AIM OF THE PROJECT:

The aim of this code is to implement an Invoice management app with login system along a graphical interface for proper functionality. It utilizes the concepts of File Handling and Structure in C programming language.

Group members:

23EE01004: - Anand Roy

23EC01005: - Aum Vyas

23ME01003: - Aditya

23ME01044: - Naman

Important Highlights of the project:

- User Interface using time and console related header files.
- File system integration for storing data in a separate file, using binary and text file format.

Building blocks of the code:

main():
The function controls the entire flow of code and prompts user for inputs depending on required task.

pass():

This function provides the login page and introductory UI for the code, it does not need any input or parameters to initialize.

It prints the following:

- Intro page
- Member details
- Project title screen
- Login page for "employees"
- Redirects to main menu interface

clearscreen():

As the name suggest clears the console for printing new content

Generate functions:

They generate the body of the invoice; they are divided into 3 parts for printing head body and footer of the invoice respectively.

Time related functions():

Delay(), delayy(), tm() for incurring time delayed element printing for typewriter effect

tlog():

For saving the time when invoice management system was opened.

endingscreen():

For printing the ending screen banner.

How these block work together:

The code proceeds as follows:

- Inside the main function it initiates the pass() function, which prompts the user for credential information, which if not passed will close the application abruptly, but if passed will store the time of login into a log file.
- After this the user is greeted with menu screen and will be prompted with options to proceed with various options.
- In the first option the customer's name is entered along with the items they need, this is done by clearing the screen and displaying the menu by calling appropriate function, once it's done the invoice is displayed along with the amount the customer needs to pay.
- After it is done, we prompt the user to confirm whether the invoice needs to be saved, if confirmed yes. A file system is created, and the respective file is opened or created if already there, and the structure holding the order details is saved in that file.
- After it is saved the user is prompted whether they want to continue or not, if yes, they are sent in the main menu.
- In the second option, the available invoices are displayed that were saved in the file.
- In the third option, an invoice with certain customer name is searched and displayed.
- In the fourth option, all log information is shown.
- And after the user is done with the code, it ends with a banner sliding in an infinite loop across the screen.

Work assigned:

Anand Roy: Skeleton code, Intro display, File system integration for storing invoice, log file system, documentation.

Aum Vyas: UI For main menu, Ending screen banner, Administrator menu.

Aditya: Premium members database

Naman: Food item menu integration, Premium members menu

Areas of improvement:

Had been given more time we, definitely, would have worked on user interface using gtk header file, which for some reason couldn’t be loaded for windows-based compiler and worked on adding even more features than this app currently has.

What did you learn from this exercise?

We learnt about using file system in c, using which we can and how one can use console-based codes for making the output screen look presentable.