Project Presenter Name: Sara Padula

Scope

District: 122

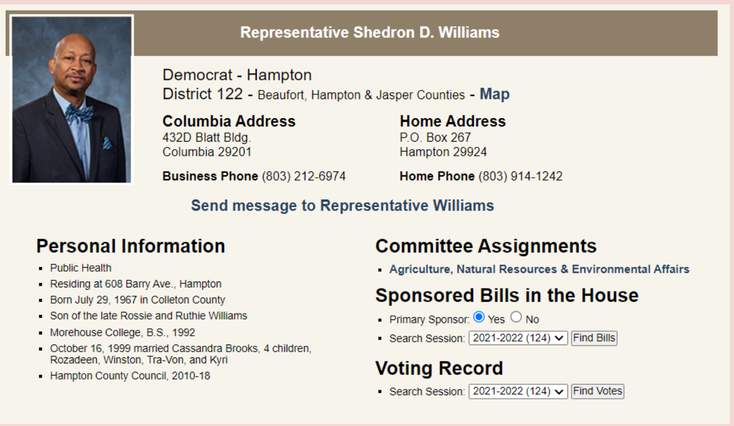
Representative: Shedron D. Williams

Language: C++

Advantages:	Disadvantages:
Portability	Use of Pointers
Object-oriented	Absence of Garbage Collector
Multi-paradigm	Absence of Built-in Thread

Data

What?



How?

Saved HTML code and parsed it

Queries Snapshot

- Differently formatted questions (can handle upper and lower case)

```
what is their name


Shedron D. Williams

What would you like to know about the representative of district 122?
Enter 'quit' or 'q' to exit the program.


I want to know all information


Name: Shedron D. Williams
Region: Beaufort, Hampton, & Jasper
Columbia Address: 432D Blatt Bldg.
Home Address: P.O. Box 267 Hampton 29924
Business Phone: (803) 212-6974
Home Phone: (803)914-1242
Eduction: Morehouse College, B.S., Public Health
DOB: July 29, 1967
Birth Place: Colleton County
Spouse: Cassandra Brooks,
Children: 4 children: Rozadeen, Winston, Tra-Von, and Kyri
Job Experience: Hampton County Council, 2010-18
Parents: late Rossie and Ruthie Williams
Committee: Agriculture, Natural Resources & Environmental Affairs
```

- Misspelled and random words handled

```
What would you like to know about the representative of district 122?
Enter 'quit' or 'q' to exit the program.


fanily


Parents: late Rossie and Ruthie Williams
Children: 4 children: Rozadeen, Winston, Tra-Von, and Kyri
Spouse: Cassandra Brooks,

What would you like to know about the representative of district 122?
Enter 'quit' or 'q' to exit the program.


blah blah


Invalid entry or we do not have that information. Try again.
```

- Video Link: https://github.com/sarapadula/sarapadula/tree/main/prog_assignments/Final%20Project/doc

Experience Implementing the Chat Bot

- Trial and Error
  - For many parts of the project, I implemented one idea then had to completely overhaul it in order to get the program to work
  - For example, in project 1 I attempted to extract the district information directly from the website, but after many attempts to implement this, I decided to copy the HTML code of the website to a txt file and extract the information from there instead
- Online help
  - For little additions to my project to make the chatbot work better I would refer to online forums to get ideas that I could implement into my project
  - For example, in project 3 I included a function toLowerCase() which would make any user query all lowercase so I did not have to worry about case-sensitivity. I got this idea and the subsequent code from an online forum.

Project 1: Extraction

Created a parser function

Initialized a char array and called the parser function

```
void parser(char* S)
{
    string filename("output.txt");
    ofstream file_out;
    file_out.open(filename, std::ios_base::app);
    int n = strlen(S);
    int start = 0, end = 0;

    for (int i = 0; i < n; i++) {
        if (S[i] == '>') {
            start = i + 1;
            break;
        }
    }
    while (S[start] == ' ') {
        start++;
    }
    for (int i = start; i < n; i++) {
        if (S[i] == '<') {
            end = i - 1;
            break;
        }
    }
    for (int j = start; j <= end; j++) {
        file_out << S[j];
    }

    file_out << "\t";
}
```

```
if (myfile.is_open())
{
    while (getline(myfile, line))

    {
        int size = 10000;
        char* str = new char[size];
        for (int i = 0; i <= line.length(); i++)
        {
            str[i] = line[i];

        }
        parser(str);

        delete[] str;

    }
    myfile.close();

}
else cout << "unable to open file";
```

Project 2: Processor

Prompted the user decide what category of information they wanted to know about

In each else if statement the output file from proj 1 is parsed to find that information

Project 4: User Intent Query Mapper

- Created a function that takes in the user's input and one of the hardcoded queries
- It compares the two strings and determines the percentage of matches between the two and returns the percentage
- If the percentage returned is greater than 70% then the query match is successful and the info is output to the console

```
double compareString(string str1, string str2)
{
    int n = 0;
    if (str1.length() < str2.length())
    {
        n = str1.length();
    }
    else
    {
        n = str2.length();
    }
    double equals = 0.0;
    double length = str1.length();
    double length2 = str2.length();
    double percentageOfMatches1 = 0.0;
    double percentageOfMatches2 = 0.0;

    for (int i = 0; i < n; i++)
    {
        if (str1[i] == str2[i])
        {
            equals++;
        }
    }
    percentageOfMatches1 = 100 * (equals / length);
    percentageOfMatches2 = 100 * (equals / length2);
    if (percentageOfMatches1 <= percentageOfMatches2)
    {
        if (percentageOfMatches1 > 100)
        {
            return 1;
        }
        else
        {
            return percentageOfMatches1;
        }

    }
    else
    {
        if (percentageOfMatches2 > 100)
        {
            return 1;
        }
        else
        {
            return percentageOfMatches2;
        }

    }
}
```

Project 5: Session Logger

```
void create(int user, int computer, double time, string filename, string countfile)
{
    fstream fout;
    fout.open("data\\chat_statistics.csv", ios::out | ios::app);
    int number;
    int iterations = 0;

    ifstream input_file(countfile);
    while (input_file >> number)
    {
        iterations = iterations + number;

    }

    input_file.close();

    fout << iterations << ","
        << filename << ","
        << user << ","
        << computer << ","
        << time << "\n";
}
```

```
struct ChatSession
{
public:
    ChatSession(
        int chatid,
        string chatFile,
        int user,
        int comp,
        int time
    ) {
        ChatId = chatid;
        ChatFile = chatFile;
        User = user;
        Comp = comp;
        Time = time;

    }
    void display()
    {
        cout << "Chat Id: " << ChatId << endl;
        cout << "Chat File: " << ChatFile << endl;
        cout << "User Chats: " << User << endl;
        cout << "Computer Chats: " << Comp << endl;
        cout << "Time Elapsed: " << Time << endl;
    }

    int ChatId;
    string ChatFile;
    int User;
    int Comp;
    int Time;
};
```

```
int chatsession = 0;
cin >> chatsession;
vector<ChatSession> chats;
ifstream inputFile;
inputFile.open("data\\chat_statistics.csv");
string line = "";
getline(inputFile, line);
line = "";

while (getline(inputFile, line))
{
    int chatId;
    string chatFile;
    int userCount;
    int compCount;
    int time;
    string tempString = "";


    stringstream inputString(line);

    getline(inputString, tempString, ',');
    chatId = atoi(tempString.c_str());
    getline(inputString, chatFile, ',');
    tempString = "";
    getline(inputString, tempString, ',');
    userCount = atoi(tempString.c_str());
    tempString = "";
    getline(inputString, tempString, ',');
    compCount = atoi(tempString.c_str());
    tempString = "";
    getline(inputString, tempString, ',');
    time = atoi(tempString.c_str());

    ChatSession chat(chatId, chatFile, userCount, compCount, time);
    chats.push_back(chat);

    line = "";
}
chatsession = chatsession + 1;
cout << "Chat ID: " << chats[chatsession].ChatId << endl;
cout << "Chat File: " << chats[chatsession].ChatFile << endl;
cout << "User Chats: " << chats[chatsession].User << endl;
cout << "Computer Chats: " << chats[chatsession].Comp << endl;
cout << "Time Elapsed: " << chats[chatsession].Time << " Seconds" << endl;
```