TYPING TUTOR

A

Mini Project Report

Submitted in partial fulfillment of the

Requirements for the award of the Degree of

BACHELOR OF ENGINEERING

IN

INFORMATION TECHNOLOGY

By

NAVEEN MANDA 1602-19-737-142

SIDDHARTH JETLING 1602-19-737-172

Department of Information Technology

Vasavi College of Engineering (Autonomous)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad-31

2020

Vasavi College of Engineering (Autonomous)

(Affiliated to Osmania University)

Hyderabad-500 031

Department of Information Technology

DECLARATION BY THE CANDIDATE

We, **NAVEEN MANDA** and **SIDDHARTH JETLING,** bearing hall ticket numbers, **1602-19-737-142** and **1602-19-737-172**, hereby declare that the project report entitled **"TYPING TUTOR"** is submitted in partial fulfillment of the requirement for the award of the degree of **Bachelor of Engineering** in **Information Technology.**

This is a record of bonafide work carried out by us and the results embodied in this project report have not been submitted to any other university or institute for the award of any other degree or diploma.

Naveen Manda

1602-19-737-142

Siddharth Jetling

1602-19-737-172

(Faculty In-Charge) (Head, Dept of IT)

AKNOWLEDGEMENTS

We take this opportunity to express our gratitude to everyone who helped us in making this project a success. Primarily we would like to thank our college for giving us this wonderful opportunity to work on a project. This project helped us learn many things.

We would like to thank Ms L. Divya, Assistant Professor, Vasavi College of Engineering; whose valuable guidance and support helped us in various phases of completion of this project.

We would like to thank all our friends who supported us in every stage of project.

Naveen Manda
1602-19-737-142
Siddharth Jetling
1602-19-737-172

ABSTRACT

Having a good command over typing and reading comprehension can turn out to be really helpful for any professional or student. Our aim is to create a tool which helps the user to enhance these skills. A tool which will not only show you how well you performed, but also tracks it and gives you relevant feedback. Moreover you can compete with others and look at their performances as well. This tool includes multiple logins User has to login with one's name and when the same user is logging in several times, his/her performance is tracked and a feedback is given accordingly.

TABLE OF CONTENTS

1.	**Introduction**		**1**
	1.1	About the project	1
	1.2	Domain	1
	1.3	Features	1
2.	**Technology**		**3**
	2.1	Software Requirements	3
	2.2	Hardware Requirements	3
3.	**Proposed Work**		**4**
	3.1	Design	4
		3.1.1 Use Case Diagram	4
		3.1.2 User Use Cases	4
		3.1.3 Admin Use Cases	5
		3.1.4 Competitor Use Cases	6
	3.2	Implementation	7
		3.2.1 Module-Wise Code	7
		3.2.2 Github/Folder Structure	62
	3.3	Testing	63
		3.3.1 User Test Cases	63
		3.3.2 Admin Test Cases	66
		3.3.3 Competitor Test Cases	68
4.	**Results**		**71**
		4.1.1 User Test Cases	73
		4.1.2 Admin Test Cases	85
		4.1.3 Competitor Test Cases	92
5.	**Additional Knowledge Acquired**		**99**
6.	**Conclusion and Future Work**		**100**
7.	**References**		**101**

1. INTRODUCTION

1.1 Problem Domain

With times, efficiency is something that has become a compulsory skill everyone one is expected to have. With the speeding world, it has become necessary for us to become quick at work. Two such skills which demand speed and accuracy are typing and reading speed on which our main emphasis was, for this mini project. Using this tool we have developed, its users can get better at these skills. To put it short, this is a **Skill Development tool** where users can learn in a fun way.

1.2 About the project

Basically the core functionalities of this tool are to calculate Typing and reading speed, and extending to some interactive games. Having a typing speed over 150 WPM needs a lot of practice. This tool help you build a foundation to reach incredible typing speeds and excellent accuracy by giving you a feedback every time which is based on your speed and accuracy.

Having good reading skills is of prime importance these days. This tool also helps you build excellent reading skills by increasing your reading speed and also reading comprehension. Your reading comprehension is determined by taking a small quiz after you complete your reading and based on both parameters it gives you a rating which determines your reading skills.

Mastering touch typing will be an easy thing to achieve with the help of this tool. Interactive games present in this tool boost up your reaction speed and help you to master touch typing.

1.3 Features

Our project “Typing Tutor” has different features which primarily focus to help its users develop their typing speed, reading and comprehension skills. The project has these below mentioned features.

1.3.1 Typing speed test

This is our core feature. Here, paragraph randomly taken from text file are displayed and user is expected to type in the text displayed. This activity is time bound. This helps user to

practice on their typing speed and accuracy. Typing speed and accuracy is calculated after the timer ends. This information is stored in a leader board and rankings are displayed.

1.3.2 Reading speed test

Reading speed and comprehending what one reads is also very important. This is where our tool can turn out to be really helpful. A paragraph randomly taken from the text files and expects user to complete reading the paragraph. After reading the paragraph a quiz is displayed to test user's reading comprehension.

1.3.3 Games

Couple of games/challenges is provided in the tool, which indirectly helps you work on your typing and reading speed

1.3.4 Admin features

Admin has a set of features like, viewing user credentials, deleting user, deleting leader boards and viewing particular user's credentials. These let admin do various things.

1.3.5 Leader board

Leader board is accessible to both admin and users. Options given are Typing and Reading leader boards. User/Admin can select one and view that particular leader board.

2. TECHNOLOGY

2.1Software Requirements

Our project "Typing Tutor" was completely built in C language. We used different types of IDE's for building this project which include code blocks and virtual studio which gave us a huge head start in building this project. We have built and tested this project on windows operating system (Windows 10).

In order to use Typing Tutor, one should have the following:

- **Operating System**: Windows 7 and above
- **C Compiler**: GNU Compiler Collection (GCC)
- **Editor**: Any text editor (Code blocks/Visual studio)

2.2Hardware Requirements

The hardware requirements for our project were limited to a PC with adequate specifications. We have built and tested this project on an x64-based PC running on an Intel core i5 processor with 8 GB of ram. Undoubtedly we are pretty sure that this project can be built and run comfortable on a lower end PC having minimum specifications: x32-based PC, Intel Pentium processor, 2 GB ram.

In order to use Typing Tutor, one should have the following:

- **System:** x32-based or x64-based
- **Processor:** Intel Core i3 and above / Intel Pentium
- **Memory:** Above 4 GB RAM

3. PROPOSED WORK

3.1 DESIGN

3.1.1 Use Case Diagram

3.1.2 USER USE CASES

3.1.2.1 Register

A new user has to register to be able to login and access all the features the tool provides. This use case basically lets new users to register by entering their username, password, age and gender. After successful registration an account s created and the credentials are stored into a file.

3.1.2.2 Login

A registered user can login using their username and password which they set at the time of registration. After a successful login, Main Menu is displayed where the features tools are displayed.

3.1.2.3 View Leader board

User can select the option on menu, “View Leader board” which lets him/her view both the

leader boards (i.e., typing speed and reading speed leader boards)

3.1.2.4 Typing Speed Test

A paragraph randomly taken from a text file is displayed. The user has to type in the text displayed. Typing speed and accuracy are calculated according to the performance of the user and are stored in leader board.

3.1.2.5 Reading speed Test

A paragraph randomly taken from a text file is displayed. The user has to type in the text displayed. Reading speed and comprehension are calculated according to the performance of the user and are stored in leader board.

3.1.3 ADMIN USE CASES

3.1.3.1 Login

Admin can login in to the tool using hardcoded admin credentials. After a successful login, Admin menu is displayed on screen. Admin can select any potion and can access the features intended for admin.

3.1.3.2 View Leader board

Admin can select the option on menu, “View Leader board” which lets him/her view both the leader boards (i.e., typing speed and reading speed leader boards)

3.1.3.3 User Credentials

Admin is given access to all the user credentials (the ones that users enter while registering an account). In addition, admin can search for just particular user in the whole file where credentials are stored.

3.1.3.4 Delete User

Incase if Admin wants to delete a user, he can just select “Delete User” option and enter a valid username. The particular user is deleted from all the data files.

3.1.3.5 Delete Leader board

Admin can delete any particular or all the leader boards that are there. This act clears data in all the selected leader boards.

3.1.4 COMPETITOR USE CASES

Competitors come into picture only when Multi Player Mode is selected in games section. If single player mode is selected, the user can just play the two available games (as mentioned in **3.1.4.1**)

3.1.4.1 Games

Here, User is given option to select the mode, either single or multi player mode. According to the mode selected, games are displayed. In single player mode, competitor plays game and his/her result is stored in a file whereas in Multi Player Mode, Competitor takes turns and participate in challenges, the final individual scores are displayed and winner is displayed. These scores are not stored anywhere

3.1.4.2 Join With Names

Competitor will have to join with any name of their choice before playing game. No two competitor's names should match.

3.1.4.3 Participate in challenges

Competitors after joining with their names, takes turn and participate in the challenges/games.

3.1.4.4 View Result

After every Competitor has completed their turn, their individual score are displayed on leader board and winner among them will be declared.

3.2 IMPLEMENTATION

3.2.1 Module-Wise Code

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
#include<time.h>
#include<windows.h>

COORD coord = {0, 0};
// int totalLines = 0;

void userRegistration();
void userLogin();
void adminLogin();
void mainLoadingScreen();
void mainMenu(char UserName[]);
void adminMainMenu();
void adminUserView();
void homePage();
void Typingspeed(char UserName[]);
void Add_To_Typing_Leaderboard(float s,float a,char UserName[]);
void Display_Typing_Leaderboard();
void Readingspeed(char UserName[]);
double Readingcomprehension(int randomNumber);
void Add_To_Reading_Leaderboard(double reading_speed, double reading_comprehension, char UserName[]);
void Display_Reading_Leaderboard();
void viewLeaderBoards();
void Delete_User(char  filename[],char userName[]);
void Delete_User_credentials(char  filename[],char userName[]);
void clear_Leaderboard(char filename[]);
void compete();
void typingCompetition(int typing_duration,float *speed , float *accuracy);
void readingCompetition(float *speed , float *accuracy);
void games();
void gameMenu(char userName[]);
void game1(char userName[], int level, char singleOrNot[]);
void game2(char userName[], int level, char singleOrNot[]);
void ranArray(int randomisedArray[],int n);
int is_Registered_user(char name[], char filename[]);
void display_Feedback(char name[], double present_Typing_speed, double present_Typing_accuracy, char filename[]);
void arrow(int rel,int arrowPosition);
int levelSelector();
```

```c

struct user
{
    char userName[20];
    char userPassword[20];
    int userAge;
    char gender[8];
    char gameWord[15];
};

struct player
{
    float typing_speed;
    char name[20];
    float typing_accuracy;
    float reading_speed;
    float reading_comprehension;
    int gameScore;
};

void gotoxy(int x, int y)
{
    coord.X = x; coord.Y = y; // X and Y coordinates
    SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), coord);
}

int main()
{
    system("color 71");
    mainLoadingScreen();
    homePage();
    return 0;
}

void mainLoadingScreen()
{
    system("cls");

    gotoxy(45,10);printf("*************************");
    gotoxy(45,11);printf("*\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB*");
    gotoxy(45,12);printf("*\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB *");
    gotoxy(45,13);printf("*\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB  *");
    gotoxy(45,14);printf("*\xDB\xDB\xDB        \xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB   *");
    gotoxy(45,15);printf("*\xDB\xDB\xDB        \xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB    *");
    gotoxy(45,16);printf("*\xDB\xDB\xDB\xDB\xDB\xDB\xDB  \xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB     *");
    gotoxy(45,17);printf("*\xDB\xDB\xDB\xDB\xDB\xDB\xDB  \xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB      *");
    gotoxy(45,18);printf("*\xDB\xDB\xDB\xDB\xDB\xDB\xDB  \xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB       *");
    gotoxy(45,19);printf("*\xDB\xDB\xDB\xDB\xDB\xDB\xDB  \xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB        *");
    gotoxy(45,20);printf("*\xDB\xDB\xDB\xDB\xDB\xDB\xDB  \xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB         *");
    gotoxy(45,21);printf("*\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB
```

```
void homePage()
{
    system("cls");
    int position = 1;
    char exitPermission;        //for knowing if user wants to exit current page
    int keyPressed = 0;

    while(keyPressed != 13)
    {
        system("cls");

        gotoxy(30,4);
        printf("\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xD
        gotoxy(30,5);
        printf("\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xD
        gotoxy(30,6);
        printf("                                                    ");
        gotoxy(30,7);
        printf("                    HOME                    ");
        gotoxy(30,8);
        printf("                                                    ");
        gotoxy(30,9);
        printf("\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xD
        gotoxy(30,10);
        printf("\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xD

        gotoxy(30,13);
        printf("-----------------------------------------------------");

        //arrow function prints arrow depending on the up/down key pressed by user
        gotoxy(30,15);
        printf("\t\t"); arrow(1,position); printf("REGISTER");
        gotoxy(30,16);
        printf("\t\t");  arrow(2,position); printf("LOGIN");
        gotoxy(30,17);
        printf("\t\t"); arrow(3,position); printf("ADMIN LOGIN");
        gotoxy(30,18);
        printf("\t\t"); arrow(4,position); printf("GAMES");
        gotoxy(30,19);
        printf("\t\t"); arrow(5,position); printf("EXIT");

        gotoxy(30,21);
        printf("-----------------------------------------------------");

        gotoxy(30,24);
```

```
        gotoxy(30,24);
        printf("\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xD
        gotoxy(30,25);
        printf("\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xD
        gotoxy(30,26);
        printf("\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xD
        gotoxy(30,27);
        printf("\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xD
        keyPressed = getch();


        if(keyPressed == 80 && position != 5)        //increment value of position if down arrow key is pressed
            position++;
        else if(keyPressed == 72 && position != 1)     //decrement value of position if up arrow key is pressed
            position--;
        else                                            //no change in value of position if any other key rather than up/down key is pressed
            position = position;
    }

    switch (position)
    {
        case 1: userRegistration();
                break;
        case 2: userLogin();
                break;
        case 3: adminLogin();
                break;
        case 4: games();
                break;
        case 5:
                gotoxy(35,29);printf("You want to log out?(y/n)");
                exitPermission = getch();
                if(exitPermission == 'y')
                {
                    gotoxy(35,31);printf("Exiting...");
                    Sleep(500);
                    exit(0);
                }
                else homePage();

                break;
        default: gotoxy(35,29);printf("Invalid!");
                break;
    }
}
```

```
void mainMenu(char UserName[])
{
    system("cls");
    int position = 1;
    int keyPressed = 0;
    char exitPermission;
    while(keyPressed != 13)
    {
        system("cls");

        gotoxy(30,4);
        printf("\xDB\xDB\xDB
```

```
void adminMainMenu()
{
    system("cls");
    int position = 1;
    char exitPermission;
    int keyPressed = 0;
    while(keyPressed != 13)
    {
        system("cls");

        gotoxy(30,4);
        printf("\xDB\xDB\xDB
```

```
        printf("\xDB\xDB\xDB
```

```
void arrow(int rel,int arrowPosition)          //just prints arrow on menu pages when parameters are equal
{
    if(rel == arrowPosition)
    {
        printf(" ---> ");
    }
    else
    {
        printf("      ");
    }
}

void userRegistration()
{
    system("cls");
    gotoxy(35,6);
    printf("\xB2\xB2\xB2
```

```
while((ch = getc(flines)) != EOF)
{
    if(ch == '\n')
    {
        totalLines++;
    }
}

FILE *fr;
fr = fopen("usernames.txt","r");

for(int i=0 ; i<totalLines ; i++)
{
    fscanf(fr, "%s %s %d %s",u[i].userName,u[i].userPassword,&u[i].userAge,u[i].gender);
    if(strcmp(u[i].userName,UserName) != 0)          //checking if the username entered already exits in the file
    {
        notEqual++;
    }
    else
    {
        break;
    }
}

if(notEqual != totalLines)        //if username is already taken, system prompts to try some other usernames
{
    gotoxy(35,20);printf("UserName is already taken, try something different");
    Sleep(2000);
    userRegistration();
}
fclose(fr);

password:
gotoxy(37,10);printf("Enter password:");
gotoxy(55,10);scanf("%s",UserPassword);

if(strlen(UserPassword) < 8)          //checking if password is less than 8 characters
{
    gotoxy(35,20);printf("Password must contain atleast 8 characters");
    gotoxy(55,10);printf("          ");
    Sleep(2000);
    goto password;
}
gotoxy(35,20);printf("                                                    ");
userage:
```

```
    gotoxy(37,12);printf("Enter Age:");
    gotoxy(55,12);printf("          ");
    gotoxy(55,12);scanf("%s",UserAge);
    int userAge;
    if(atoi(UserAge) && (atoi(UserAge) >= 1 && atoi(UserAge) <151))        //checking if age is in permisible range
    {
        userAge = atoi(UserAge);
    }
    else
    {
        gotoxy(30,20);printf("Age must be in the limits of 1-150, please enter valid age");
        Sleep(2000);
        gotoxy(30,20);printf("                                                                  ");
        goto userage;
    }

    usergender:
    gotoxy(37,14);printf("Enter Gender(select one option):");
    gotoxy(37,15);printf("1. Male 2. Female 3. Others");
    gotoxy(55,17);printf("          ");
    gotoxy(55,17);scanf("%s",UserGender);
    if(atoi(UserGender))
    {
        if(atoi(UserGender) == 1)
        {
            strcpy(userGenderString,male);
        }
        else if(atoi(UserGender) == 2)
        {
            strcpy(userGenderString,female);
        }
        else if(atoi(UserGender) == 3)
        {
        strcpy(userGenderString,others);
        }
        else          //for invalid gender input
        {
            gotoxy(30,20);printf("     Enter a valid option for Gender");
            Sleep(1000);
            gotoxy(30,20);printf("                                                                  ");
            goto usergender;
        }
    }
    else          //for invalid gender input
    {
    {
        gotoxy(30,20);printf("     Enter a valid option for Gender");
        Sleep(1000);
        gotoxy(30,19);printf("                                                                  ");
        goto usergender;
    }



    FILE *fua, *fpa;
    fua = fopen("usernames.txt","a");

    //updating the file that contains all the user details
    fprintf(fua,"%s %s %d %s\n",UserName,UserPassword,userAge,userGenderString);

    fclose(fua);
    gotoxy(37,20);printf("Account is successfully registered");
    Sleep(2000);
    homePage();
    return;
}
```

```
566 void userLogin()
567 {
568     int count = 0;
569     start:
570     system("cls");
571     gotoxy(35,14);printf("User Login: ");
572     gotoxy(35,15);
573     printf("\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\x
574     printf("\xB2");gotoxy(75,15);printf("\xB2\n");
575     gotoxy(35,16);
576     printf("\xB2");gotoxy(75,16);printf("\xB2\n");
577     gotoxy(35,17);
578     printf("\xB2");gotoxy(75,17);printf("\xB2\n");
579     gotoxy(35,18);
580     printf("\xB2");gotoxy(75,18);printf("\xB2\n");
581     gotoxy(35,19);
582     printf("\xB2");gotoxy(75,19);printf("\xB2\n");
583     gotoxy(35,20);
584     printf("\xB2");gotoxy(75,20);printf("\xB2\n");
585     gotoxy(35,21);
586     printf("\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\x
587     int notEqual = 0,totalLines = 0;
588     struct user u[200];
589     char UserName[30],UserPassword[30];
590     gotoxy(37,17);printf("Enter a UserName: ");
591     gotoxy(55,17);scanf("%s",UserName);
592
593     FILE *flines;
594     flines = fopen("usernames.txt","r");
595
596     char ch;
597     while((ch = getc(flines)) != EOF)
598     {
599         if(ch == '\n')
600         {
601             totalLines++;
602         }
603     }
604
605     FILE *fr;
606     fr = fopen("usernames.txt","r");
607
608     for(int i=0 ; i<totalLines ; i++)
609     {
610
```

```
    fscanf(fr,"%s %s %d %s",u[i].userName,u[i].userPassword,&u[i].userAge,u[i].gender);
    if(strcmp(u[i].userName,UserName) != 0)
    {
        notEqual++;            ///keep incrementing until the username entered is found in the data file
    }
    else
    {
        break;
    }
}

char password[30];
int l=0;

if(notEqual != totalLines && count<3)
{

    gotoxy(37,19);printf("Enter password:");
    gotoxy(55,19);

    while(ch != 13)     //to print '*' when entering password
    {
        fflush(stdin);
        ch = getch();
        if(ch != 13 && ch != 8)     //printing '*' when a character other than enter and backspace is entered
        {
            printf("*");
            password[l] = ch;
            l++;
        }
        else if(ch == 8 )
        {
            if(l>0)
            {
                l--;
                password[l] = '\0';
                printf("\b \b");        //when user enters backspace
            }
        }
        if(ch == 13)
        {
            break;
        }
    }
    strcpy(UserPassword,password);
```

```
        strcpy(UserPassword,password);
        if(strcmp(UserPassword,u[notEqual].userPassword) != 0)
        {
            gotoxy(39,23);printf("Invalid password!");
            Sleep(1500);
            gotoxy(39,23);printf("Please try again!");
            gotoxy(55,19);printf("          ");
            Sleep(1000);
            count++;
            if(count == 3)      //allows only three invalid attempts to login
            {
                goto counting;
            }
            goto start;
        }
        else
        {
            gotoxy(39,23);printf("Successful login!");
            Sleep(2000);
            mainMenu(UserName);
        }
    }

    else
    {
            gotoxy(39,23);printf("Invalid Username!");
            Sleep(2000);
            userLogin();
    }
    counting:
    if(count == 3)
    {
        gotoxy(35,23);printf("Too many attempts! Please try again!");
        Sleep(2000);
        homePage();
    }

    fclose(fr);
}
```

```
695 void adminLogin()
696 {
697     int count = 0;
698     start:
699     system("cls");
700     gotoxy(35,14);printf("Admin Login: ");
701     gotoxy(35,15);
702     printf("\xB2\xB2\xB2
```

```
            ch = getch();
            if(ch != 13 && ch != 8)     //printing '*' when a character other than enter and backspace is entered
            {
                printf("*");
                password[l] = ch;
                l++;
            }
            else if(ch == 8)            //when user enters backspace
            {
                if(l>0)
                {
                    l--;
                    password[l] = '\0';
                    printf("\b \b");
                }
            }
            if(ch == 13)
            {
                break;
            }
        }
        strcpy(adminPassword,password);
        if(strcmp(adminPassword,adminPass) != 0)
        {
            gotoxy(39,23);printf("Invalid password!");
            Sleep(1500);
            gotoxy(39,23);printf("Please try again!");
            gotoxy(55,10);printf("          ");
            Sleep(1000);
            count++;
            if(count == 3)          //allows only three invalid attempts to login
            {
                goto counting;
            }
            goto start;
        }
        else
        {
            gotoxy(39,23);printf("Successful login!");
            Sleep(2000);
            adminMainMenu();
        }
    }
    else
    {
        gotoxy(39,23);printf("Invalid Username!");
        Sleep(2000);
        adminLogin();
    }
    counting:
    if(count == 3)
    {
        gotoxy(39,23);printf("Too many attempts! Please try again!");
        Sleep(2000);
        homePage();
    }
}
```

```c
void adminUserView()
{
    system("cls");
    struct user u[200];
    FILE *fview;
    fview = fopen("usernames.txt","r");
    int totalLines = 0;

    char ch;
    while((ch = getc(fview)) != EOF)
    {
        if(ch == '\n')
        {
            totalLines++;
        }
    }

    FILE *fr;
    fr = fopen("usernames.txt","r");

    for(int i=0 ; i<totalLines ; i++)
    {
        fscanf(fr,"%s %s %d %s",u[i].userName,u[i].userPassword,&u[i].userAge,u[i].gender);
    }

    printf("\n\n\t-----------------------------------------------------\n");
    printf("\tS.No.\tUser Name\tPassword\tAge\tgender\n");
    printf("\t-----------------------------------------------------\n\n");

    for(int i=0 ; i<totalLines ; i++)
    {
        printf( "\t  %-5d\t%-12s\t%-8s\t%-3d\t%-10s\n",i+1,u[i].userName,u[i].userPassword,u[i].userAge,u[i].gender);
    }

    printf("\n\tSearch for particular user's details?(y/n)");
    char choice = getch();
    char searchUser[30];
    int count = 1;

    if(choice == 'y')
    {
        printf("\n\tEnter a valid user name to search:");
        scanf("%s",searchUser);
        for(int i=0 ; i<totalLines ; i++)
        {
            if(strcmp(searchUser,u[i].userName) == 0)
```

```
        {
            if(strcmp(searchUser,u[i].userName) == 0)
            {
                printf("\n\n\tS.No.\tUser Name\tPassword\tAge\tgender\n\n");
                printf( "\t  %-5d\t %-12s\t %-8s\t %-3d\t %-10s\n",i+1,u[i].userName,u[i].userPassword,u[i].userAge,u[i].gender);
                Sleep(2000);
                printf("\n\n\tPress Enter key to go back to main menu");
                char enter = getch() ;
                adminMainMenu();
            }
            else if(count == totalLines)
            {
                printf("Invalid username");
                Sleep(2000);
                adminMainMenu();
            }
            else
            {
                count++;
            }
        }
    }
    else
    {
        adminMainMenu();
    }
    fclose(fr);
    fclose(fview);
}

void games()
{
    char anonymousUser[30];
    system("cls");

    system("cls");
    int position = 1;
    int keyPressed = 0;
    while(keyPressed != 13)
    {
        system("cls");

        gotoxy(30,4);
        printf("\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\x
        gotoxy(30,5);
        printf("\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\x
        gotoxy(30,6);
```

```
887         gotoxy(30,6);
888         printf("                                                  ");
889         gotoxy(30,7);
890         printf("                 SELECT MODE                      ");
891         gotoxy(30,8);
892         printf("                                                  ");
893         gotoxy(30,9);
894         printf("\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\x
895         gotoxy(30,10);
896         printf("\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\x
897
898         gotoxy(30,13);
899         printf("-----------------------------------------------------");
900         gotoxy(30,15);
901         printf("\t\t"); arrow(1,position); printf("SINGLE PLAYER");
902         gotoxy(30,16);
903         printf("\t\t");  arrow(2,position); printf("MULTI PLAYER (COMPETE)");
904         gotoxy(30,17);
905         printf("\t\t"); arrow(3,position); printf("EXIT");
906
907         gotoxy(30,18);
908         printf("-----------------------------------------------------");
909
910         gotoxy(30,22);
911         printf("\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\x
912         gotoxy(30,23);
913         printf("\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\x
914         gotoxy(30,24);
915         printf("\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\x
916         gotoxy(30,25);
917         printf("\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\x
918         keyPressed = getch();
919
920         if(keyPressed == 80 && position != 3)
921             position++;
922         else if(keyPressed == 72 && position != 1)
923             position--;
924         else
925             position = position;
926     }
927     switch(position)
928     {
929         case 1:printf("\n\n\n\t\t\t\tEnter player name: ");
930                 scanf("%s",anonymousUser);
931                 gameMenu(anonymousUser);
932         case 2:compete();
```

```
        case 2:compete();
                break;
        case 3:homePage();
                break;
        default:printf("Invalid entry!!!");
                homePage();
                break;
    }
}

void compete()
{
    int numOfPlayers,gameScore;
    char numPlayers[2], typing_duration[3];
    float speed,accuracy,comprehension;
    float max_reading_speed = 6000.0;
    struct player p[200];
    char score;
    system("cls");

    system("cls");
    int position = 1;
    int keyPressed = 0;
    while(keyPressed != 13)
    {
        system("cls");

        gotoxy(30,4);
        printf("\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\x
        gotoxy(30,5);
        printf("\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\x
        gotoxy(30,6);
        printf("                                                    ");
        gotoxy(30,7);
        printf("                    COMPETE                    ");
        gotoxy(30,8);
        printf("                                                    ");
        gotoxy(30,9);
        printf("\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\x
        gotoxy(30,10);
        printf("\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\x

        gotoxy(30,13);
        printf("-----------------------------------------------------");
        gotoxy(30,15);
        printf("\t\t"); arrow(1,position); printf("TYPING CHALLENGE");
```

```
        gotoxy(30,16);
        printf("\t\t");  arrow(2,position); printf("READING CHALLENGE");
        gotoxy(30,17);
        printf("\t\t");  arrow(3,position); printf("DISAPPEARING WORDS (GAME)");
        gotoxy(30,18);
        printf("\t\t");  arrow(4,position); printf("YOU GOT EM ALL?");
        gotoxy(30,19);
        printf("\t\t"); arrow(5,position); printf("EXIT");
        gotoxy(30,21);
        printf("-------------------------------------------------------");

        gotoxy(30,24);
        printf("\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\x
        gotoxy(30,25);
        printf("\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\x
        gotoxy(30,26);
        printf("\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\x
        gotoxy(30,27);
        printf("\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\x
        keyPressed = getch();

        if(keyPressed == 80 && position != 5)
            position++;
        else if(keyPressed == 72 && position != 1)
            position--;
        else
            position = position;
    }
    if(position == 5)
    {
        games();
    }
    label:
    system("cls");
    gotoxy(35,10);
    printf("Enter the number of players: ");
    gotoxy(64,10);gets(numPlayers);
    if(atoi(numPlayers) && (atoi(numPlayers) >= 1 && atoi(numPlayers) <=10))       //checking if only permissible number of users number is entered or not
    {
        Sleep(2000);
        system("cls");
    }
    else
    {
        gotoxy(35,12);printf("Enter a number between 1 and 10");
        Sleep(3000);
```

```
        gotoxy(35,12);printf("                                        ");
        gotoxy(64,10);printf("      ");
        goto label;
    }

    for(int i=0 ; i<atoi(numPlayers) ; i++)        //systems prompts competitors to enter their names
    {
        players:
        gotoxy(35,14);printf("Enter player %d name: ",i+1);
        gotoxy(35,18);printf("                                                  ");
        gotoxy(35,16);printf("                          ");
        gotoxy(35,16);scanf("%s",p[i].name);
        if(i>0)
        {
            for(int j = i-1 ; j >=0 ; j--)
            {
                if(strcmp(p[i].name,p[j].name) == 0)
                {
                    gotoxy(35,18);printf("Name is already taken, try something else");
                    gotoxy(35,16);printf("                          ");
                    Sleep(2000);
                    goto players;
                }
            }
        }
    }

    if(position == 1)
    {
        system("cls");

        gotoxy(35,10);
        printf("\xB2\xB2\xB2
```

```
int typingDuration;
if(atoi(typing_duration) && (atoi(typing_duration) < 120 && atoi(typing_duration) >= 1))
{
    typingDuration = atoi(typing_duration);
}
else
{
    gotoxy(35,16);printf("Enter a value from 1-120");
    Sleep(1000);
    gotoxy(71,12);printf("    ");
    goto duration;
}

for(int i=0 ; i<atoi(numPlayers) ; i++)        //system prompts competitor to participate in turns
{
    system("cls");
    printf("\n\n\n\n\n\n\n\n\t\t\t\t\t\t\t%s's turn\n\n",p[i].name);
    typingCompetition(typingDuration,&speed,&accuracy);
    p[i].typing_speed = speed;
    p[i].typing_accuracy = accuracy;
}

for(int i=0 ; i < atoi(numPlayers) ; i++)
{
    for(int j=0 ; j< atoi(numPlayers) ; j++)
    {
        struct player temp;
        if(p[i].typing_speed < p[i].typing_speed)
        {
            temp=p[i];
            p[i]=p[j];
            p[j]=temp;
        }
    }
}

system("cls");
printf("\n\n\t\tResults: ");
printf("\n\t-----------------------------------------------------------------------------------");
printf("\n\tS.no\tName\t\tTyping Speed\tTyping Accuracy");
printf("\n\t-----------------------------------------------------------------------------------\n\n");
for(int i=0 ; i<atoi(numPlayers) ; i++)
{
    printf("\t%4d\t%4s\t\t%12f\t%15f\n",i+1,p[i].name,p[i].typing_speed,p[i].typing_accuracy);
}
printf("\n\n\t\t\t%s is the winner. Congratulations!!!\n\n",p[0].name);
```

```
        printf("\n\n\t\t\t%s is the winner, Congratulations!!!\n\n",p[0].name);
    }

    else if(position == 2)
    {
        for(int i=0 ; i<atoi(numPlayers) ; i++)
        {
            system("cls");
            printf("\n\n\n\n\n\n\n\n\t\t\t\t\t\t\t%s's turn\n\n",p[i].name);
            Sleep(3000);
            readingCompetition(&speed,&comprehension);
            p[i].reading_speed = speed;
            p[i].reading_comprehension = comprehension;
        }
        system("cls");


        for(int i=0;i<atoi(numPlayers);i++)
        {
            for(int j=i;j<atoi(numPlayers);j++)
            {
                struct player temp;
                int x,y;
                x = (p[i].reading_speed)*100/(max_reading_speed) + p[i].reading_comprehension;
                y = (p[j].reading_speed)*100/(max_reading_speed) + p[j].reading_comprehension;
                if(x < y)
                {
                    temp=p[i];
                    p[i]=p[j];
                    p[j]=temp;
                }
            }
        }

        system("cls");
        printf("\n\n\t\tResults: ");
        printf("\n\t-----------------------------------------------------------------------------------");
        printf("\n\tS.no Name\t\tReading Speed Reading Comprehension");
        printf("\n\t-----------------------------------------------------------------------------------\n\n");
        for(int i=0 ; i<atoi(numPlayers) ; i++)
        {
            printf("\t%-4d% -4s %-13.1f %-21.1f\n",i+1,p[i].name, p[i].reading_speed,p[i].reading_comprehension);
        }

        printf("\n\n\t\t\t%s is the winner, Congratulations!!!\n\n",p[0].name);
    }
```

```
else if(position == 3)
{
    int level = levelSelector();
    Sleep(1000);
    if(level == 3)
    {
        compete();
    }
    FILE *gl;

    for(int i=0 ; i<atoi(numPlayers) ; i++)
    {
        system("cls");
        printf("\n\n\n\n\n\n\n\t\t\t\t\t\t%s's turn\n\n",p[i].name);
        Sleep(3000);
        game1(p[i].name,level,"no");
    }
    if(level == 1)
    {
        gl = fopen("game1level1.txt","r");
    }
    else if(level == 2)
    {
        gl = fopen("game1level2.txt","r");
    }

    char c,name[30];
    int i=0;
    while((c = getc(gl)) != EOF)
    {
        fscanf(gl,"%s %d",name,&p[i].gameScore);
        i++;
    }
    fclose(gl);
    system("cls");

    for(int i=0;i<atoi(numPlayers);i++)
    {
        for(int j=i;j<atoi(numPlayers);j++)
        {
            struct player temp;

            if(p[i].gameScore < p[i].gameScore)
            {
                temp=p[i];
                p[i]=p[j];
```

```
                p[i]=p[j];
                p[j]=temp;
            }
        }
    }
    printf("\n\n\t\tResults: ");
    printf("\n\t\t-----------------------------------------------");
    printf("\n\t\tS.no Name\t\tScore");
    printf("\n\t\t-----------------------------------------------\n\n");
    for(int i=0 ; i<atoi(numPlayers) ; i++)
    {
        printf("\t\t%-4d% -4s\t\t%-5d\n",i+1,p[i].name, p[i].gameScore);
    }

    printf("\n\n\t\t\t%s is the winner, Congratulations!!!\n\n",p[0].name);

    clear_Leaderboard("game1level1.txt");
    clear_Leaderboard("game1level2.txt");
}

else if(position == 4)
{
    int level = levelSelector();
    printf("%d",level);
    Sleep(1000);
    if(level == 3)
    {
        compete();
    }
    FILE *gl;

    for(int i=0 ; i<atoi(numPlayers) ; i++)
    {
        system("cls");
        printf("\n\n\n\n\n\n\n\t\t\t\t\t\t\t%s's turn\n\n",p[i].name);
        Sleep(3000);
        game2(p[i].name,level,"no");
    }

    if(level == 1)
    {
        gl = fopen("game1level1.txt","r");
    }
    else if(level == 2)
    {
        gl = fopen("game1level2.txt","r");
```

```
        }

        char c,name[30];
        int i=0;
        while((c = getc(gl)) != EOF)
        {
            fscanf(gl,"%s %d",name,&p[i].gameScore);
            i++;
        }
        fclose(gl);
        system("cls");

        for(int i=0;i<atoi(numPlayers);i++)
        {
            for(int j=i;j<atoi(numPlayers);j++)
            {
                struct player temp;

                if(p[i].gameScore < p[j].gameScore)
                {
                    temp=p[i];
                    p[i]=p[j];
                    p[j]=temp;
                }
            }
        }
        printf("\n\n\t\tResults: ");
        printf("\n\t\t-----------------------------------------------");
        printf("\n\t\tS.no Name\t\tScore");
        printf("\n\t\t-----------------------------------------------\n\n");
        for(int i=0 ; i<atoi(numPlayers) ; i++)
        {
            printf("\t\t%-4d% -4s\t\t%-5d\n",i+1,p[i].name, p[i].gameScore);
        }
        clear_Leaderboard("game1level1.txt");
        clear_Leaderboard("game1level2.txt");
    }

    printf("Press any key to go back to Menu:");
    getch();
    compete();
}

void typingCompetition(int typing_duration,float *speed , float *accuracy) ...

void readingCompetition(float *speed , float *comprehension) ...
```

```c
void typingCompetition(int typing_duration,float *speed , float *accuracy)
{
    char ch,c;
    char a[2000];
    int k=0,count=0,i=0,b,wrong=0, num_letters_typed=0;
    float typing_accuracy;
    int randomisedArray[9];            // change size according to number of paragraphs
    time_t start,end;
    FILE *ft;

    printf("\n\t\t\t----------Passage----------\n\n\n");

    ft = fopen("paragraphsTyping.txt","r");

    ranArray(randomisedArray,9);

    int randomNumber = randomisedArray[0];        //takes a random number and prints that numbered paragraph from text file

    int parNumber =0;

    while((ch = getc(ft)) != EOF)
    {
        if(ch == '~')       //paragraphs are separated by '~'
        {
            parNumber++;            //paragraph number in text file
            if(parNumber == randomNumber)       //when paragraph number = rand number, print that paragraph
            {
                int i = 0;
                printf("\t");
                while((ch=getc(ft)) != '~'  && ch != EOF)
                {
                    a[i] = ch;          //copy that particular paragraph for furthur usage
                    if(ch == '\n')
                    {
                        printf("\n\t");
                    }
                    else
                    {
                        putchar(ch);
                    }
                    i++;
                }
                a[i] = '\0';
            }
        }
    }
```

```
fclose(ft);

printf("\n\n");

printf("\n\n\t\t---->Timer automatically starts when you start typing <---\n\n\n");

// p=fopen("Typing_file1.txt","r");
int m = 0;
while(a[m] != '\0')        //for assessing user input(whether is matches the paragraph displayed or not)
{
    label:
    b=0;
    c=getch();
    if(k==0)        //if entering the loop for the first time
    {
        start=time(NULL);
    }
    if(c == a[m])        //if character matches
    {
        //Beep(600,100);
        putchar(c);
        if(c==' ')
        {
            count++;
        }
        num_letters_typed++;
    }
    else if(c=='\r'&&a[m]=='\n')        //if 'enter' is pressed
    {
        //Beep(600,100);
        putchar(c);
        printf("\n");
        num_letters_typed++;
    }
    else        //if character mismatch
    {
        b=1;
        wrong++;
        //Beep(600,100);
    }
    end=time(NULL);
    if(difftime(end,start)>=typing_duration)
    {
        printf("\a");
        printf("\n\n\t\t!!!!Time over!!!!\n\n");
        break;
```

```
        }
        if(b==1)
        {
            goto label;
        }
        k++;
        m++;
    }

    float typing_speed;
    char lastScore[5];
    typing_speed=((float)num_letters_typed/5)/((float)typing_duration/60);  //calculating typing speed
    printf("\n\t\tYour typing speed");
    printf("\n\n\t\t***** %.2f WPM ******\n\n",typing_speed);
    *speed = typing_speed;

    typing_accuracy=(float)num_letters_typed/(num_letters_typed+wrong)*100;     //calculating typing accuracy

    printf("\n\t\tYour accuracy\n");
    printf("\n\t\t----%.1f percentage----", typing_accuracy);
    *accuracy = typing_accuracy;


    printf("\n\n\t\tPress Enter to continue...\n\n");
    while((getch())!='\r');
}

void readingCompetition(float *speed , float *comprehension)
{
    system("cls");
    int length_of_text = 1,k=0;
    char ch,a[2000];
    double reading_speed, reading_comprehension, time_elapsed;
    int randomisedArray[5];            // change size according to number of paragraphs
    time_t start,end;
    FILE *ft,*fp;

    system("cls");
    gotoxy(50,20);printf("\nPassage will appear in 3 seconds");
    Sleep(3000);

    system("cls");

    ft = fopen("pragraphsTyping.txt","r");

    printf("\n\n\n\t\t\t---------------Passage----------------\n\n\n");
```

```
ranArray(randomisedArray,5);

int randomNumber = randomisedArray[0];     //takes a random number and prints that numbered paragraph from text file
// printf("%d",randomNumber);

int parNumber =0;

while((ch = getc(ft)) != EOF)
{
    if(ch == '~')      //paragraphs are separated by '~'
    {
        parNumber++;
        if(parNumber == randomNumber)
        {
            int i = 0;
            printf("\t");
            while((ch=getc(ft)) != '~'  && ch != EOF)
            {
                a[i] = ch;     //copy into a char string for surthur purpose
                if(ch == '\n')
                {
                    printf("\n\t");
                }
                else
                {
                    putchar(ch);
                }
                i++;
            }
            a[i] = '\0';
        }
    }
}
fclose(ft);

printf("\n\n");

start=time(NULL);      //storing starting time
printf("\n\nPress Enter if you have completed reading....\n");
while((getch())!='\r');

system("cls");
printf("\n\n");

end = time(NULL);      //end time when user completes reading
time_elapsed = difftime(end,start);
```

```
    printf("Time elapsed : %.0f sec", time_elapsed);

    int i = 0;
    while(a[i] != '\0')        //using a[] char string for finding length of the paragraph
    {
        if(a[i]=='\n' || a[i]==' ')
        {
            length_of_text++;
        }
        i++;
    }

    reading_speed = (length_of_text/time_elapsed)*60;        //calculating reading speed
    reading_comprehension = Readingcomprehension(randomNumber);  //calculating reading comprehension

    printf("\n\n\t\t----Your Reading speed is %.2f WPM----\n\n", reading_speed);
    *speed = reading_speed;

    printf("\n\n\t\t----Your Reading comprehension is %.2f percentage----\n\n", reading_comprehension);
    *comprehension = reading_comprehension;

    printf("\n\n\t\tPress Enter to continue...\n\n");
    while((getch())!='\r');
}

void Typingspeed(char UserName[])       //Calculates Typing speed
{
    char ch,c;
    char a[2000];       //for storing a random passage into a string
    int k=0,count=0,i=0,b,wrong=0, num_letters_typed=0;
    char typing_duration[3];
    float typing_accuracy;
    time_t start,end;       //keeping the track of starting and ending time
    int randomisedArray[9];
    FILE *ft;

    system("cls");

    gotoxy(35,10);
    printf("\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\x
    gotoxy(35,11);
    printf("\xB2");gotoxy(75,11);printf("\xB2");
    gotoxy(35,12);
    printf("\xB2");gotoxy(37,12);printf("Enter your typing duration(sec): ");gotoxy(75,12);printf("\xB2");
    gotoxy(35,13);
    printf("\xB2");gotoxy(75,13);printf("\xB2");
```

```
gotoxy(35,14);
printf("\xB2\xB2\xB2
```

```
            int i = 0;
            printf("\t");
            while((ch=getc(ft)) != '~'  && ch != EOF)
            {
                a[i] = ch;
                if(ch == '\n')
                {
                    printf("\n\t");
                }
                else
                {
                    putchar(ch);
                }
                i++;
            }
            a[i] = '\0';
        }
    }
}
fclose(ft);

printf("\n\n");

printf("\n\n\t\t---->Timer automatically starts when you start typing <---\n\n\n");
// p=fopen("Typing_file1.txt","r");
int m = 0;

while(a[m] != '\0')
{
    label:
    b=0;
    c=getch();
    if(k==0)        //if entering the loop for the first time
    {
        start=time(NULL);
    }
    if(c == a[m])       //if character matches
    {
        //Beep(600,100);
        putchar(c);
        if(c==' ')
        {
            count++;
        }
        num_letters_typed++;
    }
```

```
        else if(c=='\r'&& a[m]=='\n')      //if 'enter' is pressed
        {
            //Beep(600,100);
            putchar(c);
            printf("\n");
            num_letters_typed++;
        }
        else       //if character mismatch
        {
            b=1;
            wrong++;
            //Beep(600,100);
        }
        end=time(NULL);

        if(difftime(end,start) >= typingDuration)      //if time has finished
        {
            printf("\a");
            printf("\n\n\t\t!!!!Time over!!!!\n\n");
            break;
        }
        if(b==1)
        {
            goto label;
        }
        k++;
        m++;
    }

    float typing_speed;

     typing_speed=((float)num_letters_typed/5)/((float)typingDuration/60);      //calculating typing speed
    printf("\n\t\tYour typing speed");
    printf("\n\n\t\t***** %.2f WPM ******\n\n",typing_speed);

    typing_accuracy=(float)num_letters_typed/(num_letters_typed+wrong)*100;     //calculating typing accuracy
    printf("\n\t\tYour accuracy\n");
    printf("\n\t\t----%.1f percentage----", typing_accuracy);
    printf("\n\n\t\tPress Enter to continue...\n\n");


    while((getch())!='\r');     //enter to continue

    Add_To_Typing_Leaderboard(typing_speed, typing_accuracy,UserName);      //adding to typing leaderboard

}
```

```c
void Add_To_Typing_Leaderboard(float typing_speed, float typing_accuracy, char UserName[])     //adds user to typing leaderboard
{
    struct player p[200],temp;
    FILE *fp;
    char c;
    int l=0,i,j;

    fflush(stdin);

    if(is_Registered_user(UserName, "Typing_Leaderboard.txt"))     //checking if user is registered
    {
        printf("-------------Registered-------------");
        display_Feedback(UserName, typing_speed, typing_accuracy, "Typing_Leaderboard.txt");       //displays feedback
        Delete_User("Typing_Leaderboard.txt", UserName);       //removes the previous record of the user
    }

    fp=fopen("Typing_Leaderboard.txt","a");
   // printf("******%s********",UserName);
    fprintf(fp,"%s %.1f %.1f\n",UserName, typing_speed, typing_accuracy);      //appending results to file
    fclose(fp);

    Display_Typing_Leaderboard(UserName);      //displays the leaderboard

}

void Display_Typing_Leaderboard(char UserName[])
{
    struct player p[200],temp;
    FILE *fp;
    char c;
    int l=0,i,j;
    system("cls");
    printf("\n\t\tleaderboard :\n\n");
    printf("\n\t-----------------------------------------------------------------------------------------");
    printf("\n\tS.No   Name\t\t\t\tSpeed\tAccuracy");
    printf("\n\t-----------------------------------------------------------------------------------------\n\n");

    fp=fopen("Typing_Leaderboard.txt","r");
    while((c=getc(fp))!=EOF)       //calculating the number of lines in the leaderboard file
    {
        if(c=='\n')
        {
            l++;
        }
    }
    fclose(fp);
```

```
    fp=fopen("Typing_Leaderboard.txt","r");
    for(i=0;i<l;i++)        //storing the leaderboard data into a structure array
    {
        fscanf(fp,"%s %f %f",p[i].name,&p[i].typing_speed,&p[i].typing_accuracy);
    }
    fclose(fp);

    for(i=0;i<l;i++)        //sorting the data in leaderboard
    {
        for(j=i;j<l;j++)
        {
            if(p[i].typing_speed<p[j].typing_speed)
            {
                temp=p[i];
                p[i]=p[j];
                p[j]=temp;
            }
        }
    }

    for(i=0;i<l;i++)        //displaying the leaderboard
    {
        printf("\t %-4d %-4s\t\t\t\t%-5.1f\t%-8.1f\n",i+1,p[i].name,p[i].typing_speed,p[i].typing_accuracy);
    }

    Sleep(2000);
    printf("\nPress Enter key to go back to main menu");
    char enter = getch() ;
    if(strcmp(UserName,"admin") == 0)
        adminMainMenu();
    else
        mainMenu(UserName);
}

void Readingspeed(char UserName[])
{
    system("cls");
    int length_of_text = 1,k=0;
    char ch,a[2000];
    int randomisedArray[5];
    double reading_speed, reading_comprehension, time_elapsed;
    time_t start,end;
    FILE *ft;

    system("cls");
    gotoxy(35,20);printf("Passage will appear in 3 seconds");
```

```
Sleep(3000);

system("cls");
ft = fopen("paragraphsReading.txt","r");

printf("\n\n\n\t\t\t----------Passage----------\n\n\n");

ranArray(randomisedArray,5);

int randomNumber = randomisedArray[0];
// printf("%d",randomNumber);

int parNumber =0;

while((ch = getc(ft)) != EOF)      //choosing a random paragraph
{
    if(ch == '~')      //paragraphs are separated by '~'
    {
        parNumber++;
        if(parNumber == randomNumber)
        {
            int i = 0;
            printf("\t");
            while((ch=getc(ft)) != '~'  && ch != EOF)
            {
                a[i] = ch;
                if(ch == '\n')
                {
                    printf("\n\t");
                }
                else
                {
                    putchar(ch);
                }
                i++;
            }
            a[i] = '\0';
        }
    }
}
fclose(ft);

printf("\n\n");

start=time(NULL);      //storing starting time
printf("\n\n\t\t\tPress Enter if you have completed reading....\n");
```

```
    while((getch())!='\r');

    system("cls");
    printf("\n\n");

    end = time(NULL);      //end time when user completes reading
    time_elapsed = difftime(end,start);
    printf("Time elapsed : %.0f sec", time_elapsed);

    int i = 0;
    while(a[i] != '\0')     //using a[] char string for finding length of the paragraph
    {
        if(a[i]=='\n' || a[i]==' ')
        {
            length_of_text++;
        }
        i++;
    }

    reading_speed = (length_of_text/time_elapsed)*60;      //calculating reading speed
    reading_comprehension = Readingcomprehension(randomNumber);     //calculating comprehension
    printf("\n\n\t\t----Your Reading speed is %.2f WPM----\n\n", reading_speed);
    printf("\n\n\t\t----Your Reading comprehension is %.2f percentage----\n\n", reading_comprehension);

    printf("\n\n\t\tPress Enter to continue...\n\n");
    while((getch())!='\r');

    Add_To_Reading_Leaderboard(reading_speed, reading_comprehension,UserName);     //adding user to reading leaderboard

}

double Readingcomprehension(int randomNumber)
{
    system("cls");
    printf("\n\n\t\t------------Reading Comprehension------------\n\n");
    int num_questions = 5, points = 0;
    int i;
    char question[100], answer[20], response[20];
    double reading_comprehension;
    char ch;
    FILE *fp;
    fp = fopen("Quiz.txt","r");

    for(int count =0; count < randomNumber; count++)       //to select random quiz questions
    {
        while((ch = getc(fp)) != '~' && ch != EOF);     //group of questions are separated by '~'
```

```
    }

    for(i=0; i<num_questions; i++)      //displaying questions and getting responses
    {
        fscanf(fp, "%[^?]s ",question);
        fseek(fp, 1, 1);
        fscanf(fp, "%s ",answer);
        printf("%s ---- %s\n",question, answer);
        fflush(stdin);
        printf("%s?\nYour response : ", question);
        gets(response);
        printf("\n");

        if(strcmp(response, answer) == 0)          //if correct answer
        {
            points++;
        }
    }

    reading_comprehension = ((double)points/num_questions) * 100;
    return reading_comprehension;
    fclose(fp);

}

void Display_Reading_Leaderboard(char UserName[])
{
    struct player p[200], temp;
    FILE *fp;
    char c;
    float max_reading_speed = 6000.0;       //This is an approximate maximum reading speed
    int l=0,i,j;
    system("cls");
    printf("\n\t\tReading_Leaderboard :\n\n");
    printf("\n\t-----------------------------------------------------------------------------------------");
    printf("\n\tS.no  Name \t\tReading_speed(WPM)\t\tReading_comprehension(%%)");
    printf("\n\t-----------------------------------------------------------------------------------------\n\n");

    fp=fopen("Reading_Leaderboard.txt","r");
    while((c=getc(fp))!=EOF)        //calculating number of lines
    {
        if(c=='\n')
        {
            l++;
        }
    }
```

```
    fclose(fp);

    fp=fopen("Reading_Leaderboard.txt","r");
    for(i=0;i<l;i++)        //retrieving data from file to a structure array
    {
        fscanf(fp,"%s %f %f",p[i].name,&p[i].reading_speed,&p[i].reading_comprehension);
    }
    fclose(fp);

    for(i=0;i<l;i++)        //sorting leaderboard taking both speed and comprehension into consideration
    {
        for(j=i;j<l;j++)
        {
            int x,y;
            x = (p[i].reading_speed)*100/(max_reading_speed) + p[i].reading_comprehension;
            y = (p[j].reading_speed)*100/(max_reading_speed) + p[j].reading_comprehension;
            if(x < y)
            {
                temp=p[i];
                p[i]=p[j];
                p[j]=temp;
            }
        }
    }

    for(i=0;i<l;i++)        //displaying leaderboard
    {
        printf("\t %-6d%-16s%-18.1f\t\t%-21.1f\n",i+1,p[i].name,p[i].reading_speed,p[i].reading_comprehension);
    }

    Sleep(2000);
    printf("\nPress Enter key to go back to main menu");
    char enter = getch() ;
    if(strcmp(UserName,"admin") == 0)
        adminMainMenu();
    else
        mainMenu(UserName);
}

void Add_To_Reading_Leaderboard(double reading_speed, double reading_comprehension, char UserName[])
{
    struct player p[200],temp;
    FILE *fp;
    char c, name[50], filename[50];
    int l=0,i,j;
```

```
1941    fflush(stdin);
1942
1943    if(is_Registered_user(UserName, "Reading_Leaderboard.txt"))     //checking if user is registered
1944    {
1945        //printf("-------------Registered----------");
1946        display_Feedback(UserName, reading_speed, reading_comprehension, "Reading_Leaderboard.txt");       //displaying feedback
1947        Delete_User("Reading_Leaderboard.txt", UserName);
1948    }
1949
1950    fp=fopen("Reading_Leaderboard.txt","a");
1951    fprintf(fp,"%s %.1f %.1f\n",UserName, reading_speed, reading_comprehension);
1952    fclose(fp);
1953
1954    Display_Reading_Leaderboard(UserName);      //displays the leaderboard
1955 }
1956
1957 void viewLeaderBoards(char UserName[])
1958 {
1959    system("cls");
1960    int choice,position = 1,keyPressed = 0;
1961    while(keyPressed != 13)
1962    {
1963
1964        system("cls");
1965        gotoxy(20,4);
1966        printf("\xDB\xDB\xDB
```

```
        else
            position = position;
    }
    switch (position)
    {
            case 1: Display_Typing_Leaderboard(UserName);
                    break;
            case 2: Display_Reading_Leaderboard(UserName);
                    break;
    }
}

void Delete_User(char  filename[],char userName[])      //deletes a user from all the leaderboars and from the registration file
{
    int temp = 1;
    int num_lines = 0, delete_line, counter = 0, user_found = 0;
    char username[50], c, ch;
    struct player p[200];
    FILE *fp;
    FILE *fileptr1, *fileptr2;

    fp=fopen(filename,"r");
    while((c=getc(fp))!=EOF)        //calculating number of lines
    {
        if(c=='\n')
        {
            num_lines++;
        }
    }
    fclose(fp);

    fp=fopen(filename,"r");
    for(int i=0;i<num_lines;i++)        //searching for the user to be deleted
    {
        counter++;
        fscanf(fp,"%s %f %f",p[i].name,&p[i].typing_speed,&p[i].typing_accuracy);

        if(strcmp(p[i].name, userName) == 0)
        {
            delete_line = counter;
            user_found = 1;
            break;
        }
    }
    // printf("%d",counter);
    fclose(fp);
```

```
    if(user_found != 1)     //if user not found
    {
        printf("\n\n\t\t\t\t\t\t---------User Not Found----------\n\n");
        return;
    }
    fileptr1 = fopen(filename, "r");
    fileptr2 = fopen("replica.txt", "w");

    while ((ch = getc(fileptr1)) != EOF)
    {
            if (temp != delete_line)
            {
                //copy all lines in file replica.c
                putc(ch, fileptr2);
            }
            if (ch == '\n') temp++;
    }
    fclose(fileptr1);
    fclose(fileptr2);
    remove(filename);
    //rename the file replica.c to original name
    rename("replica.txt", filename);
    printf("\n\n\t\t\t\t\t\t------------User : successfully removed---------------\n\n");
}

void Delete_User_credentials(char  filename[],char userName[])
{
    int temp = 1;
    int num_lines = 0, delete_line, counter = 0, user_found = 0;
    char username[50], c, ch;
    struct user u[200];
    FILE *fp;
    FILE *fileptr1, *fileptr2;

    fp=fopen(filename,"r");
    while((c=getc(fp))!=EOF)
    {
        if(c=='\n')
        {
            num_lines++;
        }
    }
    fclose(fp);

    fp=fopen(filename,"r");
    for(int i=0;i<num_lines;i++)
```

```
    if(user_found != 1)
    {
        printf("\n\n\t\t\t\t\t\t---------User Not Found in %s---------- \n\n",filename);
        return;
    }
    fileptr1 = fopen(filename, "r");
    fileptr2 = fopen("replica.txt", "w");

    while ((ch = getc(fileptr1)) != EOF)
    {
            if (temp != delete_line)
            {
                //copy all lines in file replica.c
                putc(ch, fileptr2);
            }
            if (ch == '\n') temp++;
    }

    fclose(fileptr1);
    fclose(fileptr2);
    remove("usernames.txt");
    // printf("%s",filename);
    //rename the file replica.c to original name
    rename("replica.txt","usernames.txt");
    printf("\n\n\t\t\t\t\t\t\t\t------------User successfully removed from %s---------------\n\n",filename);
}

void clear_Leaderboard(char filename[])     //clears the leaderboard
{
    fclose(fopen(filename, "w"));
    // printf("\n\n\t\t--------------Leaderboard cleared----------------\n\n");
}

void gameMenu(char userName[])
{
    system("cls");
    int position = 1;
    int keyPressed = 0;

    while(keyPressed != 13)
    {
            system("cls");

            gotoxy(30,4);
            printf("\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xD
```

```
}

void gameMenu(char userName[])
{
    system("cls");
    int position = 1;
    int keyPressed = 0;

    while(keyPressed != 13)
    {
        system("cls");

        gotoxy(30,4);
        printf("\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xD
        gotoxy(30,5);
        printf("\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xD
        gotoxy(30,6);
        printf("                                                    ");
        gotoxy(30,7);
        printf("                    GAME                    ");
        gotoxy(30,8);
        printf("                                                    ");
        gotoxy(30,9);
        printf("\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xD
        gotoxy(30,10);
        printf("\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xD

        gotoxy(30,13);
        printf("-------------------------------------------------------");

        gotoxy(30,15);
        printf("\t\t"); arrow(1,position); printf("DISAPPEARING WORDS");
        gotoxy(30,16);
        printf("\t\t");  arrow(2,position); printf("YOU GOT EM ALL?");
        gotoxy(30,17);
        printf("\t\t"); arrow(3,position); printf("EXIT");

        gotoxy(30,19);
        printf("-------------------------------------------------------");

        gotoxy(30,22);
        printf("\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xD
        gotoxy(30,23);
        printf("\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xD
        gotoxy(30,24);
        printf("\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xDB\xD
```

```
        gotoxy(30,25);
        printf("\xDB\xDB\xDB
```

```
2215    struct user u[100];
2216    char enteredWord[30];
2217    int position = 1;
2218    int a[96];
2219    int points = 0;
2220    int keyPressed = 0;
2221    time_t start,end;
2222    FILE *gf,*gl1,*gl2;
2223    gf = fopen("gameWords.txt","r");
2224
2225    if(level == 3)
2226    {
2227        gameMenu(userName);
2228    }
2229    system("cls");
2230    printf("\n\n\n\n\n\t\t\t\t\tWords will be displayed for a limited period of time\n\t\t\t\tcorrectly type in as many words as possible\n\t\t\t\tClock starts ticking as soon as ga
2231    Sleep(5000);
2232    if(level == 1)
2233    {
2234        system("cls");
2235        char ch;
2236        int i=0;
2237        while((ch = getc(gf)) != EOF)
2238        {
2239            fscanf(gf,"%s",u[i].gameWord);
2240            i++;
2241        }
2242
2243        start = time(NULL);
2244        ranArray(a,96);
2245        int j=0;
2246        end = time(NULL);
2247        gotoxy(30,18);
2248        printf("\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2
2249
2250        gotoxy(30,27);
2251        printf("\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2
2252
2253        while(difftime(end,start) < 30)
2254        {
2255            gotoxy(45,20);printf("%s",u[a[j]].gameWord);
2256            Sleep(500);
2257            gotoxy(45,20);printf("                    ");
2258            gotoxy(32,25);printf("Enter the Word:                ");
2259            gotoxy(52,25);scanf("%s",enteredWord);
2260            if(strcmp(enteredWord,u[a[j]].gameWord) == 0)       //if user enters correct word, he/she gets 2 points
```

```
        if(strcmp(enteredWord,u[a[j]].gameWord) == 0)    //if user enters correct word, he/she gets 2 points
        {
            points += 2;
        }
        end = time(NULL);
        j++;
    }

    if(strcmp(singleOrNot,"no") != 0)
    {
        gl1 = fopen("game1level1.txt","a");
        fprintf(gl1,"%s %d\n",userName,points);
        fclose(gl1);
    }
    gotoxy(45,29);printf("Time up!!!");
    gotoxy(45,30);printf("Your score: %d",points);
    Sleep(3000);
}
else if(level == 2)
{
    system("cls");
    char ch;
    int i=0;
    while((ch = getc(gf)) != EOF)
    {
        fscanf(gf,"%s",u[i].gameWord);
        i++;
    }

    start = time(NULL);
    ranArray(a,96);
    int j=0;
    end = time(NULL);
    gotoxy(30,18);
    printf("\xB2\xB2\xB2
```

```
            gotoxy(52,25);scanf("%s",enteredWord);
            if(strcmp(enteredWord,u[a[j]].gameWord) == 0)
            {
                points += 2;
            }
            end = time(NULL);
            j++;
        }


        if(strcmp(singleOrNot,"no") != 0)
        {
            gl2 = fopen("game1level2.txt","a");
            fprintf(gl2,"%s %d",userName,points);
            fclose(gl2);
        }
        gotoxy(45,29);printf("Time up!!!");
        gotoxy(45,30);printf("Your score: %d",points);Sleep(3000);
    }
    fclose(gf);
    printf("\n\t\tPress any key..");
    getch();
    int flag =0;
    if(strcmp(singleOrNot,"no") == 0)       //"no" parameter to show that player is playing in multiple player mode
    {
        flag = 0;
    }
    else
    {
        gameMenu(userName);
    }
}

void game2(char userName[], int level, char singleOrNot[])
{
    system("cls");
    struct user u[100];
    char enteredWord[30];
    int position = 1;
    int a[96];
    int points = 0;
    int keyPressed = 0;
    time_t start,end;
    FILE *gf,*gl1,*gl2;
    gf = fopen("gameWords.txt","r");
```

```
2350
2351    if(level == 3)
2352    {
2353        gameMenu(userName);
2354    }
2355    system("cls");
2356    printf("\n\n\n\n\n\t\t\t\tWords will be displayed for a limited period of time\n\t\t\t\tcorrectly type in as many words as possible\n\t\t\t\tClock starts ticking as soon as ga
2357    Sleep(5000);
2358    if(position == 1)
2359    {
2360        system("cls");
2361        char ch;
2362        int i=0;
2363        while((ch = getc(gf)) != EOF)
2364        {
2365            fscanf(gf,"%s",u[i].gameWord);
2366            i++;
2367        }
2368
2369        start = time(NULL);
2370        ranArray(a,96);
2371        int j=0;
2372        end = time(NULL);
2373        gotoxy(30,18);
2374        printf("\xB2\xB2\xB2
```

```
        }
        else
        {
            break;
        }

    }

    if(strcmp(singleOrNot,"no") != 0)       //"no" parameter to show that player is playing in multiple player mode
    {
        gl1 = fopen("game1level1.txt","a");
        fprintf(gl1,"%s %d\n",userName,points);
        fclose(gl1);
    }
    gotoxy(45,30);printf("Your score: %d",points);
    Sleep(3000);
}
else if(level == 2)
{
    system("cls");
    char ch;
    int i=0;
    while((ch = getc(gf)) != EOF)
    {
        fscanf(gf,"%s",u[i].gameWord);
        i++;
    }

    start = time(NULL);
    ranArray(a,96);
    int j=0;
    end = time(NULL);
    gotoxy(30,18);
    printf("\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB

    gotoxy(30,27);
    printf("\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB
    int expected_words[100], k=0;
    while(difftime(end,start) < 10)
    {
        gotoxy(45,20);printf("%s",u[a[j]].gameWord);
        Sleep(500);
        gotoxy(45,20);printf("                    ");
        end = time(NULL);
        expected_words[k] = j;
```

```
            k++;
            j++;
        }
        for(int i=0; i<k; i++)
        {
            gotoxy(32,25);printf("Enter the Word:                 ");
            gotoxy(52,25);scanf("%s",enteredWord);
            if(strcmp(enteredWord,u[a[expected_words[i]]].gameWord) == 0)
            {
                points += 2;
            }
            else
            {
                break;
            }

        }

        if(strcmp(singleOrNot,"no") != 0)
        {
            gl2 = fopen("game1level2.txt","a");
            fprintf(gl2,"%s %d",userName,points);
            fclose(gl2);
        }
        // gotoxy(45,29);printf("Time up!!!");
        gotoxy(45,30);printf("Your score: %d",points);
        Sleep(3000);
    }
    fclose(gf);
    printf("\n\n\t\t\tPress any key..");
    getch();
    int flag =0 ;
    if(strcmp(singleOrNot,"no") == 0)
    {
        flag = 0;
    }
    else
    {
        gameMenu(userName);
    }
}

void ranArray(int randomisedArray[],int n)
{
    int i,j,temp;
    srand(time(NULL));
```

```
    srand(time(NULL));

    for(i=0;i<n;i++)
        randomisedArray[i]=i+1;

    for(i=n-1;i>=1;i--)
    {
        j = rand()%(i+1);
        temp = randomisedArray[i];
        randomisedArray[i]=randomisedArray[j];
        randomisedArray[j] = temp;
    }
}

void display_Feedback(char name[], double present_Typing_speed, double present_Typing_accuracy, char filename[])       //displays feedback
{
    struct player p[200],temp;
    FILE *fp;
    char c;
    int l=0,i,j, position=0;

    system("cls");

    printf("\n\n\t==========================================================================================");
    printf("\n\t\t                                User Feedback                                                    ");
    printf("\n\t==========================================================================================\n");

    fp=fopen(filename,"r");
    while((c=getc(fp))!=EOF)        //calculating number of lines
    {
        if(c=='\n')
        {
            l++;
        }
    }
    fclose(fp);

    fp=fopen(filename,"r");
    for(i=0;i<l;i++)        //retrieving data into a structure array
    {
        fscanf(fp,"%s %f %f",p[i].name,&p[i].typing_speed,&p[i].typing_accuracy);
    }
    fclose(fp);

    for(i=0; i<l; i++)      //searching for the user in file
    {
```

```
    {
        if(strcmp(p[i].name, name) == 0)
        {
            position = i;
            break;
        }
    }

    printf("\n\n\t\t---->present speed: %f",present_Typing_speed);
    printf("\t\t---->previous speed : %f",p[position].typing_speed);

    //comparing with past performance and giving suitable feedback

    if(present_Typing_speed > p[position].typing_speed )
    {
        printf("\n\n\t\tHurrayy!!!! you have performed better than last time");
    }
    else if(present_Typing_speed == p[position].typing_speed)
    {
        printf("\n\n\t\tYou are performing consistently, keep it up");
    }
    else
    {
        printf("\n\n\t\tYour performance has decreased, better luck next time");
    }

    printf("\n\n\t\tPress Enter to continue...");
    while((getch())!='\r');
}

int is_Registered_user(char name[], char filename[])        //checks if user is registered
{
    int temp = 1;
    int num_lines = 0, delete_line, counter = 0, user_found = 0;
    char username[50], c, ch;
    struct player p[200];
    FILE *fp;
    FILE *fileptr1, *fileptr2;

    fp=fopen(filename,"r");
    while((c=getc(fp))!=EOF)        //counting number of lines
    {
        if(c=='\n')
        {
            num_lines++;
        }
```

```
    }
    fclose(fp);

    fp=fopen(filename,"r");
    for(int i=0;i<num_lines;i++)        //searching for the user in the file
    {
        counter++;
        fscanf(fp,"%s %f %f",p[i].name,&p[i].typing_speed,&p[i].typing_accuracy);

        if(strcmp(p[i].name, name) == 0)
        {
            printf("\n\n\t\t---------User Found----------\n\n");
            fclose(fp);
            return 1;
        }
    }
    printf("\n\n\t\t---------User Not Found----------\n\n");
    fclose(fp);
    return 0;
}

int levelSelector()
{
    int keyPressed = 0;
    int position = 1;
    while(keyPressed != 13)
    {
        system("cls");

        gotoxy(30,13);
        printf("-----------------------------------------------------");

        gotoxy(30,15);
        printf("\t\t"); arrow(1,position); printf("LEVEL 1");
        gotoxy(30,16);
        printf("\t\t");  arrow(2,position); printf("LEVEL 2");
        gotoxy(30,17);
        printf("\t\t"); arrow(3,position); printf("EXIT");

        gotoxy(30,19);
        printf("-----------------------------------------------------");

        keyPressed = getch();

        if(keyPressed == 80 && position != 3)
            position++;
        else if(keyPressed == 72 && position != 1)
            position--;
        else
            position = position;
    }
    return position;
}
```

3.2.2 Github/Folder Structure

In the Github repository “Typing-Tutor”, We have a folder and a file along with a README. We have uploaded all the text files into a folder named Data Files. Source code “main.c” is uploaded onto the repository. README has a brief description of what the project is all about.

Github/Folder structure:

naveen-manda Add files via upload		0b6dfad 19 minutes ago	3 commits
Data Files	Add files via upload		20 minutes ago
README.md	Initial commit		22 minutes ago
main.c	Add files via upload		19 minutes ago

In Data Files folder

naveen-manda Add files via upload		df92b55 24 minutes ago	History
..			
Quiz.txt	Add files via upload		24 minutes ago
Reading_Leaderboard.txt	Add files via upload		24 minutes ago
Typing_Leaderboard.txt	Add files via upload		24 minutes ago
game1level1.txt	Add files via upload		24 minutes ago
game1level2.txt	Add files via upload		24 minutes ago
gameWords.txt	Add files via upload		24 minutes ago
paragraphsReading.txt	Add files via upload		24 minutes ago
paragraphsTyping.txt	Add files via upload		24 minutes ago
usernames.txt	Add files via upload		24 minutes ago

3.3 TESTING

3.3.1 User Test Cases

Test Case ID: TC01		Use Case ID
Test Case Title: Register		UC01
Test Case Description: Allows new user to register for an account		
Test Steps	Expected Result	Test Steps
1. System displays prompt for user to enter password. 2. User enters a password which does meet the criteria and is a new user.	System should display "Successfully registered" and should add the user to the database.	1. System displays prompt for user to enter password. 2. User enters a password which does meet the criteria and is a new user.

Test Case ID: TC02		Use Case ID
Test Case Title: Register		UC01
Test Case Description: Allows new user to register for an account		
Test Steps	Expected Result	Test Steps
1. System displays prompt for user to enter password. 2. User enters a password which does not meet the criteria or is not a new user.	System should display "Invalid credentials" with respective valid error.	1. System displays prompt for user to enter password. 2. User enters a password which does not meet the criteria or is not a new user.

Test Case ID: TC03		Use Case ID
Test Case Title: Login		UC02
Test Case Description: Allows registered users to access features		
Test Steps	Expected Result	Test Steps
1. System displays prompt for user to login 2. User enters invalid password.	System should display "Invalid credentials".	1. System displays prompt for user to login 2. User enters invalid password.

Test Case ID: TC04		Use Case ID
Test Case Title: Login		UC02
Test Case Description: Allows registered users to access features		
1. System displays prompt for user to login 2. User enters a username and password which is not present in the database.	System should display message referring to user not present in database.	System displays "User Not Found!".

Test Case ID: TC05		Use Case ID
Test Case Title: Leader board		UC03
Test Case Description: Allows users to view the leader board		
Test Steps	Expected Result	Actual Result
1. User chooses to view the leader board.	The leader board should be displayed.	The leader board is displayed.

Test Case ID: TC06		Use Case ID
Test Case Title: Leader board		UC03
Test Case Description: Allows users to view the leader board		
Test Steps	Expected Result	Actual Result
1. User chooses to view the leader board, but there are no users present.	Error message is displayed.	Message is displayed referring to lack of users.

Test Case ID: TC07		Use Case ID
Test Case Title: Typing Speed Test		UC04
Test Case Description: Allows users practice fast typing		
Test Steps	Expected Result	Actual Result
1. User chooses to take the typing test. 2. System displays a random generated passage. 3. User has to type the given	System displays the typing speed of the user and saves it in leader board.	The results are displayed and updated in the leader board file.

passage in limited time.		

Test Case ID: TC08		Use Case ID
Test Case Title: Typing Speed Test		UC04
Test Case Description: Allows users practice fast typing		
Test Steps	Expected Result	Actual Result
1. A non registered user chooses to take the typing test. 2. System displays a random generated passage. 3. User has to type the given passage in limited time.	System displays the typing speed of the user and does not save it in leader board.	The results are displayed and are not updated in the leader board file.

Test Case ID: TC09		Use Case ID
Test Case Title: Reading Speed and Comprehension		UC05
Test Case Description: Allows users to practice on their reading skills		
Test Steps	Expected Result	Actual Result
1. User chooses to take the reading test. 2. System displays a random generated passage. 3. User has to read the given passage in limited time and answer to the quiz.	System displays the reading speed and comprehension of the user and saves it in leader board	The results are displayed and updated in the leader board file.

Test Case ID: TC10		Use Case ID
Test Case Title: Reading Speed and Comprehension		UC05
Test Case Description: Allows users to practice on their reading skills		
Test Steps	Expected Result	Actual Result
1. A non registered user chooses to take the reading test. 2. System displays a random generated passage. 3. User has to read the given passage in limited time and has to take a quiz.	System displays the reading speed and comprehension of the user and does not save it in leader board.	The results are displayed and are not updated in the leader board file.

3.1.1 Admin Test Cases

Test Case ID: TC11		Use Case ID
Test Case Title: Login		UC06
Test Case Description: Allows admin to login		
Test Steps	Expected Result	Actual Result
1. System displays prompt for Admin to login. 2. Admin enters invalid username	System should display message referring to invalid username.	System displays "User not found" error message

Test Case ID: TC12		Use Case ID
Test Case Title: Login		UC06
Test Case Description: Allows admin to login		
Test Steps	Expected Result	Actual Result
1. System displays prompt for Admin to login. 2. Admin enters invalid password.	System should display message referring to invalid credentials.	System displays "Invalid Credentials" as an error message.

Test Case ID: TC13		Use Case ID
Test Case Title: View Leader board		UC07
Test Case Description: Allows admin to look at the leader boards		
Test Steps	Expected Result	Actual Result
1. Admin chooses to view the leader board.	The leader board should be displayed.	The leader board is displayed

Test Case ID: TC14		Use Case ID
Test Case Title: View Leader board		UC07
Test Case Description: Allows admin to look at the leader boards		
Test Steps	Expected Result	Actual Result
1. Admin chooses to view the leader board.	Error message is displayed.	Message is displayed referring to lack of users.

Test Case ID: TC15	Use Case ID
Test Case Title:: Delete complete/particular leader board	

Test Case Description: Allows admin to delete leader boards		
Test Steps	Expected Result	Actual Result
1. System displays prompt to choose the leader board to delete. 2. User chooses a non- empty leader board.	System should display that the particular leader board has been deleted.	System displays that the particular leader board has been deleted successfully.

Test Case ID: TC16		Use Case ID
Test Case Title:: Delete complete/particular leader board		UC08
Test Case Description: Allows admin to delete leader boards		
Test Steps	Expected Result	Actual Result
1. System displays prompt to choose the leader board to delete. 2. User chooses an empty leader board.	System should display error message regarding empty leader board.	System displays "Leader board is empty!" as an error message.

Test Case ID: TC17		Use Case ID
Test Case Title: Access to user credentials		UC09
Test Case Description: Allows admin to view user's credentials		
Test Steps	Expected Result	Actual Result
1. Admin chooses to view the user table.	System should display the user table.	System displays the user table.

Test Case ID: TC18		Use Case ID
Test Case Title: Access to user credentials		UC09
Test Case Description: Allows admin to view user's credentials		
Test Steps	Expected Result	Actual Result
1. Admin chooses to view the user table.	System should display message referring to lack of users.	System displays lack of users as an error message.

Test Case ID: TC19		Use Case ID
Test Case Title: Search for particular user's credentials		UC10
Test Case Description: Allows admin to look some user's credentials		
Test Steps	Expected Result	Actual Result
1. Admin chooses to view a	System should display the	System displays the

particular registered user credentials.	particular user credentials.	particular user credentials.

Test Case ID: TC20		Use Case ID
Test Case Title: Search for particular user's credentials		UC10
Test Case Description: Allows admin to look some user's credentials		
Test Steps	Expected Result	Actual Result
1. Admin chooses to view a particular non registered user credentials.	System should display message referring to user not found.	System displays message referring to user not found.

Test Case ID: TC21		Use Case ID
Test Case Title: Delete User		UC11
Test Case Description: Allows admin to delete a particular User		
Test Steps	Expected Result	Actual Result
1. System prompt to enter username of user whose details you want to edit. 2. User enters a valid username.	System should display that the user has been deleted.	System displays that user has been deleted successfully.

Test Case ID: TC22		Use Case ID
Test Case Title: Delete User		UC11
Test Case Description: Allows admin to delete a particular User		
Test Steps	Expected Result	Actual Result
1. System prompt to enter username of user Admin wants to delete. 2. User enters an invalid username	System should display error message regarding invalid username.	System displays "User Not Found!" as an error message.

3.1.1 Competitor Test Cases

Test Case ID: TC23		Use Case ID
Test Case Title: Games		UC12
Test Case Description: Allows user to play games in either single or multi player mode		
Test Steps	Expected Result	Actual Result
1.System prompts to select the	System should display	System displays all the

game mode (single or multiplayer mode) 2. Competitor selects mode 3. Different games/challenges are displayed	the game according to the user input and let the user play the game	games available for the particular mode selected by the user

Test Case ID: TC24		Use Case ID
Test Case Title: Games		UC12
Test Case Description: Allows user to play games in either single or multi player mode		
Test Steps	Expected Result	Actual Result
1.System prompts to select the game mode (single or multiplayer mode) 2.		

Test Case ID: TC25		Use Case ID
Test Case Title: Join with names		UC13
Test Case Description: Allows competitors to join with the name of their choice		
Test Steps	Expected Result	Actual Result
1.System prompts to enter number of player(in multi player mode) or just a name in single player mode 2. Enters valid names(such that no two names match) 3.System lets the user play games	System should let the user play the game they selected	System lets the user play the game they selected

Test Case ID: TC26		Use Case ID
Test Case Title: Join with names		UC13
Test Case Description: Allows competitors to join with the name of their choice		
Test Steps	Expected Result	Actual Result
1.System prompts to enter number of player(in multi player mode) or just a name in single player mode	System should prompt the user to join with some other name	System displays “Username is already taken, try something else” message

2. User enters a name which is already being entered by some other competitor		

Test Case ID: TC27		Use Case ID
Test Case Title: Participate in challenges(Multi Player Mode)		UC14
Test Case Description: Allows competitors to participate in the game, in turns		
Test Steps	Expected Result	Actual Result
1.System should let competitors to take turns and participate in the games/challenges(in multiplayer mode)	Every competitor must be given chance to play by system	System prompts every competitor to play one after the other

Test Case ID: TC29		Use Case ID
Test Case Title: View Result		UC15
Test Case Description: A leader board is displayed at the end of the game		
Test Steps	Expected Result	Actual Result
1.Competitors must complete their respective turns and press enter	A table showing rankings and individuals scores of competitors. Winner among competitor is declared	Table containing individual scores is displayed and Winner is declared.

4. RESULTS

4.1 Main Loading Screen

4.2 Home Page

C:\Windows\system32\cmd.exe - merge7
HOME

---> REGISTER
LOGIN
ADMIN LOGIN
GAMES
EXIT

4.2 User Test Cases

C:\Windows\system32\cmd.exe - merge7
Enter a UserName: raju
Enter password:
Password must contain atleast 8 characters

C:\Windows\system32\cmd.exe - merge7
Enter a UserName: raju
Enter password: password123
Enter Age: -1
Age must be in the limits of 1-150, please enter valid age

C:\Windows\system32\cmd.exe - merge7
Enter a UserName: raju
Enter password: password123
Enter Age: 24
Enter Gender(select one option):
1. Male 2. Female 3. Others
4
Enter a valid option for Gender

C:\Windows\system32\cmd.exe - merge7
Enter a UserName: raju
Enter password: password123
Enter Age: 24
Enter Gender(select one option):
1. Male 2. Female 3. Others
1
Account is successfully registered

C:\Windows\system32\cmd.exe - merge7
HOME
REGISTER
---> LOGIN
ADMIN LOGIN
GAMES
EXIT

C:\Windows\system32\cmd.exe - merge7
User Login:
Enter a UserName: ra
Invalid Username!

C:\Windows\system32\cmd.exe - merge7
User Login:
Enter a UserName: raju
Enter password: **
Invalid password!

C:\Windows\system32\cmd.exe - merge7
User Login:
Enter a UserName: raju
Enter password:
Too many attempts! Please try again!

C:\Windows\system32\cmd.exe - merge7
User Login:
Enter a UserName: raju
Enter password: **********
Successful login!

C:\Windows\system32\cmd.exe - merge7
TYPING TUTOR

---> TYPING
READING & COMPREHENSION
LEADER BOARD
EXIT

```
C:\Windows\system32\cmd.exe - merge7

Enter your typing duration(sec):  300

Enter a value from 1-120
```

```
C:\Windows\system32\cmd.exe - merge7

                ---------------------------------------------------------

                                    Typing speed

                ---------------------------------------------------------

                        ---------------Passage-----------------

        HTML stands for Hyper Text Markup Language
        HTML is the standard markup language for creating Web pages
        HTML describes the structure of a Web page
        HTML consists of a series of elements
        HTML elements tell the browser how to display the content
        HTML elements label pieces of content.


                ---->Timer automatically starts when you start typing <---

HTML stands for

                !!!!Time over!!!!

                Your typing speed

                ***** 38.40 WPM ******

                Your accuracy

                ----100.0 percentage----

                Press Enter to continue...
```

C:\Windows\system32\cmd.exe - merge7

```
                leaderboard :

        ----------------------------------------------------------------------------
        S.No   Name                                Speed   Accuracy
        ----------------------------------------------------------------------------

         1    raashi                               50.4    95.5
         2    ramesh                               45.6    100.0
         3    siddarth                             43.2    100.0
         4    sushmitha                            43.2    100.0
         5    ravi                                 40.8    100.0
         6    raju                                 38.4    100.0
         7    divya                                34.8    76.3
         8    naveen                               31.2    100.0

Press Enter key to go back to main menu_
```

C:\Windows\system32\cmd.exe - merge7

```
                    TYPING TUTOR

    ----------------------------------------------------

                 TYPING
            ---> READING & COMPREHENSION
                 LEADER BOARD
                 EXIT

    ----------------------------------------------------
```

```
C:\Windows\system32\cmd.exe - merge7

                    ---------------Passage-----------------

        C is a procedural programming language. It was initially developed by Dennis Ritchie
        in the year 1972.It was mainly developed as a system programming language to write an
        operating system. The mainfeatures of C language include low-level access to memory, a
        simple set of keywords, and clean style, thesefeatures make C language suitable for
        system programmings like an operating system or compiler development.Many later
        languages have borrowed syntax/features directly or indirectly from C language. Like
        syntaxof Java, PHP, JavaScript, and many other languages are mainly based on C
        language. C++ is nearly a superset of C language (There are few programs that may
        compile in C, but not in C++).


                    Press Enter if you have completed reading....
```

```
C:\Windows\system32\cmd.exe - merge7

                ------------Reading Comprehension------------

C language was developed by?
Your response : Dennis

ritchie
C language was developed in the year?
Your response : 1972

Which language is a superset of C?
Your response : c++

C language was developed to write?
Your response : english

system
C language suitable for system programming(true/false)?
Your response : true


                ----Your Reading speed is 132.69 WPM----


                ----Your Reading comprehension is 60.00 percentage----


                Press Enter to continue...
```

```
C:\Windows\system32\cmd.exe - merge7

==========================================================================================
                                        User Feedback
==========================================================================================

        ---->present speed: 132.692308          ---->previous speed : 288.000000
        Your performance has decreased, better luck next time
        Press Enter to continue...
```

```
C:\Windows\system32\cmd.exe - merge7

                    TYPING TUTOR

-----------------------------------------------------
                TYPING
                READING & COMPREHENSION
            ---> LEADER BOARD
                EXIT
-----------------------------------------------------
```

C:\Windows\system32\cmd.exe - merge7
---> TYPING SPEED LEADER BOARD
READING SPEED LEADER BOARD

C:\Windows\system32\cmd.exe - merge7
leaderboard :
S.No Name Speed Accuracy
1 raashi 50.4 95.5
2 ramesh 45.6 100.0
3 siddarth 43.2 100.0
4 sushmitha 43.2 100.0
5 ravi 40.8 100.0
6 raju 38.4 100.0
7 divya 34.8 76.3
8 naveen 31.2 100.0
Press Enter key to go back to main menu

C:\Windows\system32\cmd.exe - merge7
TYPING TUTOR
TYPING
READING & COMPREHENSION
LEADER BOARD
---> EXIT
You want to log out?(y/n)_

4.3 Admin Test Cases

C:\Windows\system32\cmd.exe - merge7
Admin Login:
Admin User name: admin
Enter password: ********
Successful login!

C:\Windows\system32\cmd.exe - merge7
A D M I N
---> LEADER BOARD
DELETE LEADERBOARD
DELETE USER
USER'S CREDENTIALS
EXIT

C:\Windows\system32\cmd.exe - merge7
A D M I N
LEADER BOARD
---> DELETE LEADERBOARD
DELETE USER
USER'S CREDENTIALS
EXIT

C:\Windows\system32\cmd.exe - merge7
A D M I N
LEADER BOARD
---> DELETE LEADERBOARD
DELETE USER
USER'S CREDENTIALS
EXIT
Which Leader Board you want to delete?
1.Typing Leader Board
2. Reading Leader Board
3.All

A D M I N
LEADER BOARD
---> DELETE LEADERBOARD
DELETE USER
USER'S CREDENTIALS
EXIT
Which Leader Board you want to dele
1.Typing Leader Board
2. Reading Leader Board
3.All
1
Leaderboard successfully deleted_
Typing_Leaderboard - Notepad
File Edit Format View Help

C:\Windows\system32\cmd.exe - merge7
A D M I N
LEADER BOARD
DELETE LEADERBOARD
---> DELETE USER
USER'S CREDENTIALS
EXIT
Enter a valid UserName:_

```
C:\Windows\system32\cmd.exe - merge7

[redacted]

                    A D M I N

[redacted]

-----------------------------------------------------
                LEADER BOARD
                DELETE LEADERBOARD
           ---> DELETE USER
                USER'S CREDENTIALS
                EXIT
-----------------------------------------------------

[redacted]
          Enter a valid UserName:eeeeeeee

                 ---------User Not Found in usernames.txt----------

                 ---------User Not Found----------

                 ---------User Not Found----------
```

```
C:\Windows\system32\cmd.exe - merge7

[redacted]

                    A D M I N

[redacted]

-----------------------------------------------------
                LEADER BOARD
                DELETE LEADERBOARD
                DELETE USER
           ---> USER'S CREDENTIALS
                EXIT
-----------------------------------------------------

[redacted]
```

```
C:\Windows\system32\cmd.exe - merge7

        -------------------------------------------------------
        S.No.   User Name       Password        Age     gender
        -------------------------------------------------------

          1     sdbkjas         22222222        25      Female
          2     raashi          22222222        20      Female
          3     siddarth        33333333        45      Others
          4     ramesh          44444444        23      Male
          5     sushmitha       66666666        18      Female
          6     ravi            77777777        28      Male
          7     divya           88888888        25      Female

        Search for particular user's details?(y/n)
        Enter a valid user name to search:raashi


        S.No.   User Name       Password        Age     gender

          2      raashi          22222222        20      Female


        Press Enter key to go back to main menu_
```

```
C:\Windows\system32\cmd.exe - merge7

        -------------------------------------------------------
        S.No.   User Name       Password        Age     gender
        -------------------------------------------------------

          1     sdbkjas         22222222        25      Female
          2     raashi          22222222        20      Female
          3     siddarth        33333333        45      Others
          4     ramesh          44444444        23      Male
          5     sushmitha       66666666        18      Female
          6     ravi            77777777        28      Male
          7     divya           88888888        25      Female

        Search for particular user's details?(y/n)
        Enter a valid user name to search:raaaj
Invalid username
```

C:\Windows\system32\cmd.exe - merge7
A D M I N
LEADER BOARD
DELETE LEADERBOARD
DELETE USER
USER'S CREDENTIALS
---> EXIT
You want to log out?(y/n)_

C:\Windows\system32\cmd.exe - merge7
TYPING TUTOR
TYPING
READING & COMPREHENSION
LEADER BOARD
---> EXIT
You want to log out?(y/n)

4.4 Competitor Test Cases

C:\Windows\system32\cmd.exe - merge7
SELECT MODE

---> SINGLE PLAYER
MULTI PLAYER (COMPETE)
EXIT

Enter player name: Raj_

C:\Windows\system32\cmd.exe - merge7
GAME

---> DISAPPEARING WORDS
YOU GOT EM ALL?
EXIT

C:\Windows\system32\cmd.exe - merge7

---> LEVEL 1
LEVEL 2
EXIT

C:\Windows\system32\cmd.exe - merge7
Words will be displayed for a limited period of time
correctly type in as many words as possible
Clock starts ticking as soon as game starts
All the best!!!

C:\Windows\system32\cmd.exe - merge7
Enter the Word: aggressive
Time up!!!
Your score: 8
Press any key..

C:\Windows\system32\cmd.exe - merge7
SELECT MODE
SINGLE PLAYER
---> MULTI PLAYER (COMPETE)
EXIT

C:\Windows\system32\cmd.exe - merge7
COMPETE

---> TYPING CHALLENGE
READING CHALLENGE
DISAPPEARING WORDS (GAME)
YOU GOT EM ALL?
EXIT

C:\Windows\system32\cmd.exe - merge7
Enter the number of players: 20
Enter a number between 1 and 10

C:\Windows\system32\cmd.exe - merge7
Enter player 1 name:
raju_

C:\Windows\system32\cmd.exe - merge7
Enter player 2 name:
Name is already taken, try something else

```
C:\Windows\system32\cmd.exe - merge7

                                        raju's turn

                  ---------------Passage----------------

        One advanced diverted domestic sex repeated bringing you old.
        Possible procured her trifling laughter thoughts property she met way.
        Companions shy had solicitude favourable own.
        Which could saw guest man now heard but.
        Lasted my coming uneasy marked so should.
        Gravity letters it amongst herself dearest an windows by.
        Wooded ladies she basket season age her uneasy saw.
        Discourse unwilling am no described dejection incommode no listening of.
        Before nature his parish boy.


                ---->Timer automatically starts when you start typing <---
```

```
C:\Windows\system32\cmd.exe - merge7

                Results:
        ------------------------------------------------------------------------------
        S.no    Name            Typing Speed    Typing Accuracy
        ------------------------------------------------------------------------------

           1    raju              55.199997          95.833328
           2     ram              45.599998         100.000000
           3    naveen            52.799999         100.000000
           4    david             45.599998          95.000000

                        raju is the winner, Congratulations!!!

Press any key to go back to Menu:_
```

5. Additional Knowledge Acquired

There were many new concepts we learned while working on this project. We learnt how real time applications are built in a very structured manner so that it becomes easy to understand each other's code. We learnt how to work in teams and contribute to this project. This project introduced us to different libraries such as: 'time.h' and 'windows.h'.

We used the 'windows.h' library for setting the display colors and to control the display of the screen with some time delay. The library 'time.h' forms the base of our project and many use cases are dependent on it. Also, we have further improved in our knowledge in file-handling because of the vast amount of data manipulation we have done using text files.

Finally, we learnt how web technologies may help us in adding many other features into this project which were not feasible with just C programming language.

6. Conclusion and Future Work

Typing tutor is a simple but robust tool which calculates, tracks and gives feedback on a user's typing and reading speed.

The features of our project "Typing tutor" were limited to the programming language and technologies we used. There are many more features to be added to this project with the help of other technologies which are more sophisticated and which provide a better graphical user interface. With the usage of these technologies we are willing to make this project accessible to everyone by using web technologies such as HTML, CSS, and JavaScript etc. We are also willing to improve our project in such a way that user's performance is tracked on a long scale by collecting data every time a user logs in and takes tests. This improves the feedback giving procedure. We are confident that we could achieve this huge and organized data storage and retrieval procedure with the help of databases. We are willing to enhance our project with the help of multithreading environment, provided by most of the popular programming languages.

7. References

Stack Overflow (for debugging and knowing new concepts) https://stackoverflow.com/

C Documentation https://devdocs.io/c/

Geeks for Geeks https://www.geeksforgeeks.org/

C language Notes from 1st Semester