21CS11- C Programming

Programming Assignment [Group]

Electoral Commission Project- E-Systems

Submission Due: Week 15 (Sunday Before 12pm IST)

Weightage: 10%

Aim

This is a team project (groups of 4- Encouraged to stick in same team chosen for assignment 1) and worth 10% of your final grade. It intends to evaluate your understanding and practical skills on C programming skills with the knowledge of expressions, conversions, control statements, arrays, functions, pointers, structures and files. You are encouraged to do additional self-exploration for functions that are not covered in the lectures. You are required to complete all the tasks below.

In this Assignment, you are required to implement a practical application based on Tamilnadu State electoral commission E-system. You have to design and program an E-system, where the assessment involves

- **project brief**
- **project solution (codes) – Instructions are provided below (Task 1 & task 2)**
- **project demonstrate/presentation**
- **final project report along with a peer assessment.**

Assignment Background

Assignment comprises of two task(s) which are as follows:

Task 1: Create a primary database

You will have a database with two primary tables: voter table and candidate table as shown below. You may store the data in a .txt file (with that all the field values will be stored as string in the text file). You can treat each table as a text file (for eg. candidate table is stored in candidate.txt). You may create additional tables or fields if required.

To start with, create a file and name it as **addcandidate.c**. It should have the following functions:

- Menu options ([1] Add Candidate, [2] View Candidates, [3] Exit). User may choose the options by entering 1, 2 or 3.
- For menu option 1, create a simple input function that allows the user to add the candidate who will be standing in the election. Perform validation to ensure that the candidate's record does not exist in the txt file. Store the candidate's information into the Candidate table. Add at least 12 different candidates (3 for each division).
- For menu option 2, display all the candidates' ID and names along with party names and division.
- Menu option 3 is to allow the user to exit the program gracefully.

* Remember to perform the necessary validation checks on the input entered by the user.

Voter Table

Field Name	Data Type	Description
Voter ID	Char	Login ID for Voter (voter's name without whitespaces)
Name	Char	Name of the Voter
Age	Integer	
Division	Integer	1 – 4 (division 1-division 4)
Status	Char	Indicate whether the voter has made the vote. Default valueis "N". Change to "Y" if the voter has

Candidate Table

Field Name	Data Type	Description
Candidate ID	Char	Unique ID assigned to each candidate programmatically. It should be the first 3 letter of the party name followed by 2 digits (eg if the party is Alliance party, the ID should be ALL01)
Name	Char	Name of the candidate. Must be unique
Party	Char	Assign 3 random party names (all these 3 parties will stand in each division
Division	Integer	1 – 4 (division 1-division 4)
Count	Integer	The total number of votes

Task 2 — Voting module

In this module, you are to create the function to allow the public from different division (1- 4) to vote. Name the file as **votecandidate.c**

Your program should display a menu options to the user as follows:

1. View candidates

This option allows the user to choose whether to view all the candidates in all the divisions or view the candidates in specific division (prompt the user to enter the division) or view the candidates based on the party (prompt the user to enter the party). You are to create another menu for this. Display all the fields in the candidate table.

2. Register voter

Similar to task 1, this option allows the user to register as a voter and store the information in the voter table. If the voter's table (aka txt) is not available, then create it. Each voter is not allowed to register twice. The eligibility age for voter must be 18 years old and above.

3. Vote

This function allows the user to vote by first entering the voter's ID. Ensure that he/she is a registered voter and has not voted yet. Then display the candidates' information (Candidate ID, Name, Party) who is standing in same division as the voter.

The voting rules are:

- ✓ User must be a registered voter to vote
- ✓ The voter is only allowed to vote for a candidate in his/her division.

- ✓ A voter can only vote once

To vote, the voter will be required to enter the Candidate ID. Update the candidate and voter tables accordingly after the voter has voted.

4. View voting results and summary

Similar to option [1] above, this option allows the user to choose to

a) View the voting results of all the candidates from all the division Provide a summary of the total number of votes in all the divisions. Identify and display the candidate (ID, name, party, number of votes, percentage) with the highest votes and the candidate with the lowest votes in each division.

b) View the candidates in specific division. (Prompt the user to enter the division) Provide a summary of the total number of votes in the specific division. Identify and display the candidate (ID, name, party, number of votes, percentage) with the highest votes and the candidate with the lowest votes in that division.

5. Quit

Perform the necessary input validation. The program should only accept valid choices from the user, both upper and lowercase selections should be allowed.

- If an illegal choice is made, you should display, "Invalid selection, please try again "and the menu options should be displayed again.
- If the table is empty you should display, "No voters yet. Please try again later"
- When the user quits the program, display 'Goodbye" before terminating the program
- Ensure that the program does not crash at any point

Divide your work equally among your team member. Every member is expected to code some parts of the project. Set the programming standards to be used among your team members.

Before you begin, plan properly. Write out the steps you need to take and decide in what order they should be done. Think about what loops you should use as well as what you will use for your selection logic. Your creativity and assumption of rules are encouraged to complete the tasks

This assignment can be challenging to some! It may likely take a few attempts before you complete it -- that's normal! Most importantly, enjoy the process!!

Finally, be sure to test your program as you go and at the end.

Assignment Requirements

This assignment must be written in C

- Your code must have appropriate comments including your name and student number, the name of the .c file, the purpose of the program, brief explanations of variables and explanations of any code, which is not obvious to another programmer.
- Include a block (multiline) comment summarising the input, output and local variables used in your program.
- Include a block comment stating any equations, and test data.

Submission requirements for tasks:

1. Project report: Write a short report, which illustrates your program design (algorithm or flowchart, identification of variables, constants), short explanation of your program and include evidence of testing – screen shots. You are also to include the task assigned to each member.
2. Working solutions (.c and .txt)
3. To be submitted by the team leader

Plagiarism not applicable for me

The submitted assignment must be Teamwork, and any parts that are not created by you must be properly referenced. Plagiarism is treated very seriously at Department of AI and DS, Coimbatore Institute of Technology. It includes submitting the code and/or text copied from other students, the Internet or other resources without proper references. Allowing others to copy your work is also plagiarism. Please note that you should always create your coding even if you have very similar ideas with other students.

I did on my own