```cpp
#include <iostream>
#include <string>
#include <fstream>
#include <iomanip>
using namespace std;
const int SIZE = 20;

void AddNew(fstream& data);
void ViewStatus(fstream& data, int);
void read(fstream& data, int&);
void payFee(fstream& data, int);
int search(int[]);
void deleteRecord(fstream& data, int);

struct info {
    string name[SIZE];
    int matno[SIZE];
    string course[SIZE];
    double fee[SIZE];
    double notPaid[SIZE];
    double paid[SIZE];
    int bilangan;
};
info student;  //declare struct

int main()
{
    int choice = 0;
    fstream data;


    do
    {

        system("cls");
        cout << "Welcome To The Maple High School Finance Website" << endl;
        cout << "__________________________________________________" << endl;
        cout << "__________________________________________________" << endl;

        cout << "PLEASE SELECT THE OPTION BELOW" << endl;
        cout << "1. Registration" << endl;
        cout << "2. View Financial Statement" << endl;
        cout << "3. Make Payment " << endl;
        cout << "4. Delete record" << endl;
        cout << "5. Exit " << endl;

        cout << endl << "Input your Choice : " << endl;
        cin >> choice;

        for (choice; choice < 1 || choice > 5;)
        {
            cout << "Invalid.Please enter a valid choice : ";
            cin >> choice;
        }
        switch (choice)
        {
        case 1: AddNew(data); break;
        case 2: read(data, student.bilangan);
            ViewStatus(data, student.bilangan); break;
        case 3: read(data, student.bilangan);
            payFee(data, student.bilangan); break;
        case 4: read(data, student.bilangan);
            deleteRecord(data, student.bilangan); break;
        default: cout << "Thank You :)" << endl;
        }

    } while (choice != 5);


    return 0;

}


void AddNew(fstream& data) //addnewname
{
    system("cls");
    cout << "---Add new student data---" << endl << endl;
    data.open("Database.txt", ios::out | ios::app);
    string name[SIZE];
    int matricno[SIZE], i = 0, select, j;
    double fee[SIZE];
    double paid[SIZE];
    double notPaid[SIZE];
    string course[SIZE];
    bool cont = false;
    char respond;


    do
    {
        cin.ignore();
        cout << "Please enter your name : ";
        getline(cin, name[i]);
        cout << endl;

        cout << "Please enter your matric number : ";
        cin >> matricno[i];
```

```cpp
        cout << endl;

        cout << " Please select your course " << endl;
        cout << "  1. Engine (Fee = RM 2000) " << endl;
        cout << "  2. Law (Fee = RM 2500) " << endl;
        cout << "  3. Econs (Fee = RM 1500) " << endl;
        cout << "  4. ICT (Fee = RM 1000) " << endl;
        cout << "  5. Medic (Fee = RM 3000) " << endl;
        cout << "  6. Education (Fee = RM 2600) " << endl;

        cin >> select;
        while (select < 1 || select > 6)
        {
            cout << "The course that you entered does not exist! Please re-enter : " << endl;
            cout << " 1. Engine" << endl;
            cout << " 2. Law" << endl;
            cout << " 3. Econs" << endl;
            cout << " 4. Ict" << endl;
            cout << " 5. Medic" << endl;
            cout << " 6. Education" << endl;
            cin >> select;
        }

        if (select == 1) {

            fee[i] = 2000;
            course[i] = "Engin";
        }
        else if (select == 2) {

            fee[i] = 2500;
            course[i] = "Law";
        }
        else if (select == 3) {

            fee[i] = 2500;
            course[i] = "Econs";
        }
        else if (select == 4){

            fee[i] = 1000;
            course[i] = "ICT";
        }
        else if (select == 5){

            fee[i] = 3000;
            course[i] = "Medic";
        }
        else{

            fee[i] = 2600;
            course[i] = "Edu";
        }


        paid[i] = 0;
        notPaid[i] = fee[i] - paid[i];
        i++;

        cout << "Do you wish to add another student ? (Y/N)" << endl;
        cin >> respond;

        switch (respond)
        {
        case 'y':
        case 'Y': cont = true; break;
        case 'n':
        case 'N' or 'n': cont = false; break;
        }

    } while (cont && (i < SIZE));

    j = i;



    for (i = 0; i < j; i++)
    {
        data << name[i] << '$' << matricno[i] << " " << course[i] << " " << fee[i] << " " << paid[i] << " " << notPaid[i] << endl;
    }
    cout << "Data Successfully Recorded " << endl;
    system("pause");
    data.close();

}

void read(fstream& data, int& totalist) {  //read from file
    data.open("Database.txt", ios::in);
    system("cls");
    string name[SIZE];
    int matno[SIZE];
    double fee[SIZE];
    string course[SIZE];


    int i = 0;
```

```cpp

    getline(data, student.name[i], '$');
    data >> student.matno[i] >> student.course[i] >> student.fee[i] >> student.paid[i] >> student.notPaid[i];
    while (data) {
        i++;
        data.ignore();
        getline(data, student.name[i], '$');
        data >> student.matno[i] >> student.course[i] >> student.fee[i] >> student.paid[i] >> student.notPaid[i];
    }
    totalist = i;
    data.close();
}

void ViewStatus(fstream& data, int i) { //financial statement

    cout << "---Financial status for all students---" << endl << endl;
    cout << setprecision(2) << showpoint << fixed;
    cout << setw(15) << "Name" << setw(23) << "Matric Number" << setw(13) << "Course" << setw(15) << "Fee" << setw(19) << "Amount Paid" << setw(19) << "\tAmount Not Paid" << endl;
    for (int j = 0; j < i; j++) { //print
        cout << endl;
        cout << (j + 1) << ". ";
        cout << setw(23) << left << student.name[j] << setw(15) << left << student.matno[j] << setw(13) << left << student.course[j] << "RM " << setw(14) << left << student.fee[j];
        cout << "RM " << setw(18) << left << student.paid[j] << "RM " << setw(13) << left << student.notPaid[j];
    }
    cout << endl;
    cout << "\n--------END OF THE LIST--------" << endl << endl;
    system("pause");
}


void payFee(fstream& data, int j) {
    data.open("Database.txt", ios::out); //rewrite the file
    double payment;
    cout << "----Make Payment----\n" << endl;
    int pos = search(student.matno);

    if (pos > -1) {
        cout << "Initiating payment for student...\n" << endl;
        cout << "\tName: " << student.name[pos] << endl;
        cout << "\tCourse: " << student.course[pos] << endl << endl;
        cout << setprecision(2) << showpoint << fixed;
        cout << "Total fee:       RM " << student.fee[pos] << endl;
        cout << "Outstanding fee: RM " << student.notPaid[pos] << endl;
        cout << "Amount to pay?:  RM ";
        cin >> payment;

        while (payment > student.notPaid[pos]) {
            cout << "Error!" << endl;
            cout << "The amount you entered is higher than the amount you need to pay." << endl;
            cout << "Please re-enter amount: ";
            cin >> payment;
        }
        char respond;
        cout << endl <<"You have entered RM " << payment << "\tConfirm? (Y/N)";
        cin >> respond;
        switch (respond) {
        case 'Y':
        case 'y':
            student.paid[pos] = student.paid[pos] + payment;
            student.notPaid[pos] = student.notPaid[pos] - payment;
            cout << "\nPayment successfully recorded" << endl << endl; break;
        case 'n':
        case 'N': cout << "Payment cancelled." << endl; break;
        }

    }
    else
        cout << "Sorry. Matric number entered not matching with any students." << endl;


    for (int i = 0; i < j; i++)
    {
        data << student.name[i] << '$' << student.matno[i] << " " << student.course[i] << " " << student.fee[i] << " " << student.paid[i] << " " << student.notPaid[i] << endl;
    }

    system("pause");
    data.close();
}

int search(int arrayMatricNum[]) { //return position of array
    int index = 0;
    int position = -1;
    bool found = false;
    int value;

    cout << "Please enter matric number of the student: ";
    cin >> value;
    while (index < SIZE && !found) {
        if (arrayMatricNum[index] == value) {
            found = true;
            position = index;
            cout << "Record found!\n" << endl;
        }
        index++;
    }
    return position;
}

void deleteRecord(fstream& data, int z) {
    cout << "---Delete student record from the system---" << endl << endl;
    cout << "Note: This is aplicable to student that had complete \n      their fee payment" << endl << endl;
    char respond;
    int pos = search(student.matno);

    if (pos > -1) {
        cout << endl;
        cout << "\tName: " << student.name[pos] << endl;
        cout << "\tCourse: " << student.course[pos] << endl << endl;

        if (student.notPaid[pos] == 0) {
            cout << "This student has pay full fees" << endl;
            cout << "Confirm to delete this student record? (Y/N): ";
            cin >> respond;
            switch (respond) {
            case 'Y': case 'y': {
                data.open("Database.txt", ios::out); //rewrite the file
                for (int i = 0; i < z; i++)
                {
                    if (i != pos) {
                        data << student.name[i] << '$' << student.matno[i] << " " << student.course[i] << " " << student.fee[i] << " " << student.paid[i] << " " << student.notPaid[i] << endl;
                    }
                    else
                        continue;   //<- debugging
                }
                cout << "\nSuccess.\nRecord deleted." << endl;
                data.close();
                student.bilangan = student.bilangan - 1;
            }break;

            case 'N': case 'n':
                cout << "Operation cancelled." << endl;
                break;
            }
```

```cpp
        }
        else {
            cout << "Cannot delete student data" << endl;
            cout << "This student still has outstanding fees." << endl;
        }

    }
    else {
        cout << endl;
        cout << "Sorry. Matric number entered not matching with any students." << endl;
        for (int i = 0; i < z; i++)
        {
            data << student.name[i] << '$' << student.matno[i] << " " << student.course[i] << " " << student.fee[i] << " " << student.paid[i] << " " << student.notPaid[i] << endl;
        }
    }

    cout << endl;
    system("pause");
}

```