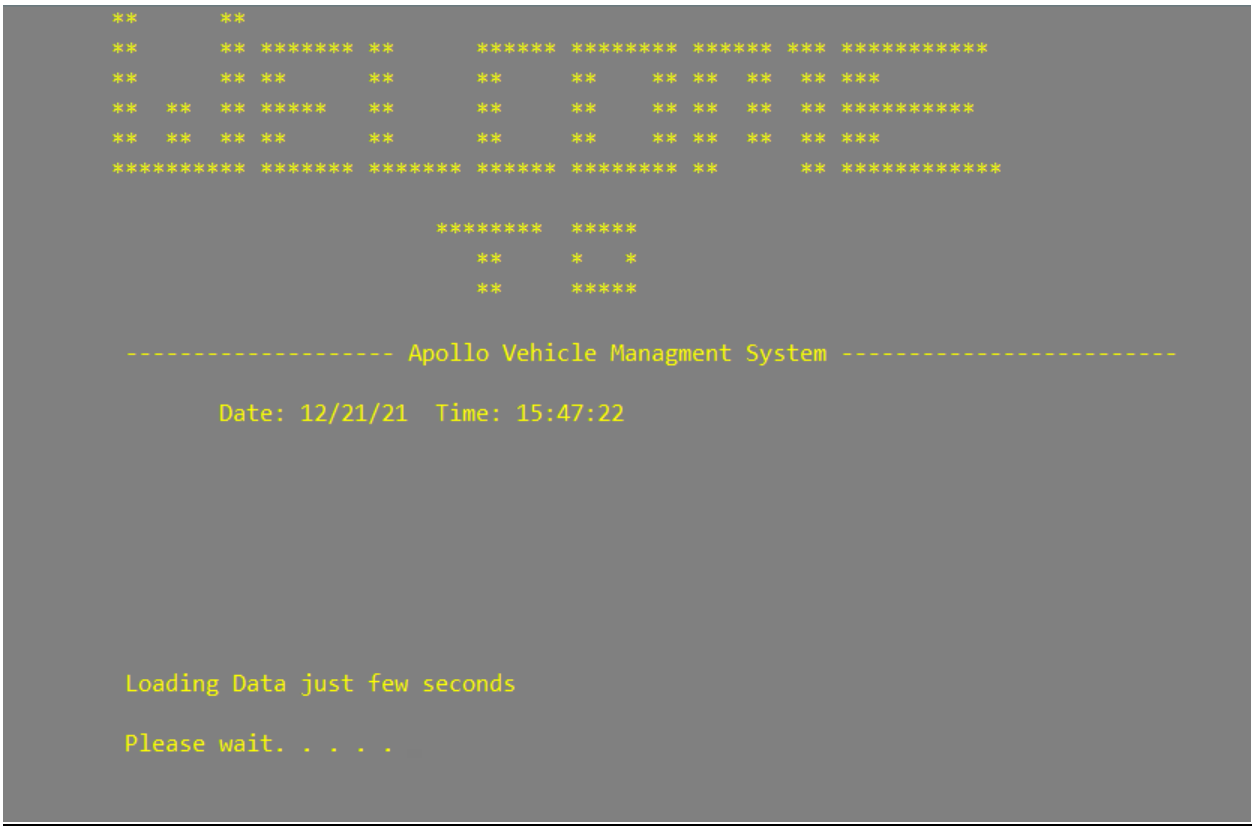
The function total is calculation the total amount of each ledger by subtracting debit & credit sum.

```
void desigr()
{
    //this is function that is for journal
    //design that will help to desiggn trial
    //for balance and taking info from journal


    string name_of_account,date_of_trial;
    ifstream in_file_desigr;

    in_file_desigr.oper("journal.txt", ios::ir);        //open file for dates and name
    getlinc(in_file_desigr, name_of_account);

    out_file_trial << name_of_account << endl;
    out_file_trial << setw(56) << "TRIAL" << endl;

    getlinc(in_file_desigr, date_of_trial);
    getlinc(in_file_desigr, date_of_trial);


    out_file_trial << setw(56) << date_of_trial << endl;
    in_file_desigr.closc();                         //close file

        for (int j = 56; j > 0; j--)
        {
            out_file_trial << "-";
        }

    out_file_trial << endl;


    out_file_trial<< setw(16) << "Ledger name" << setw(15) << "DR Amount" << setw(15) << "CR Amount" << endl;

        for (int j = 56; j > 0; j--)
        {
            out_file_trial << "-";
        }

    out_file_trial << endl;
}
```

This function design is creating the external Trial Balance file and entering the data into it by extracting data from external ledger files.

Functionalities

Home Page:



The project contains the following functionalities:

1. Making a journal general for the user:

The system will take input all the credit and the debit entries

From the users and their dates and will finally show the journal based on the journal entries that the users have entered

Taking user input:

```
ENTER DATE OF ENTRY:  31/12/2021

ENTER DEBIT ENTRY:  Dividend_Payable

ENTER DEBIT AMOUNT:  3000

ENTER CREDIT ENTRY:  Cash

ENTER CREDIT AMOUNT:  3000

ENTER DESCRIPTION:  Dividend Payable per Share

PRESS 1 TO EXIT 0 TO CONTINUE:  1
```

```
ENTER DATE OF ENTRY:  21/12/2021

ENTER DEBIT ENTRY:  Cash

ENTER DEBIT AMOUNT:  200000

ENTER CREDIT ENTRY:  Capital_Stock

ENTER CREDIT AMOUNT:  200000

ENTER DESCRIPTION:  10000 Shares sold at 200000

PRESS 1 TO EXIT 0 TO CONTINUE:  0
```

```
ENTER DATE OF ENTRY:  24/12/2021

ENTER DEBIT ENTRY:  A/C_Payable

ENTER DEBIT AMOUNT:  5000

ENTER CREDIT ENTRY:  Supplies

ENTER CREDIT AMOUNT:  5000

ENTER DESCRIPTION:  Signifies return of some supplies

PRESS 1 TO EXIT 0 TO CONTINUE:  0
```

Showing the journal ledger:

TESTING
JOURNAL
01/12/2021

DATE	DESCRIPTION	DEBIT	CREDIT
21/12/2021	Cash	200000	
	to Capital_Stock		200000
	(10000 Shares sold at 200000)		
21/12/2021	Supplies	50000	
	to A/C_Payble		50000
	(N/A)		
22/12/2021	Marketing_Expanse	4000	
	to Note_PAyable		4000
	(N/A)		
22/12/2021	Salaries_Expanse	10000	
	to Cash		10000
	(N/A)		
24/12/2021	A/C_Payable	5000	
	to Supplies		5000
	(Signifies return of some supplies)		
24/12/2021	Cash	40000	
	to Appollo(CAR)		40000
	(N/A)		
31/12/2021	Dividend_Payable	3000	
	to Cash		3000
	(Dividend Payable per Share)		

.

2. Making and showing the T accounts for journal entries:

The input entered in the journal is filtered and the t accounts are made on the basis of the entries in the journal.

The system also shows the individual t accounts for each of the element entered.

```
                                        TESTING
                                          Cash
                                        LEDGER
-------------------------------------------------------------------------------------------
Date          Particular          Amount      Date          Particular          Amount
-------------------------------------------------------------------------------------------
21/12/2021                Cash     200000|
                                         |22/12/2021                Cash           10000
24/12/2021                Cash      40000|
                                         |31/12/2021                Cash            3000
-------------------------------------------------------------------------------------------
     Total     227000
-------------------------------------------------------------------------------------------




                                      TESTING
                                   Capital_Stock
                                      LEDGER
-------------------------------------------------------------------------------------------
Date          Particular          Amount      Date          Particular          Amount
-------------------------------------------------------------------------------------------
                                        |21/12/2021       Capital_Stock          200000
-------------------------------------------------------------------------------------------
                                               Total          200000
-------------------------------------------------------------------------------------------
```

3. Storing the Trial and balance for the journal entries in files:

The system calculates the trial and balance from the journal file and creates a trial balance file that can be viewed later.

Journal File

TESTING
JOURNAL
01/12/2021

DATE	DESCRIPTION	DEBIT	CREDIT
21/12/2021	Cash	200000	
	to Capital_Stock		200000
	(10000 Shares sold at 200000)		
21/12/2021	Supplies	50000	
	to A/C_Payble		50000
	(N/A)		
22/12/2021	Marketing_Expanse	4000	
	to Note_PAyable		4000
	(N/A)		
22/12/2021	Salaries_Expanse	10000	
	to Cash		10000
	(N/A)		
24/12/2021	A/C_Payable	5000	
	to Supplies		5000
	(Signifies return of some supplies)		
24/12/2021	Cash	40000	
	to Appollo(CAR)		40000
	(N/A)		
31/12/2021	Dividend_Payable	3000	
	to Cash		3000
	(Dividend Payable per Share)		

Trial balance File

TESTING
TRIAL
01/12/2021

Ledger name	DR Amount	CR Amount
Cash	227000	
Capital_Stock		200000
Supplies	45000	
A/C_Payable	5000	
Marketing_Expanse	4000	
Note_PAyable		4000
Salaries_Expanse	10000	
Appollo(CAR)		40000
Dividend_Payable	3000	
TOTAL	294000	294000