

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 my 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

2.1 Software 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)

2.2 Hardware 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.

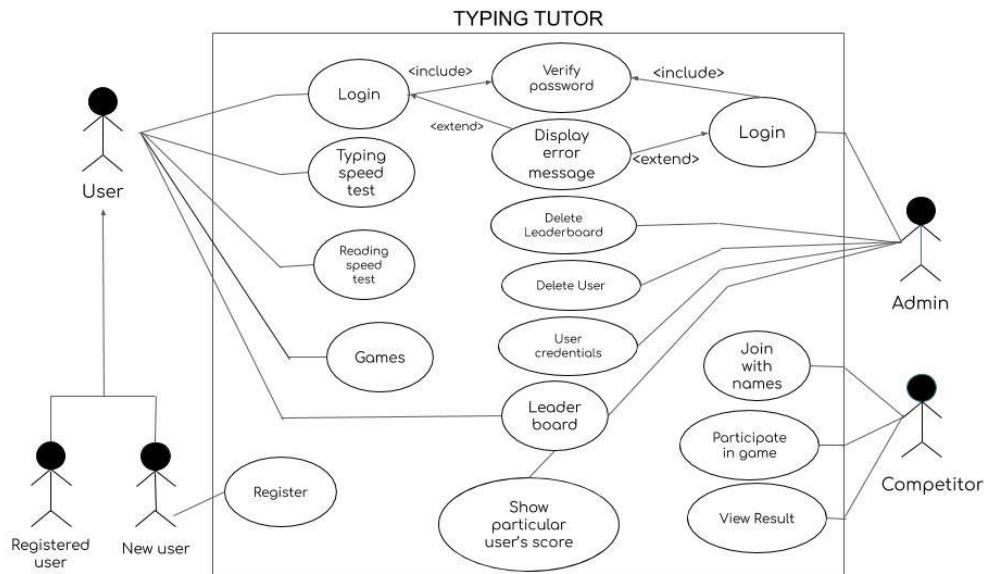
In order to use Typing Tutor, one should have the following:

- **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 is 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

```
1 #include<stdio.h>
2 #include<string.h>
3 #include<stdlib.h>
4 #include<time.h>
5 #include<windows.h>
6
7 COORD coord = {0, 0};
8 // int totalLines = 0;
9
10 void userRegistration();
11 void userLogin();
12 void adminLogin();
13 void mainLoadingScreen();
14 void mainMenu(char UserName[]);
15 void adminMainMenu();
16 void adminUserView();
17 void homePage();
18 void Typingspeed(char UserName[]);
19 void Add_To_Typing_Leaderboard(float s,float a,char UserName[]);
20 void Display_Typing_Leaderboard();
21 void Readingspeed(char UserName[]);
22 double Readingcomprehension(int randomNumber);
23 void Add_To_Reading_Leaderboard(double reading_speed, double reading_comprehension, char UserName[]);
24 void Display_Reading_Leaderboard();
25 void viewLeaderBoards();
26 void Delete_User(char filename[],char userName[]);
27 void Delete_User_credentials(char filename[],char userName[]);
28 void clear_Leaderboard(char filename[]);
29 void compete();
30 void typingCompetition(int typing_duration,float *speed , float *accuracy);
31 void readingCompetition(float *speed , float *accuracy);
32 void games();
33 void gameMenu(char userName[]);
34 void game1(char userName[], int level, char singleOrNot[]);
35 void game2(char userName[], int level, char singleOrNot[]);
36 void ranArray(int randomisedArray[],int n);
37 int is_Registered_user(char name[], char filename[]);
38 void display_Feedback(char name[], double present_Typing_speed, double present_Typing_accuracy, char filename[]);
39 void arrow(int rel,int arrowPosition);
40 int levelSelector();
```



```

455     while((ch = getc(flines)) != EOF)
456     {
457         if(ch == '\n')
458         {
459             totalLines++;
460         }
461     }
462
463     FILE *fr;
464     fr = fopen("usernames.txt","r");
465
466     for(int i=0 ; i<totalLines ; i++)
467     {
468         fscanf(fr, "%s %s %d",u[i].userName,u[i].userPassword,&u[i].userAge,u[i].gender);
469         if(strcmp(u[i].userName,UserName) != 0)           //checking if the username entered already exists in the file
470         {
471             notEqual++;
472         }
473         else
474         {
475             break;
476         }
477     }
478
479     if(notEqual != totalLines)      //if username is already taken, system prompts to try some other usernames
480     {
481         gotoxy(35,20);printf("UserName is already taken, try something different");
482         Sleep(2000);
483         userRegistration();
484     }
485     fclose(fr);
486
487     password:
488     gotoxy(37,10);printf("Enter password:");
489     gotoxy(55,10);scanf("%s",UserPassword);
490
491     if(strlen(UserPassword) < 8)      //checking if password is less than 8 characters
492     {
493         gotoxy(35,20);printf("Password must contain atleast 8 characters");
494         gotoxy(55,10);printf("      ");
495         Sleep(2000);
496         goto password;
497     }
498     gotoxy(35,20);printf("      ");
499     userage:

```

```

500     gotoxy(37,12);printf("Enter Age:");
501     gotoxy(55,12);printf("      ");
502     gotoxy(55,12);scanf("%s",UserAge);
503     int userAge;
504     if(atoi(UserAge) && (atoi(UserAge) >= 1 && atoi(UserAge) <151)) //checking if age is in permissible range
505     {
506         userAge = atoi(UserAge);
507     }
508     else
509     {
510         gotoxy(30,20);printf("Age must be in the limits of 1-150, please enter valid age");
511         Sleep(2000);
512         gotoxy(30,20);printf("      ");
513         goto userage;
514     }
515
516 usergender:
517 gotoxy(37,14);printf("Enter Gender(select one option)");
518 gotoxy(37,15);printf("1. Male 2. Female 3. Others");
519 gotoxy(55,15);printf("      ");
520 gotoxy(55,17);scanf("%s",UserGender);
521 if(atoi(UserGender))
522 {
523     if(atoi(UserGender) == 1)
524     {
525         strcpy(userGenderString,male);
526     }
527     else if(atoi(UserGender) == 2)
528     {
529         strcpy(userGenderString,female);
530     }
531     else if(atoi(UserGender) == 3)
532     {
533         strcpy(userGenderString,others);
534     }
535     else //for invalid gender input
536     {
537         gotoxy(30,20);printf("   Enter a valid option for Gender");
538         Sleep(1000);
539         gotoxy(30,20);printf("      ");
540         goto usergender;
541     }
542 }
543 else //for invalid gender input
544 {
545     gotoxy(30,20);printf("   Enter a valid option for Gender");
546     Sleep(1000);
547     gotoxy(30,19);printf("      ");
548     goto usergender;
549 }
550
551
552
553 FILE *fua, *fpa;
554 fua = fopen("usernames.txt","a");
555
556 //updating the file that contains all the user details
557 fprintf(fua,"%s %s %d %s\n",UserName,UserPassword,userAge,userGenderString);
558
559 fclose(fua);
560 gotoxy(37,20);printf("Account is successfully registered");
561 Sleep(2000);
562 homepage();
563 return;
564 }

```



```

611     fscanf(fr,"%s %s %d %s",u[i].userName,u[i].userPassword,&u[i].userAge,u[i].gender);
612     if(strcmp(u[i].userName,UserName) != 0)
613     {
614         notEqual++;
615         //keep incrementing until the username entered is found in the data file
616     }
617     else
618     {
619         break;
620     }
621
622     char password[30];
623     int l=0;
624
625     if(notEqual != totalLines && count<3)
626     {
627
628         gotoxy(37,19);printf("Enter password:");
629         gotoxy(55,19);
630
631         while(ch != 13) //to print "" when entering password
632         {
633             fflush(stdin);
634             ch = getch();
635             if(ch != 13 && ch != 8) //printing "" when a character other than enter and backspace is entered
636             {
637                 printf("*");
638                 password[l] = ch;
639                 l++;
640             }
641             else if(ch == 8 )
642             {
643                 if(l>0)
644                 {
645                     l--;
646                     password[l] = '\0';
647                     printf("\b \b"); //when user enters backspace
648                 }
649             }
650             if(ch == 13)
651             {
652                 break;
653             }
654         }
655         strcpy(UserPassword,password);

```

```
655     strcpy(UserPassword,password);
656     if(strcmp(UserPassword,u[notEqual].userPassword) != 0)
657     {
658         gotoxy(39,23);printf("Invalid password!");
659         Sleep(1500);
660         gotoxy(39,23);printf("Please try again!");
661         gotoxy(55,19);printf("      ");
662         Sleep(1000);
663         count++;
664         if(count == 3) //allows only three invalid attempts to login
665         {
666             goto counting;
667         }
668         goto start;
669     }
670     else
671     {
672         gotoxy(39,23);printf("Successful login!");
673         Sleep(2000);
674         mainMenu(UserName);
675     }
676 }
677
678 else
679 {
680     gotoxy(39,23);printf("Invalid Username!");
681     Sleep(2000);
682     userLogin();
683 }
684
685 counting:
686 if(count == 3)
687 {
688     gotoxy(35,23);printf("Too many attempts! Please try again!");
689     Sleep(2000);
690     homePage();
691 }
692
693 fclose(fr);
694 }
```

```

739     ch = getch();
740     if(ch != 13 && ch != 8) //printing '*' when a character other than enter and backspace is entered
741     {
742         printf("*");
743         password[l] = ch;
744         l++;
745     }
746     else if(ch == 8) //when user enters backspace
747     {
748         if(l>0)
749         {
750             l--;
751             password[l] = '\0';
752             printf("\b \b");
753         }
754     }
755     if(ch == 13)
756     {
757         break;
758     }
759 }
760 strcpy(adminPassword,password);
761 if(strcmp(adminPassword,adminPass) != 0)
762 {
763     gotoxy(39,23);printf("Invalid password!");
764     Sleep(1500);
765     gotoxy(39,23);printf("Please try again!");
766     gotoxy(55,10);printf("      ");
767     Sleep(1000);
768     count++;
769     if(count == 3) //allows only three invalid attempts to login
770     {
771         goto counting;
772     }
773     goto start;
774 }
775 else
776 {
777     gotoxy(39,23);printf("Successful login!");
778     Sleep(2000);
779     adminMainMenu();
780 }
781 }
782 else
783 {
784     gotoxy(39,23);printf("Invalid Username!");
785     Sleep(2000);
786     adminLogin();
787 }
788 counting:
789 if(count == 3)
790 {
791     gotoxy(39,23);printf("Too many attempts! Please try again!");
792     Sleep(2000);
793     homePage();
794 }
795 }
796 }
```

```

797 ~ void adminUserView()
798 {
799     system("cls");
800     struct user u[200];
801     FILE *fview;
802     fview = fopen("usernames.txt","r");
803     int totalLines = 0;
804
805     char ch;
806     while((ch = getc(fview)) != EOF)
807     {
808         if(ch == '\n')
809         {
810             totalLines++;
811         }
812     }
813
814     FILE *fr;
815     fr = fopen("usernames.txt","r");
816
817     for(int i=0 ; i<totalLines ; i++)
818     {
819         fscanf(fr,"%s %s %d %s",u[i].userName,u[i].userPassword,&u[i].userAge,u[i].gender);
820     }
821
822     printf("\n\n-----\n");
823     printf("S.No.\tUser Name\tPassword\tAge\tGender\n");
824     printf("-----\n\n");
825
826     for(int i=0 ; i<totalLines ; i++)
827     {
828         printf(" %d\t%-12s\t%-8s\t%-3d\t%-10s\n",i+1,u[i].userName,u[i].userPassword,u[i].userAge,u[i].gender);
829     }
830
831     printf("\n\nSearch for particular user's details?(y/n)");
832     char choice = getch();
833     char searchUser[30];
834     int count = 1;
835
836     if(choice == 'y')
837     {
838         printf("\nEnter a valid user name to search:");
839         scanf("%s",searchUser);
840         for(int i=0 ; i<totalLines ; i++)
841         {
842             if(strcmp(searchUser,u[i].userName) == 0)
843             {
844                 printf("User found\n");
845                 count++;
846             }
847         }
848     }

```



```

1070
1071 int typingDuration;
1072 if(atoi(typing_duration) && (atoi(typing_duration) < 120 && atoi(typing_duration) >= 1))
1073 {
1074     typingDuration = atoi(typing_duration);
1075 }
1076 else
1077 {
1078     gotoxy(35,16);printf("Enter a value from 1-120");
1079     Sleep(1000);
1080     gotoxy(71,12);printf("   ");
1081     goto duration;
1082 }
1083
1084 for(int i=0 ; i<atoi(numPlayers) ; i++)      //system prompts competitor to participate in turns
1085 {
1086     system("cls");
1087     printf("\n\n\n\n\n\n\n\n\n\n\n\n%s's turn\n\n",p[i].name);
1088     typingCompetition(typingDuration,&speed,&accuracy);
1089     p[i].typing_speed = speed;
1090     p[i].typing_accuracy = accuracy;
1091 }
1092
1093 for(int i=0 ; i < atoi(numPlayers) ; i++)
1094 {
1095     for(int j=0 ; j < atoi(numPlayers) ; j++)
1096     {
1097         struct player temp;
1098         if(p[i].typing_speed < p[j].typing_speed)
1099         {
1100             temp=p[i];
1101             p[i]=p[j];
1102             p[j]=temp;
1103         }
1104     }
1105 }
1106
1107 system("cls");
1108 printf("\n\nResults: ");
1109 printf("-----");
1110 printf("\n\nS.no\tName\tTyping Speed\tTyping Accuracy");
1111 printf("\n-----");
1112 for(int i=0 ; i<atoi(numPlayers) ; i++)
1113 {
1114     printf("\t%4d\t%4s\t%12f\t%15f",i+1,p[i].name,p[i].typing_speed,p[i].typing_accuracy);
1115 }
1116

```

```

1115
1116
1117
1118 else if(position == 2)
1119 {
1120     for(int i=0 ; i<atoi(numPlayers) ; i++)
1121     {
1122         system("cls");
1123         printf("\n\n\n\n\n\n\n\n\n\n\n\n%s's turn\n\n",p[i].name);
1124         Sleep(3000);
1125         readingCompetition(&speed,&comprehension);
1126         p[i].reading_speed = speed;
1127         p[i].reading_comprehension = comprehension;
1128     }
1129     system("cls");
1130
1131
1132     for(int i=0;i<atoi(numPlayers);i++)
1133     {
1134         for(int j=i;j<atoi(numPlayers);j++)
1135         {
1136             struct player temp;
1137             int x,y;
1138             x = (p[i].reading_speed)*100/(max_reading_speed) + p[i].reading_comprehension;
1139             y = (p[j].reading_speed)*100/(max_reading_speed) + p[j].reading_comprehension;
1140             if(x < y)
1141             {
1142                 temp=p[i];
1143                 p[i]=p[j];
1144                 p[j]=temp;
1145             }
1146         }
1147     }
1148
1149     system("cls");
1150     printf("\n\n\nResults: ");
1151     printf("\n\n-----");
1152     printf("\n\nS.no\tName\tReading Speed\tReading Comprehension");
1153     printf("\n\n-----\n\n");
1154     for(int i=0 ; i<atoi(numPlayers) ; i++)
1155     {
1156         printf("\t%-4d\t%-4s\t%-13.1f\t%-21.1f\n",i+1,p[i].name, p[i].reading_speed,p[i].reading_comprehension);
1157     }
1158
1159     printf("\n\n\n\n\n\n%s is the winner, Congratulations!!!\n\n",p[0].name);
1160 }

```



```

1252 }
1253
1254 char c,name[30];
1255 int i=0;
1256 while((c = getc(gl)) != EOF)
1257 {
1258     fscanf(gl,"%s %d",name,&p[i].gameScore);
1259     i++;
1260 }
1261 fclose(gl);
1262 system("cls");
1263
1264 for(int i=0;i<atoi(numPlayers);i++)
1265 {
1266     for(int j=i;j<atoi(numPlayers);j++)
1267     {
1268         struct player temp;
1269
1270         if(p[i].gameScore < p[j].gameScore)
1271         {
1272             temp=p[i];
1273             p[i]=p[j];
1274             p[j]=temp;
1275         }
1276     }
1277 }
1278 printf("\n\n\tResults: ");
1279 printf("\n\t-----");
1280 printf("\n\tS.no Name\tScore");
1281 printf("\n\t-----\n\n");
1282 for(int i=0 ; i<atoi(numPlayers) ; i++)
1283 {
1284     printf("\t\t%-4d\t%-5d\n",i+1,p[i].name, p[i].gameScore);
1285 }
1286 clear_Leaderboard("game1level1.txt");
1287 clear_Leaderboard("game1level2.txt");
1288 }
1289
1290 printf("Press any key to go back to Menu:");
1291 getch();
1292 compete();
1293 }
1294
1295 > void typingCompetition(int typing_duration,float *speed , float *accuracy) ...
1413
1414 > void readingCompetition(float *speed , float *comprehension) ...

```

```

1295 void typingCompetition(int typing_duration,float *speed , float *accuracy)
1296 {
1297     char ch,c;
1298     char a[2000];
1299     int k=0,count=0,i=0,b,wrong=0, num_letters_typed=0;
1300     float typing_accuracy;
1301     int randomisedArray[9];           // change size according to number of paragraphs
1302     time_t start,end;
1303     FILE *ft;
1304
1305     printf("\n\n\n-----Passage-----\n\n\n");
1306
1307     ft = fopen("paragraphsTyping.txt","r");
1308
1309     ranArray(randomisedArray,9);
1310
1311     int randomNumber = randomisedArray[0];      //takes a random number and prints that numbered paragraph from text file
1312
1313     int parNumber =0;
1314
1315     while((ch = getc(ft)) != EOF)
1316     {
1317         if(ch == '~')    //paragraphs are separated by '~'
1318         {
1319             parNumber++;      //paragraph number in text file
1320             if(parNumber == randomNumber)    //when paragraph number = rand number, print that paragraph
1321             {
1322                 int i = 0;
1323                 printf("\t");
1324                 while((ch=getc(ft)) != '~' && ch != EOF)
1325                 {
1326                     a[i] = ch;      //copy that particular paragraph for furthur usage
1327                     if(ch == '\n')
1328                     {
1329                         printf("\n\t");
1330                     }
1331                     else
1332                     {
1333                         putchar(ch);
1334                     }
1335                     i++;
1336                 }
1337                 a[i] = '\0';
1338             }
1339         }
1340     }

```

```

1341 fclose(ft);
1342
1343 printf("\n\n\n");
1344
1345 printf("\n\n\n---->Timer automatically starts when you start typing <---\n\n\n");
1346
1347 // p=fopen("Typing_file1.txt","r");
1348 int m = 0;
1349 while(a[m] != '\0') //for assessing user input(whether it 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\n!!!!Time over!!!!\n\n");
        break;
    }
}

```

```

1387 }
1388 if(b==1)
1389 {
1390     goto label;
1391 }
1392 k++;
1393 m++;
1394 }
1395
1396 float typing_speed;
1397 char lastScore[5];
1398 typing_speed=((float)num_letters_typed/5)/((float)typing_duration/60); //calculating typing speed
1399 printf("\n\n\tYour typing speed");
1400 printf("\n\n\t***** %.2f WPM *****\n",typing_speed);
1401 *speed = typing_speed;
1402
1403 typing_accuracy=(float)num_letters_typed/(num_letters_typed+wrong)*100; //calculating typing accuracy
1404
1405 printf("\n\n\tYour accuracy\n");
1406 printf("\n\n\t----%.1f percentage----", typing_accuracy);
1407 *accuracy = typing_accuracy;
1408
1409 printf("\n\n\tPress Enter to continue...\n\n");
1410 while((getch())!=\'v\');
1411
1412
1413
1414 void readingCompetition(float *speed , float *comprehension)
1415 {
1416 system("cls");
1417 int length_of_text = 1,k=0;
1418 char ch,a[2000];
1419 double reading_speed, reading_comprehension, time_elapsed;
1420 int randomisedArray[5]; // change size according to number of paragraphs
1421 time_t start,end;
1422 FILE *ft,*fp;
1423
1424 system("cls");
1425 gotoxy(50,20);printf("\nPassage will appear in 3 seconds");
1426 Sleep(3000);
1427
1428 system("cls");
1429
1430 ft = fopen("praghsTyping.txt", "r");
1431 printf("\n\n\n\t-----Passage-----\n\n\n");
1432

```

```

1433 ranArray(randomisedArray,5);
1434
1435 int randomNumber = randomisedArray[0]; //takes a random number and prints that numbered paragraph from text file
1436 // printf("%d",randomNumber);
1437
1438 int parNumber =0;
1439
1440 while((ch = getc(ft)) != EOF)
1441 {
1442     if(ch == '^') //paragraphs are separated by '^'
1443     {
1444         parNumber++;
1445         if(parNumber == randomNumber)
1446         {
1447             int i = 0;
1448             printf("\t");
1449             while((ch=getc(ft)) != '^' && ch != EOF)
1450             {
1451                 a[i] = ch; //copy into a char string for surthur purpose
1452                 if(ch == '\n')
1453                 {
1454                     printf("\n\t");
1455                 }
1456                 else
1457                 {
1458                     putchar(ch);
1459                 }
1460                 i++;
1461             }
1462             a[i] = '\0';
1463         }
1464     }
1465 }
1466 fclose(ft);
1467
1468 printf("\n\n");
1469
1470 start=time(NULL); //storing starting time
1471 printf("\n\nPress Enter if you have completed reading...\n");
1472 while((getch())!='\r');
1473
1474 system("cls");
1475 printf("\n\n");
1476
1477 end = time(NULL); //end time when user completes reading
1478 time_elapsed = difftime(end,start);

```



```

1571     int i = 0;
1572     printf("\t");
1573     while((ch=getc(ft)) != EOF && ch != '\n')
1574     {
1575         a[i] = ch;
1576         if(ch == '\n')
1577         {
1578             printf("\n\t");
1579         }
1580         else
1581         {
1582             putchar(ch);
1583         }
1584         i++;
1585     }
1586     a[i] = '\0';
1587 }
1588 }
1589 fclose(ft);
1590
1591 printf("\n\n\n");
1592
1593 printf("\n\n\n--->Timer automatically starts when you start typing <---\n\n\n");
1594 // p=fopen("Typing_file1.txt","r");
1595 int m = 0;
1596
1597 while(a[m] != '\0')
1598 {
1599     label:
1600     b=0;
1601     c=getch();
1602     if(k==0)      //if entering the loop for the first time
1603     {
1604         start=time(NULL);
1605     }
1606     if(c == a[m])    //if character matches
1607     {
1608         //Beep(600,100);
1609         putchar(c);
1610         if(c==' ')
1611         {
1612             count++;
1613         }
1614         num_letters_typed++;
1615
1616

```

```

1617     else if(c=='r'&& a[m]=='\n') //if 'enter' is pressed
1618     {
1619         //Beep(600,100);
1620         putchar(c);
1621         printf("\n");
1622         num_letters_typed++;
1623     }
1624     else //if character mismatch
1625     {
1626         b=1;
1627         wrong++;
1628         //Beep(600,100);
1629     }
1630     end=time(NULL);
1631
1632     if(difftime(end,start) >= typingDuration) //if time has finished
1633     {
1634         printf("\a");
1635         printf("\n\n\t\tTime over!!!!\n\n");
1636         break;
1637     }
1638     if(b==1)
1639     {
1640         goto label;
1641     }
1642     k++;
1643     m++;
1644 }
1645
1646 float typing_speed;
1647
1648 typing_speed=((float)num_letters_typed/5)/((float)typingDuration/60); //calculating typing speed
1649 printf("\n\n\tYour typing speed");
1650 printf("\n\n\t***** %.2f WPM *****\n\n",typing_speed);
1651
1652 typing_accuracy=(float)num_letters_typed/(num_letters_typed+wrong)*100; //calculating typing accuracy
1653 printf("\n\n\tYour accuracy\n");
1654 printf("\n\n\t----%.1f percentage----", typing_accuracy);
1655 printf("\n\n\tPress Enter to continue...\n\n");
1656
1657
1658 while((getch())!=\r); //enter to continue
1659
1660 Add_To_Typing_Leaderboard(typing_speed, typing_accuracy,UserName); //adding to typing leaderboard
1661
1662 }

```

```
1664 void Add_To_Typing_Leaderboard(float typing_speed, float typing_accuracy, char UserName[])
1665 { //adds user to typing leaderboard
1666     struct player p[200].temp;
1667     FILE *fp;
1668     char c;
1669     int l=0,i,j;
1670
1671     fflush(stdin);
1672
1673     if(is_Registered_user(UserName, "Typing_Leaderboard.txt")) //checking if user is registered
1674     {
1675         printf("-----Registered-----");
1676         display_Feedback(UserName, typing_speed, typing_accuracy, "Typing_Leaderboard.txt"); //displays feedback
1677         Delete_User("Typing_Leaderboard.txt", UserName); //removes the previous record of the user
1678     }
1679
1680     fp=fopen("Typing_Leaderboard.txt", "a");
1681 // printf("*****%s*****\n",UserName);
1682     fprintf(fp,"%s %.1f %.1f\n",UserName, typing_speed, typing_accuracy); //appending results to file
1683     fclose(fp);
1684
1685     Display_Typing_Leaderboard(UserName); //displays the leaderboard
1686
1687 }
1688
1689 void Display_Typing_Leaderboard(char UserName[])
1690 {
1691     struct player p[200].temp;
1692     FILE *fp;
1693     char c;
1694     int l=0,i,j;
1695     system("cls");
1696     printf("\n\nleaderboard :\n\n");
1697     printf("-----");
1698     printf("l\tS.No\tName\tSpeed\tAccuracy");
1699     printf("-----\n\n");
1700
1701     fp=fopen("Typing_Leaderboard.txt", "r");
1702     while((c=getc(fp))!=EOF) //calculating the number of lines in the leaderboard file
1703     {
1704         if(c=='\n')
1705         {
1706             i++;
1707         }
1708     }
1709 }
```

```

1711 fp=fopen("Typing_Leaderboard.txt","r");
1712 for(i=0;i<l;i++)
1713 {
1714     fscanf(fp,"%s %f %f",p[i].name,&p[i].typing_speed,&p[i].typing_accuracy);
1715 }
1716 fclose(fp);
1717
1718 for(i=0;i<l;i++)
1719 {
1720     for(j=i;j<l;j++)
1721     {
1722         if(p[i].typing_speed<p[j].typing_speed)
1723         {
1724             temp=p[i];
1725             p[i]=p[j];
1726             p[j]=temp;
1727         }
1728     }
1729
1730 for(i=0;i<l;i++)
1731 {
1732     printf("\t %4d %4s\t\t%5.1f\t%8.1f\n",i+1,p[i].name,p[i].typing_speed,p[i].typing_accuracy);
1733 }
1734
1735 Sleep(2000);
1736 printf("\nPress Enter key to go back to main menu");
1737 char enter = getch();
1738 if(strcmp(UserName,"admin") == 0)
1739 {
1740     adminMainMenu();
1741 }
1742 else
1743 {
1744     mainMenu(UserName);
1745 }
1746
1747 void Readingspeed(char UserName[])
1748 {
1749     system("cls");
1750     int length_of_text = 1,k=0;
1751     char ch,a[2000];
1752     int randomisedArray[5];
1753     double reading_speed, reading_comprehension, time_elapsed;
1754     time_t start,end;
1755     FILE *ft;
1756
1757     system("cls");
1758     gotoxy(35,20);printf("Passage will appear in 3 seconds");

```

```

1757 Sleep(3000);
1758
1759 system("cls");
1760 ft = fopen("paragraphsReading.txt","r");
1761
1762 printf("\n\n\n\n\t-----Passage-----\n\n\n");
1763
1764 ranArray(randomisedArray,5);
1765
1766 int randomNumber = randomisedArray[0];
1767 // printf("%d",randomNumber);
1768
1769 int parNumber =0;
1770
1771 while((ch = getc(ft)) != EOF) //choosing a random paragraph
{
1772     if(ch == '\r') //paragraphs are separated by '\r'
1773     {
1774         parNumber++;
1775         if(parNumber == randomNumber)
1776         {
1777             int i = 0;
1778             printf("\t");
1779             while((ch=getc(ft)) != '\r' && ch != EOF)
1780             {
1781                 a[i] = ch;
1782                 if(ch == '\n')
1783                 {
1784                     printf("\n\t");
1785                 }
1786                 else
1787                 {
1788                     putchar(ch);
1789                 }
1790                 i++;
1791             }
1792             a[i] = '\0';
1793         }
1794     }
1795 }
1796 fclose(ft);
1797
1798 printf("\n\n");
1799
1800 start=time(NULL); //storing starting time
1801 printf("\n\n\t\t\tPress Enter if you have completed reading....\n");

```

```

1803     while((getch())!=\r);
1804
1805     system("cls");
1806     printf("\n\n");
1807
1808     end = time(NULL); //end time when user completes reading
1809     time_elapsed = difftime(end,start);
1810     printf("Time elapsed : %.0f sec", time_elapsed);
1811
1812     int i = 0;
1813     while(a[i] != '\0') //using a[] char string for finding length of the paragraph
1814     {
1815         if(a[i]=='\n' || a[i]==' ')
1816         {
1817             length_of_text++;
1818         }
1819         i++;
1820     }
1821
1822     reading_speed = (length_of_text/time_elapsed)*60; //calculating reading speed
1823     reading_comprehension = Readingcomprehension(randomNumber); //calculating comprehension
1824     printf("\n\n\t-----Your Reading speed is %.2f WPM-----\n\n", reading_speed);
1825     printf("\n\n\t-----Your Reading comprehension is %.2f percentage-----\n\n", reading_comprehension);
1826
1827     printf("\n\n\tPress Enter to continue... \n\n");
1828     while((getch())!=\r);
1829
1830     Add_To_Reading_Leaderboard(reading_speed, reading_comprehension,UserName); //adding user to reading leaderboard
1831
1832 }
1833
1834 double Readingcomprehension(int randomNumber)
1835 {
1836     system("cls");
1837     printf("\n\n\t-----Reading Comprehension-----\n\n");
1838     int num_questions = 5, points = 0;
1839     int i;
1840     char question[100], answer[20], response[20];
1841     double reading_comprehension;
1842     char ch;
1843     FILE *fp;
1844     fp = fopen("Quiz.txt","r");
1845
1846     for(int count =0; count < randomNumber; count++) //to select random quiz questions
1847     {
1848         while((ch = getc(fp)) != '-' && ch != EOF); //group of questions are separated by '-'

```

```

● 1849
1850
1851
1852
1853     for(i=0; i<num_questions; i++) //displaying questions and getting responses
1854     {
1855         fscanf(fp, "%[^?]\s",question);
1856         fseek(fp, 1, 1);
1857         fscanf(fp, "%s ",answer);
1858         printf("%s ---- %s\n",question, answer);
1859         fflush(stdin);
1860         printf("%s?\nYour response : ", question);
1861         gets(response);
1862         printf("\n");
1863
1864         if(strcmp(response, answer) == 0) //if correct answer
1865         {
1866             points++;
1867         }
1868
1869         reading_comprehension = ((double)points/num_questions) * 100;
1870     }
1871
1872 }
1873
1874 void Display_Reading_Leaderboard(char UserName[])
1875 {
1876     struct player p[200], temp;
1877     FILE *fp;
1878     char c;
1879     float max_reading_speed = 6000.0; //This is an approximate maximum reading speed
1880     int l=0,i,j;
1881     system("cls");
1882     printf("\n\t\tReading_Leaderboard :\n\n");
1883     printf("\n\t-----");
1884     printf("\n\tS.no\tName\t\tReading_speed(WPM)\t\tReading_comprehension(%)");
1885     printf("\n\t-----\n\n");
1886
1887     fp=fopen("Reading_Leaderboard.txt","r");
1888     while((c=getc(fp))!=EOF) //calculating number of lines
1889     {
1890         if(c=='\n')
1891         {
1892             l++;
1893         }
1894     }

```

```

1895 fclose(fp);
1896
1897 fp=fopen("Reading_Leaderboard.txt","r");
1898 for(i=0;i<l;i++) //retrieving data from file to a structure array
1899 {
1900     fscanf(fp,"%s %f %f",p[i].name,&p[i].reading_speed,&p[i].reading_comprehension);
1901 }
1902 fclose(fp);
1903
1904 for(i=0;i<l;i++) //sorting leaderboard taking both speed and comprehension into consideration
1905 {
1906     for(j=i;j<l;j++)
1907     {
1908         int x,y;
1909         x = (p[i].reading_speed)*100/(max_reading_speed) + p[i].reading_comprehension;
1910         y = (p[j].reading_speed)*100/(max_reading_speed) + p[j].reading_comprehension;
1911         if(x < y)
1912         {
1913             temp=p[i];
1914             p[i]=p[j];
1915             p[j]=temp;
1916         }
1917     }
1918 }
1919
1920 for(i=0;i<l;i++) //displaying leaderboard
1921 {
1922     printf("\t %6d%16s%-18.1f\t%-21.1f\n",i+1,p[i].name,p[i].reading_speed,p[i].reading_comprehension);
1923 }
1924 Sleep(2000);
1925 printf("\nPress Enter key to go back to main menu");
1926 char enter = getch();
1927 if(strcmp(UserName,"admin") == 0)
1928     adminMainMenu();
1929 else
1930     mainMenu(UserName);
1931 }
1932
1933 void Add_To_Reading_Leaderboard(double reading_speed, double reading_comprehension, char UserName[])
1934 {
1935     struct player p[200],temp;
1936     FILE *fp;
1937     char c, name[50], filename[50];
1938     int l=0,i,j;
1939
1940

```



```

1986    }
1987    |   position = position;
1988  }
1989  switch (position)
1990  {
1991      |   case 1: Display_Typing_Leaderboard(UserName);
1992      |       break;
1993      |   case 2: Display_Reading_Leaderboard(UserName);
1994      |       break;
1995  }
1996
1997
1998 void Delete_User(char filename[],char userName[]) //deletes a user from all the leaderboards and from the registration file
1999 {
2000     int temp = 1;
2001     int num_lines = 0, delete_line, counter = 0, user_found = 0;
2002     char username[50], c, ch;
2003     struct player p[200];
2004     FILE *fp;
2005     FILE *fileptr1, *fileptr2;
2006
2007     fp=fopen(filename,"r");
2008     while((c=getc(fp))!=EOF) //calculating number of lines
2009     {
2010         if(c=='\n')
2011         {
2012             num_lines++;
2013         }
2014     }
2015     fclose(fp);
2016
2017     fp=fopen(filename,"r");
2018     for(int i=0;i<num_lines;i++) //searching for the user to be deleted
2019     {
2020         counter++;
2021         fscanf(fp,"%s %f %f",p[i].name,&p[i].typing_speed,&p[i].typing_accuracy);
2022
2023         if(strcmp(p[i].name,userName) == 0)
2024         {
2025             delete_line = counter;
2026             user_found = 1;
2027             break;
2028         }
2029     }
2030     // printf("%d",counter);
2031     fclose(fp);
2032

```

```

● 2033  if(user_found != 1) //if user not found
2034  {
2035      printf("\n\n\n\n-----User Not Found-----\n\n");
2036      return;
2037  }
2038  fileptr1 = fopen(filename, "r");
2039  fileptr2 = fopen("replica.txt", "w");
2040
2041  while ((ch = getc(fileptr1)) != EOF)
2042  {
2043      if (temp != delete_line)
2044      {
2045          //copy all lines in file replica.c
2046          putc(ch, fileptr2);
2047      }
2048      if (ch == '\n') temp++;
2049  }
2050  fclose(fileptr1);
2051  fclose(fileptr2);
2052  remove(filename);
2053  //rename the file replica.c to original name
2054  rename("replica.txt", filename);
2055  printf("\n\n\n\n-----User : successfully removed-----\n\n");
2056  }
2057
2058 void Delete_User_credentials(char filename[],char userName[])
2059 {
2060     int temp = 1;
2061     int num_lines = 0, delete_line, counter = 0, user_found = 0;
2062     char username[50], c, ch;
2063     struct user u[200];
2064     FILE *fp;
2065     FILE *fileptr1, *fileptr2;
2066
2067     fp=fopen(filename,"r");
2068     while((c=getc(fp))!=EOF)
2069     {
2070         if(c=='\n')
2071         {
2072             num_lines++;
2073         }
2074     }
2075     fclose(fp);
2076
2077     fp=fopen(filename,"r");
2078     for(int i=0;i<num_lines;i++)

```



```

2305 gotoxy(52,25);scanf("%s",enteredWord);
2306 if(strcmp(enteredWord,u[a[j]]).gameWord) == 0
2307 {
2308     points += 2;
2309 }
2310 end = time(NULL);
2311 j++;
2312 }
2313
2314
2315 if(strcmp(singleOrNot,"no") != 0)
2316 {
2317     g12 = fopen("game1level2.txt","a");
2318     fprintf(g12,"%s %d",userName,points);
2319     fclose(g12);
2320 }
2321 gotoxy(45,29);printf("Time up!!!");
2322 gotoxy(45,30);printf("Your score: %d",points);Sleep(3000);
2323 }
2324 fclose(gf);
2325 printf("\n\n\t\tPress any key..");
2326 getch();
2327 int flag =0;
2328 if(strcmp(singleOrNot,"no") == 0) // "no" parameter to show that player is playing in multiple player mode
2329 {
2330     flag = 0;
2331 }
2332 else
2333 {
2334     gameMenu(userName);
2335 }
2336 }
2337
2338 void game2(char userName[], int level, char singleOrNot[])
2339 {
2340     system("cls");
2341     struct user u[100];
2342     char enteredWord[30];
2343     int position = 1;
2344     int a[96];
2345     int points = 0;
2346     int keyPressed = 0;
2347     time_t start,end;
2348     FILE *gf,*g11,*g12;
2349     gf = fopen("gameWords.txt","r");
2350

```



```

2441     k++;
2442     j++;
2443 }
2444 for(int i=0; i<k; i++)
2445 {
2446     gotoxy(32,25);printf("Enter the Word:      ");
2447     gotoxy(52,25);scanf("%s",enteredWord);
2448     if(strcmp(enteredWord,u[a[expected_words[i]]].gameWord) == 0)
2449     {
2450         points += 2;
2451     }
2452     else
2453     {
2454         break;
2455     }
2456 }
2457
2458 if(strcmp(singleOrNot,"no") != 0)
2459 {
2460     gI2 = fopen("game1level2.txt","a");
2461     fprintf(gI2,"%s %d",userName,points);
2462     fclose(gI2);
2463 }
2464 // gotoxy(45,29);printf("Time up!!!");
2465 gotoxy(45,30);printf("Your score: %d",points);
2466 Sleep(3000);
2467 }
2468 fclose(gf);
2469 printf("\n\n\n\n!Press any key..");
2470 getch();
2471 int flag =0 ;
2472 if(strcmp(singleOrNot,"no") == 0)
2473 {
2474     flag = 0;
2475 }
2476 else
2477 {
2478     gameMenu(userName);
2479 }
2480 }
2481 }
2482 void ranArray(int randomisedArray[],int n)
2483 {
2484     int i,j,temp;
2485     srand(time(NULL));

```

```

2486     srand(time(NULL));
2487
2488     for(i=0;i<n;++)
2489     {
2490         randomisedArray[i]=i+1;
2491
2492         for(i=n-1;i>=1;i--)
2493         {
2494             j = rand()%(i+1);
2495             temp = randomisedArray[i];
2496             randomisedArray[i]=randomisedArray[j];
2497             randomisedArray[j] = temp;
2498         }
2499     }
2500
2501     void display_Feedback(char name[], double present_Typing_speed, double present_Typing_accuracy, char filename[]) //displays feedback
2502     {
2503         struct player p[200],temp;
2504         FILE *fp;
2505         char c;
2506         int l=0,i,j, position=0;
2507
2508         system("cls");
2509
2510         printf("\n\n\n");
2511         printf("          User Feedback");
2512         printf("\n\n\n");
2513
2514         fp=fopen(filename,"r");
2515         while((c=getc(fp))!=EOF) //calculating number of lines
2516         {
2517             if(c=='\n')
2518             {
2519                 l++;
2520             }
2521         }
2522         fclose(fp);
2523
2524         fp=fopen(filename,"r");
2525         for(l=0;l<1;l++) //retrieving data into a structure array
2526         {
2527             fscanf(fp,"%s %f %f",p[l].name,&p[l].typing_speed,&p[l].typing_accuracy);
2528         }
2529         fclose(fp);
2530
2531         for(l=0; l<1; l++) //searching for the user in file
2532         {

```

```

2531 {
2532     if(strcmp(p[i].name, name) == 0)
2533     {
2534         position = i;
2535         break;
2536     }
2537 }
2538
2539 printf("\n\n\t\t->present speed: %f",present_Typing_speed);
2540 printf("\t\t->previous speed : %f",p[position].typing_speed);
2541
2542 //comparing with past performance and giving suitable feedback
2543
2544 if(present_Typing_speed > p[position].typing_speed )
2545 {
2546     printf("\n\n\t\tHurray!!! you have performed better than last time");
2547 }
2548 else if(present_Typing_speed == p[position].typing_speed)
2549 {
2550     printf("\n\n\t\tYou are performing consistently, keep it up");
2551 }
2552 else
2553 {
2554     printf("\n\n\t\tYour performance has decreased, better luck next time");
2555 }
2556
2557 printf("\n\n\t\tPress Enter to continue... ");
2558 while((getch())!='\r');
2559 }
2560
2561 int is_Registered_user(char name[], char filename[]) //checks if user is registered
{
2562     int temp = 1;
2563     int num_lines = 0, delete_line, counter = 0, user_found = 0;
2564     char username[50], c, ch;
2565     struct player p[200];
2566     FILE *fp;
2567     FILE *fileptr1, *fileptr2;
2568
2569     fp=fopen(filename,"r");
2570     while((c=getc(fp))!=EOF) //counting number of lines
2571     {
2572         if(c=='\n')
2573         {
2574             num_lines++;
2575         }

```

```

2577     }
2578     fclose(fp);
2579
2580     fp=fopen(filename,"r");
2581     for(int i=0;i<num_lines;i++) //searching for the user in the file
2582     {
2583         counter++;
2584         fscanf(fp,"%s %f %f",p[i].name,&p[i].typing_speed,&p[i].typing_accuracy);
2585
2586         if(strcmp(p[i].name, name) == 0)
2587         {
2588             printf("\n\n\t-----User Found-----\n\n");
2589             fclose(fp);
2590             return 1;
2591         }
2592     }
2593     printf("\n\n\t-----User Not Found-----\n\n");
2594     fclose(fp);
2595     return 0;
2596 }
2597
2598 int levelSelector()
2599 {
2600     int keyPressed = 0;
2601     int position = 1;
2602     while(keyPressed != 13)
2603     {
2604         system("cls");
2605
2606         gotoxy(30,13);
2607         printf("-----");
2608
2609         gotoxy(30,15);
2610         printf("\t\t"); arrow(1,position); printf("LEVEL 1");
2611         gotoxy(30,16);
2612         printf("\t\t"); arrow(2,position); printf("LEVEL 2");
2613         gotoxy(30,17);
2614         printf("\t\t"); arrow(3,position); printf("EXIT");
2615
2616         gotoxy(30,19);
2617         printf("-----");
2618
2619         keyPressed = getch();
2620
2621         if(keyPressed == 80 && position != 3)
2622         {
2623             position++;
2624             else if(keyPressed == 72 && position != 1)
2625             {
2626                 position--;
2627             else
2628                 position = position;
2629         }
2630         return position;
2631     }

```

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
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.
	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.

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
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.
	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.

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
1. System displays prompt for user to login 2. User enters invalid password.	System should display "Invalid credentials".
	Test Steps
	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		
Test Case Description: Allows admin to login	UC06	
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		
Test Case Description: Allows admin to login	UC06	
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		
Test Case Description: Allows admin to look at the leader boards	UC07	
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		
Test Case Description: Allows admin to look at the leader boards	UC07	
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
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
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
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

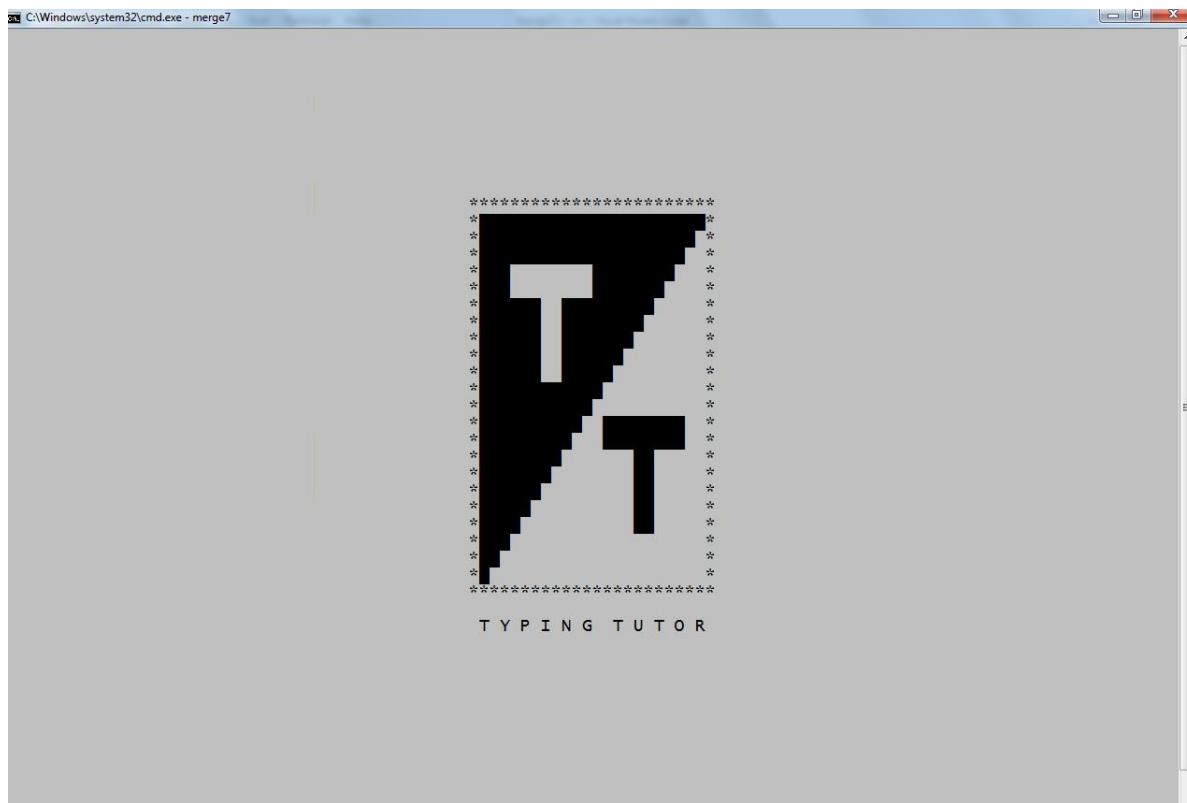
2. User enters a name which is already being entered by some other competitor		
---	--	--

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

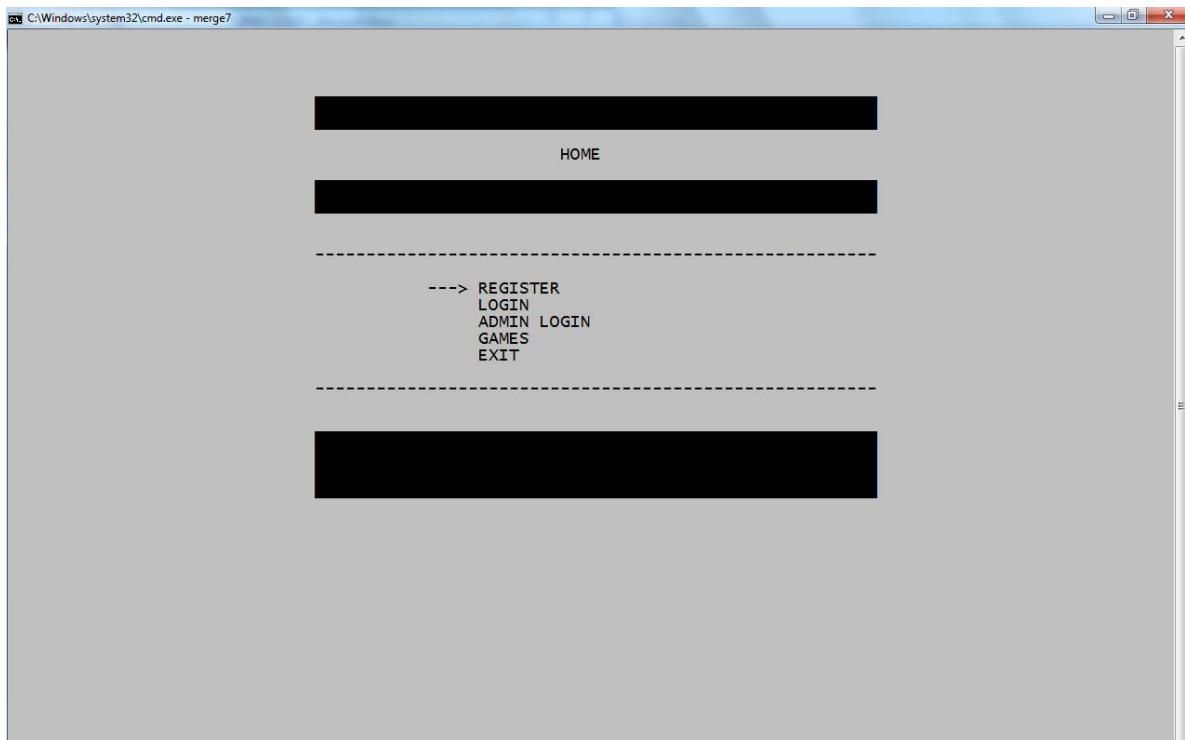
Test Case ID: TC29	Use Case ID	
Test Case Title: View Result		
Test Case Description: A leader board is displayed at the end of the game	UC15	
Test Steps	Expected Result	Actual Result
1.Competitors must complete their respective turns and press enter	A table showing rankings and individuals scores of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

4. RESULTS

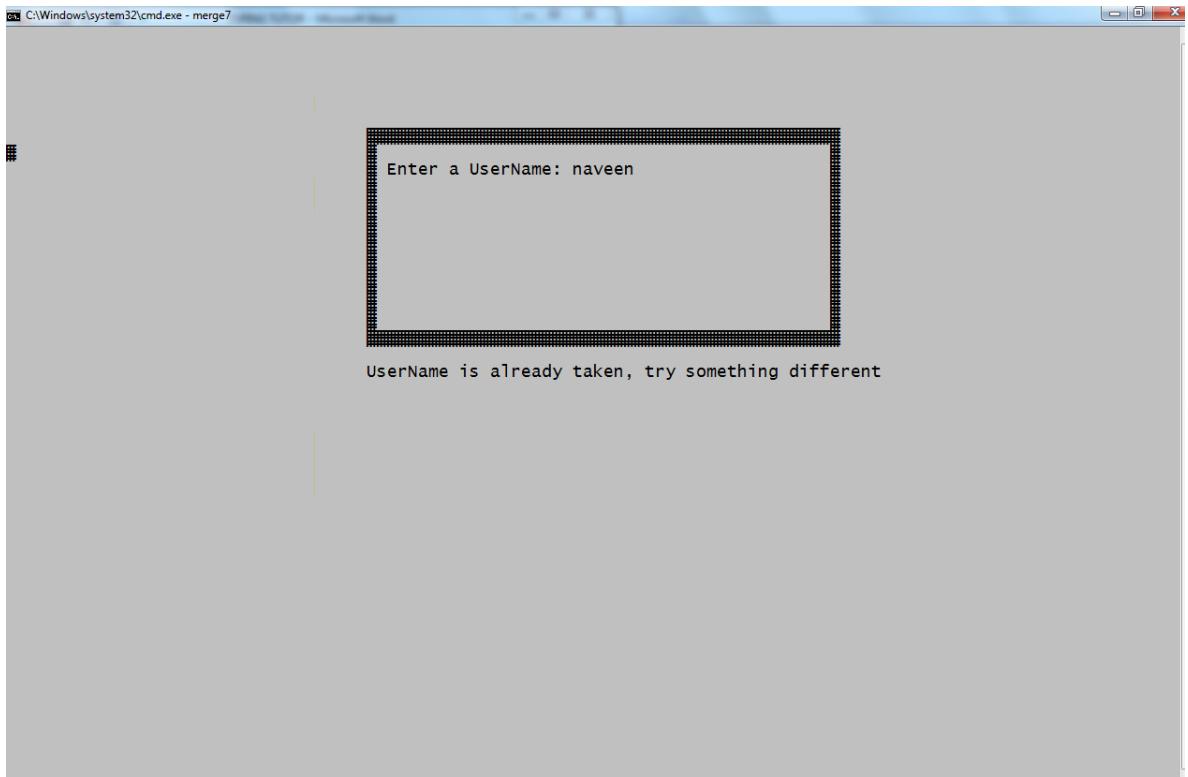
4.1 Main Loading Screen

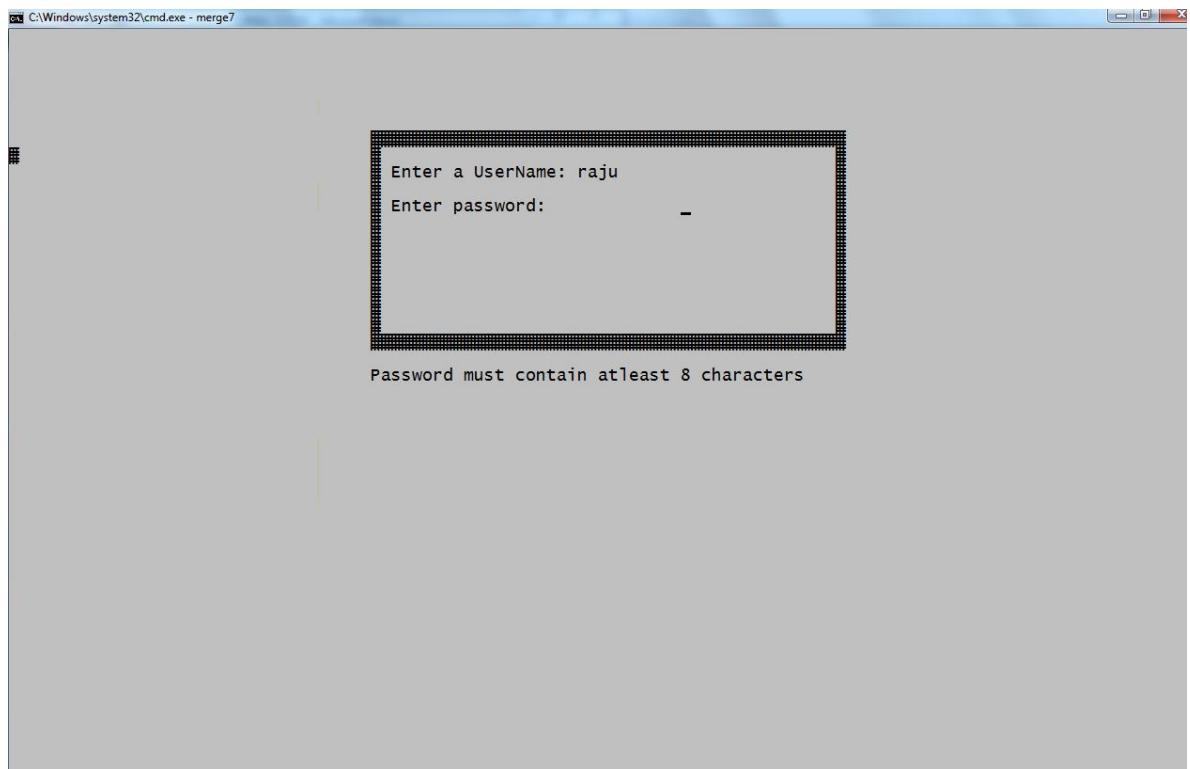


4.2 Home Page



4.2 User Test Cases





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

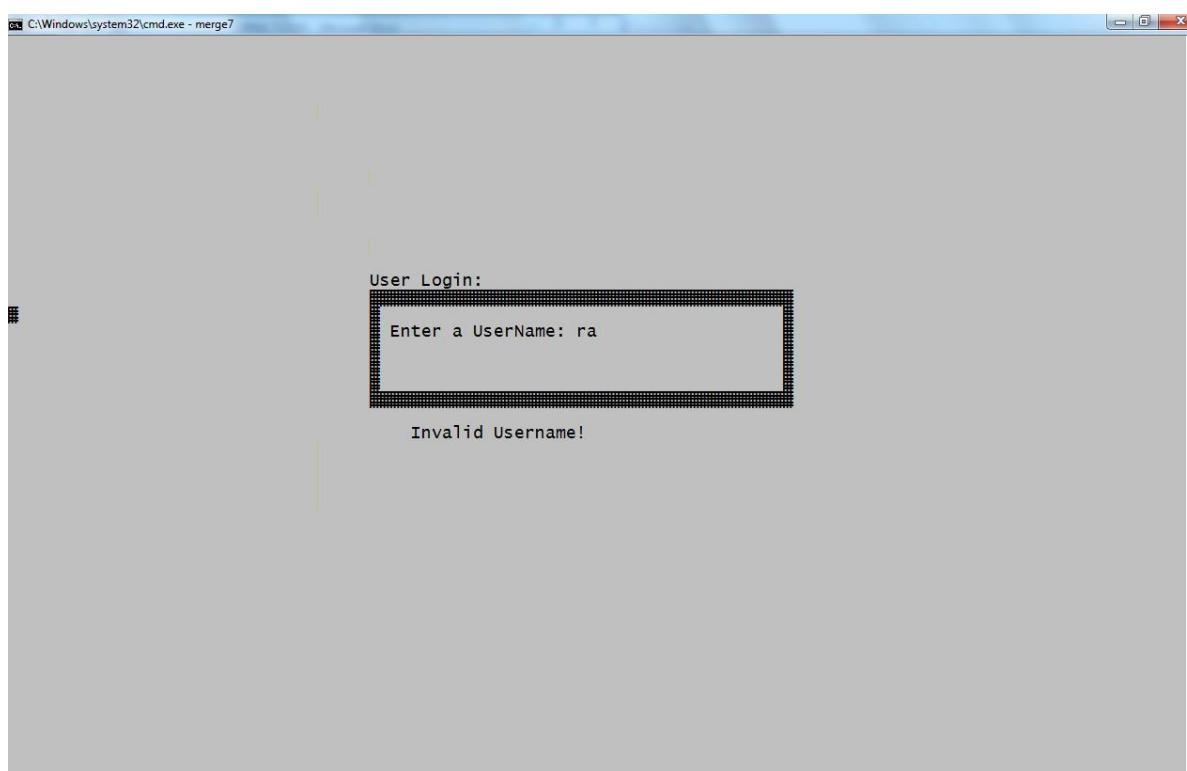
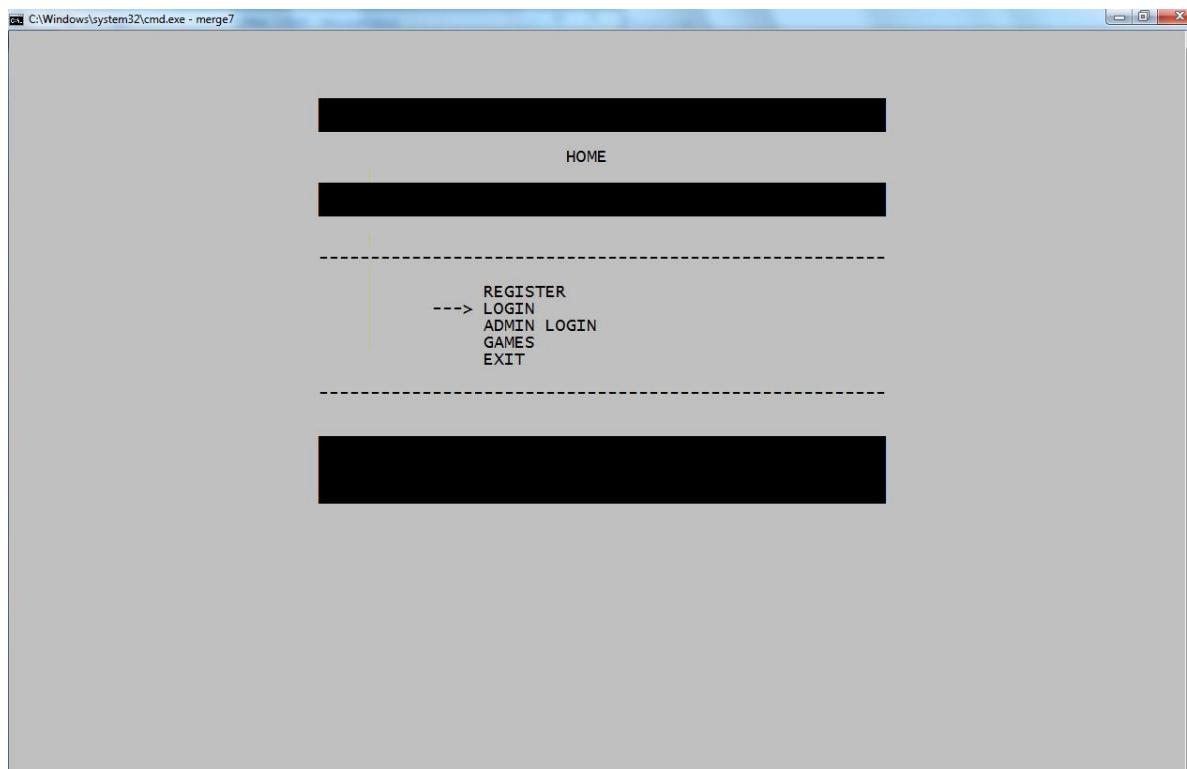
Enter a UserName: raju
Enter password: password123
Enter Age: 24
Enter Gender(select one option):
1. Male 2. Female 3. Others
4

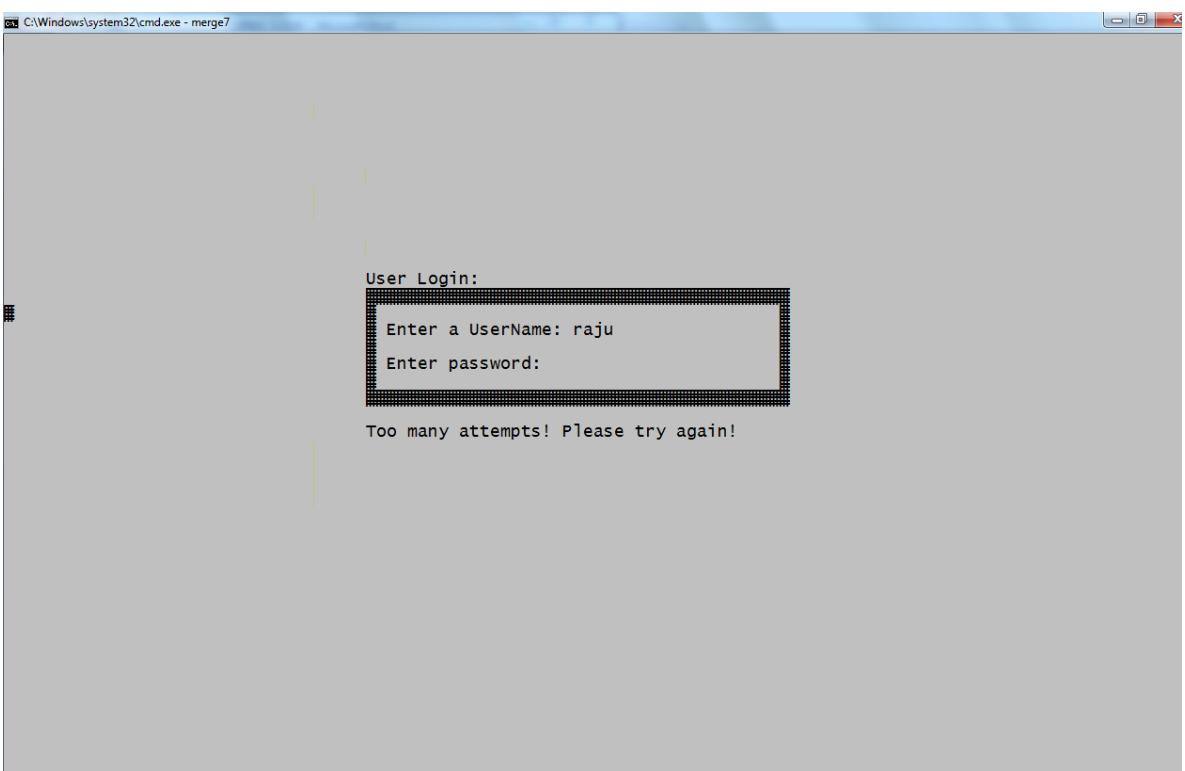
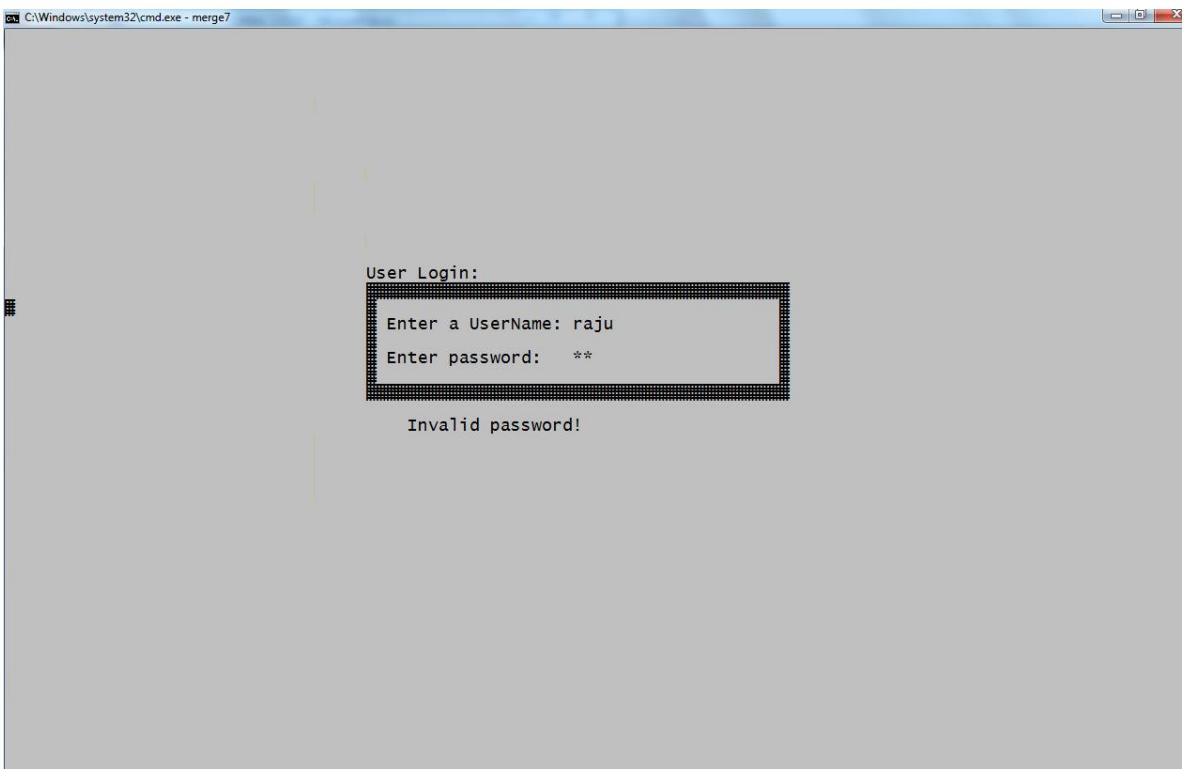
Enter a valid option for Gender
```

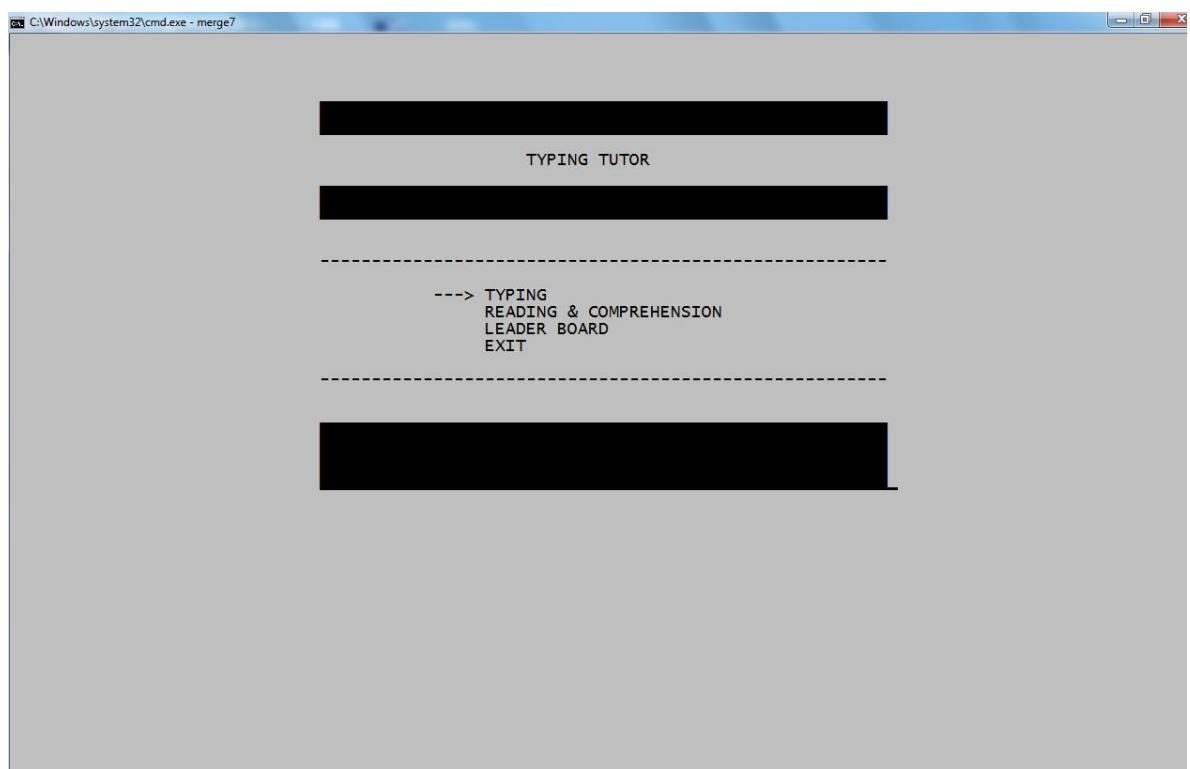
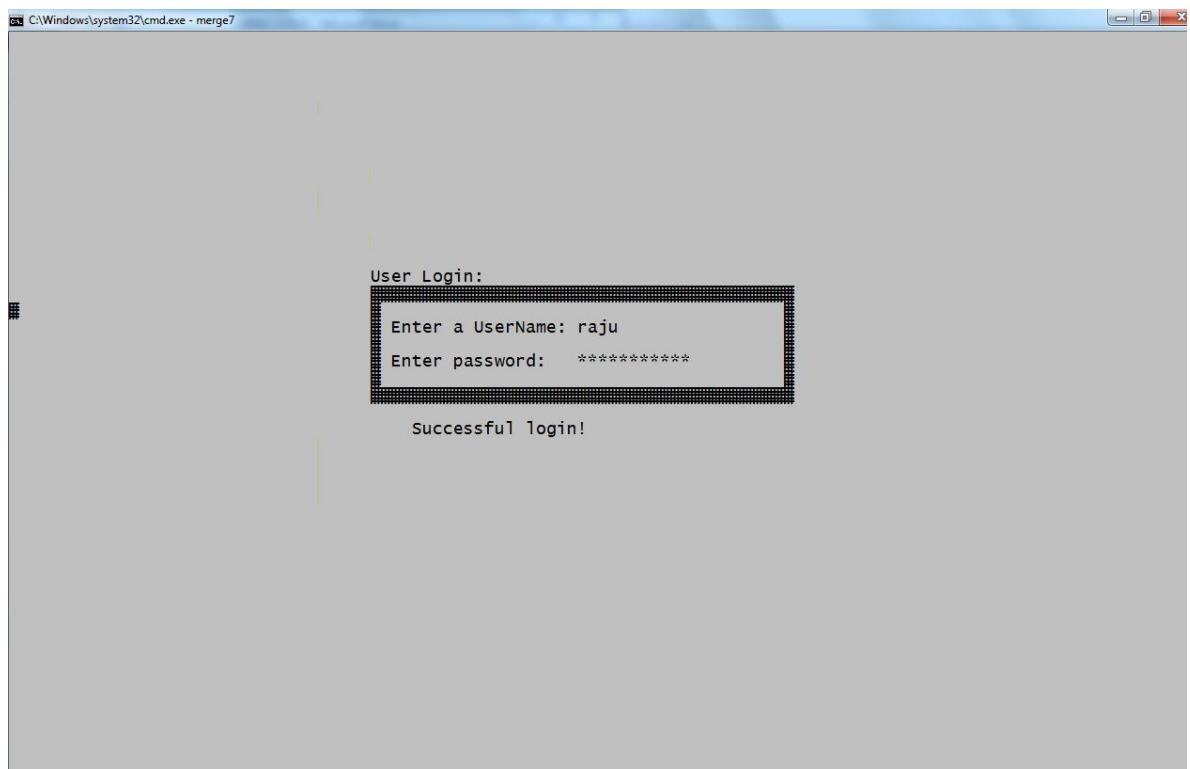
```
C:\Windows\system32\cmd.exe - merge7

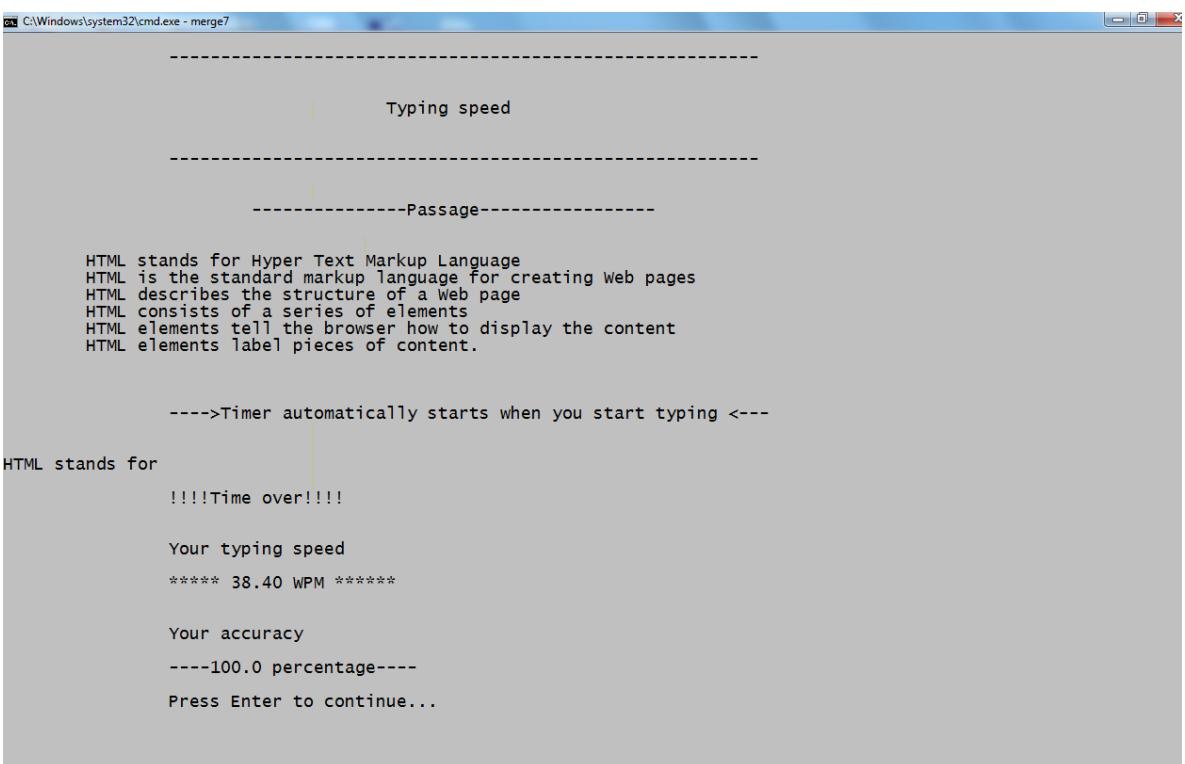
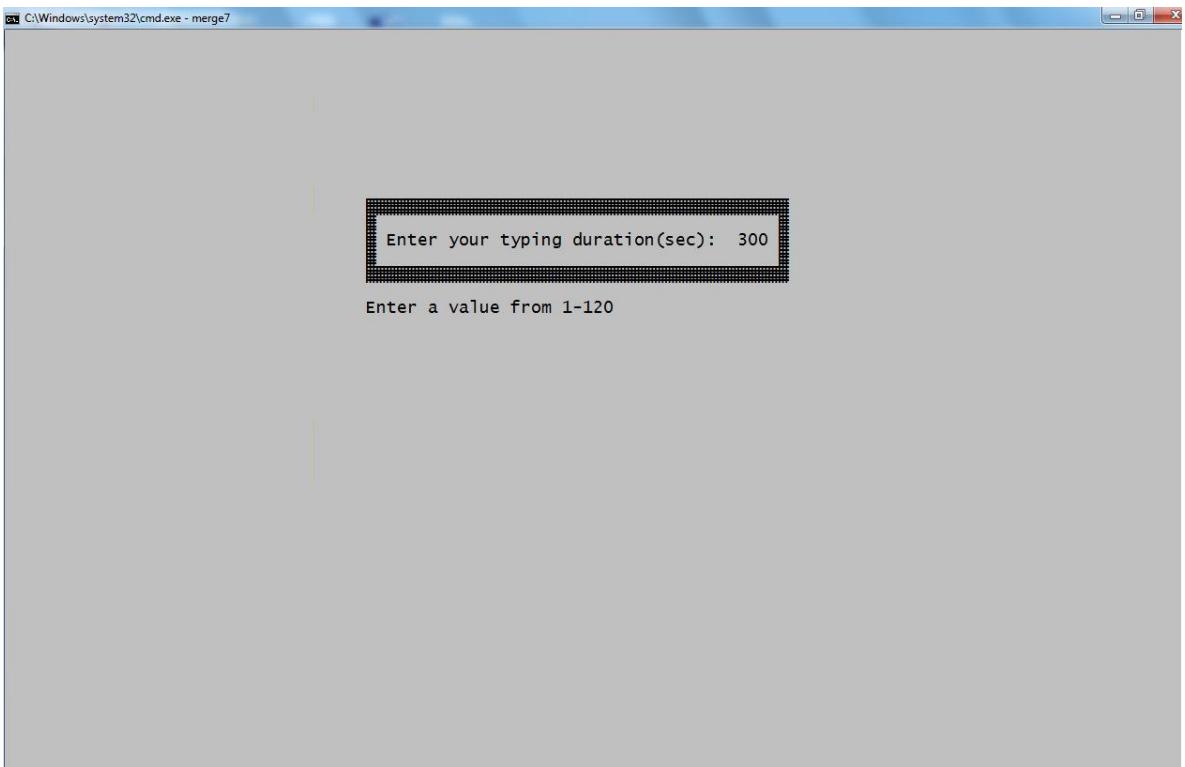
Enter a UserName: raju
Enter password: password123
Enter Age: 24
Enter Gender(select one option):
1. Male 2. Female 3. Others
1

Account is successfully registered
```









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_

[REDACTED]

TYPING TUTOR

[REDACTED]

TYPING
---> READING & COMPREHENSION
LEADER BOARD
EXIT

[REDACTED]

```
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 main features of C language include low-level access to memory, a
simple set of keywords, and clean style, these features make C language suitable for
system programming like an operating system or compiler development. Many later
languages have borrowed syntax/features directly or indirectly from C language. Like
syntax of 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

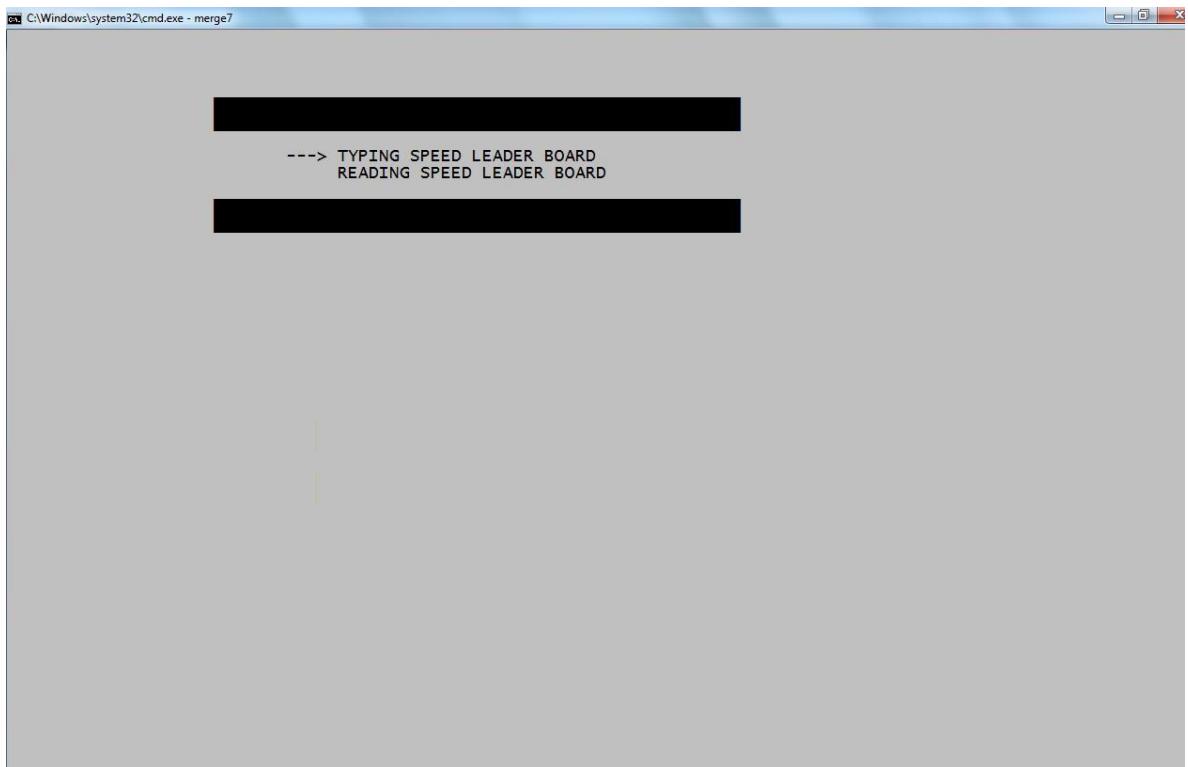
=====
User Feedback
=====

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

The screenshot shows a Windows command prompt window with the title bar 'C:\Windows\system32\cmd.exe - merge7'. The window contains a menu for a 'Typing Tutor' application. The menu items are:

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

Large black rectangular redaction boxes are present at the top and bottom of the menu area.

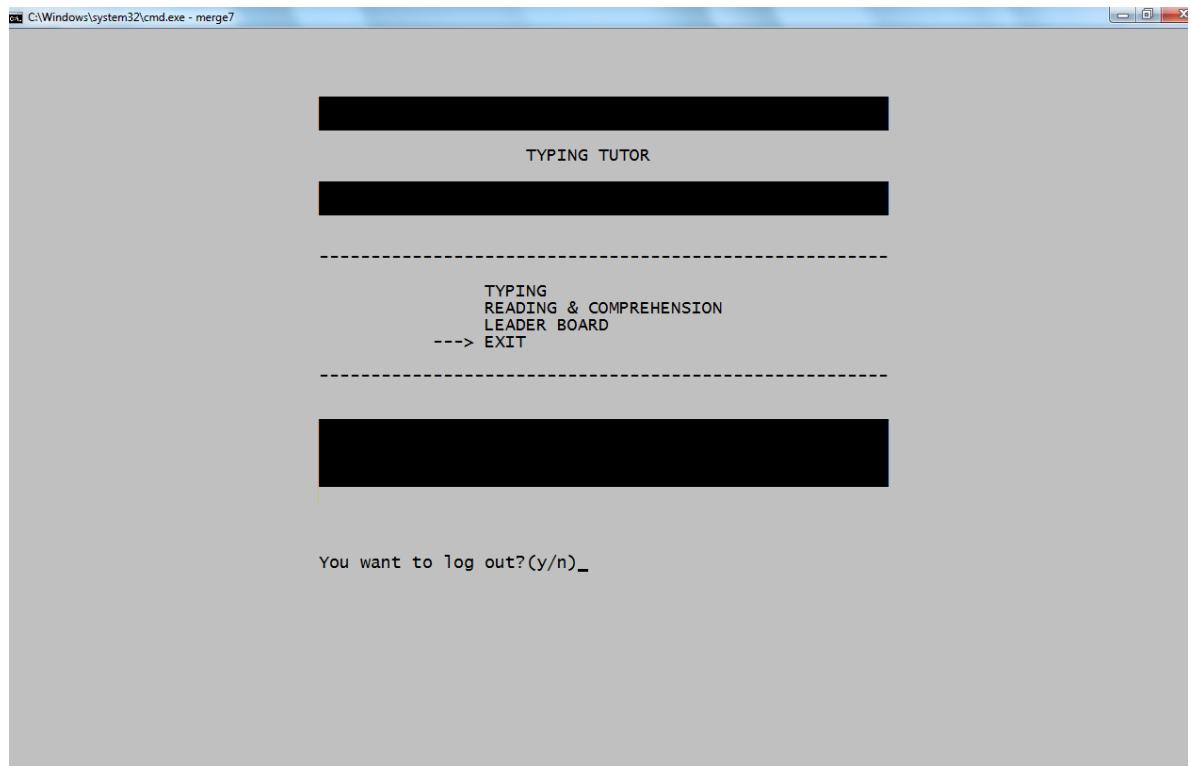


The screenshot shows a Windows command prompt window titled 'C:\Windows\system32\cmd.exe - merge7'. The window contains the following text:

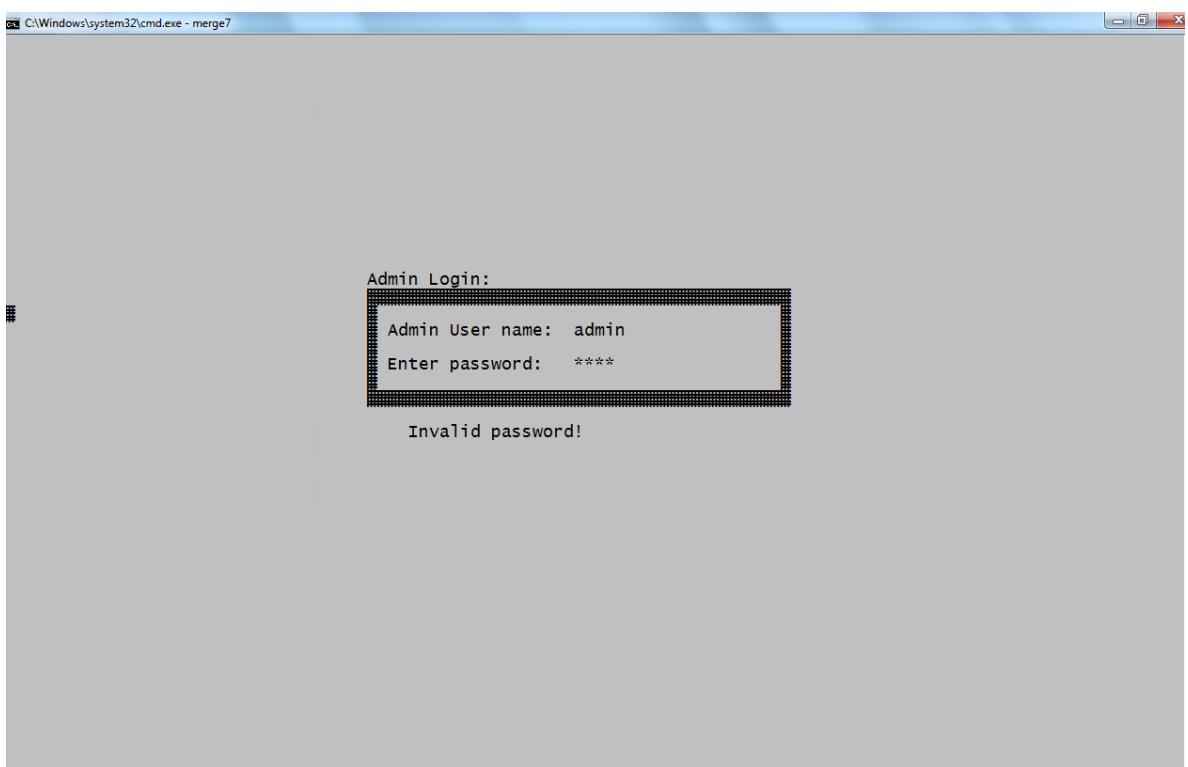
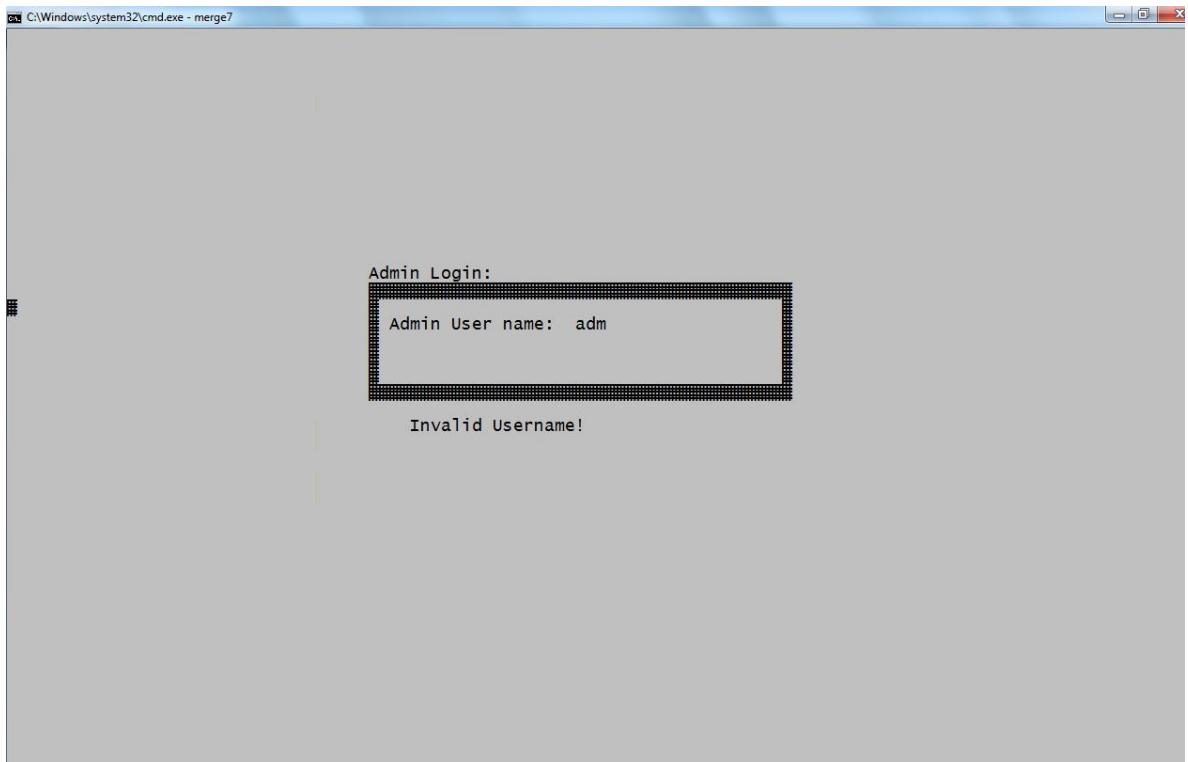
leaderboard :

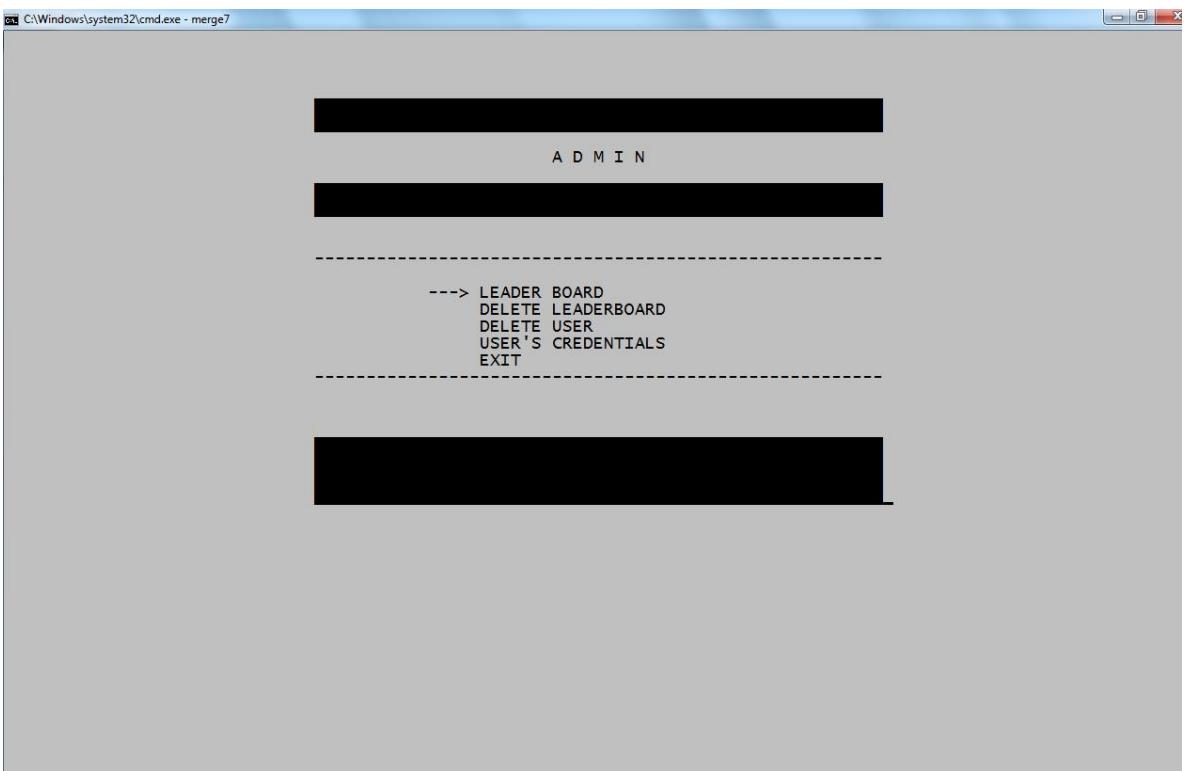
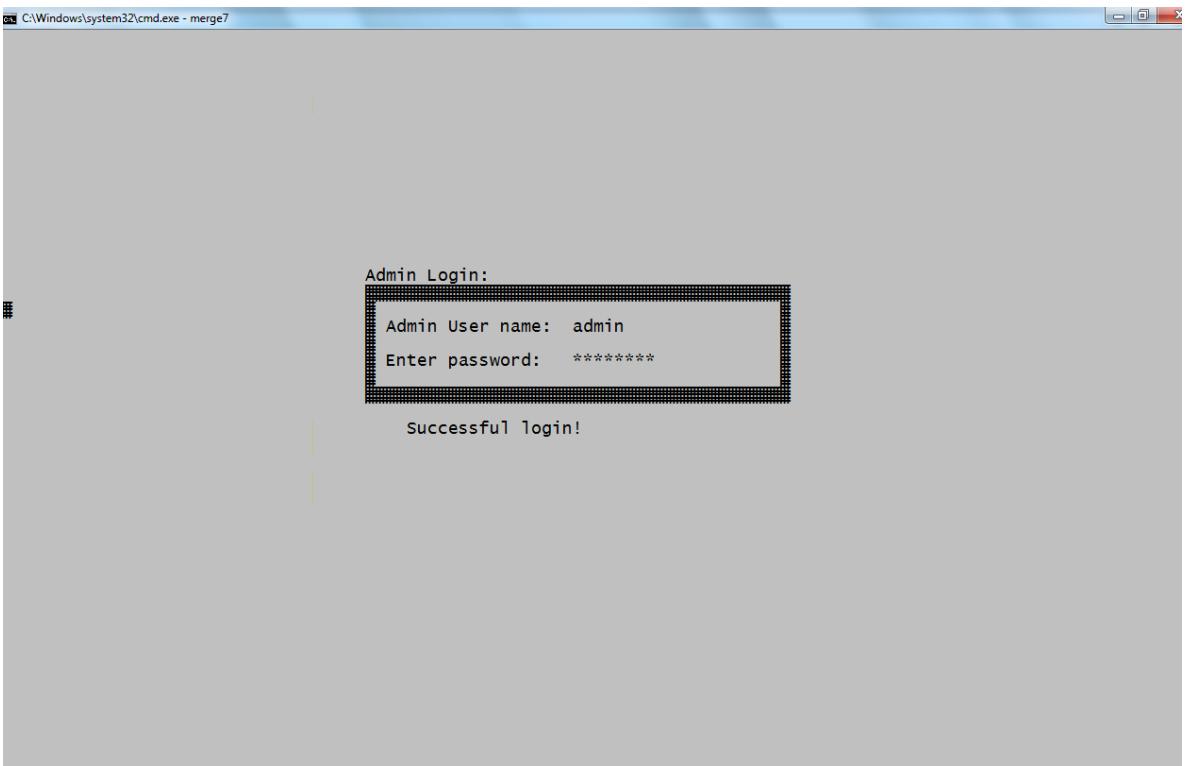
S.No	Name	Speed	Accuracy
1	raashi	50.4	95.5
2	ramesh	45.6	100.0
3	siddarth	43.2	100.0
4	sushmitha	43.2	100.0
5	ravi	40.8	100.0
6	raju	38.4	100.0
7	divya	34.8	76.3
8	naveen	31.2	100.0

Press Enter key to go back to main menu



4.3 Admin Test Cases

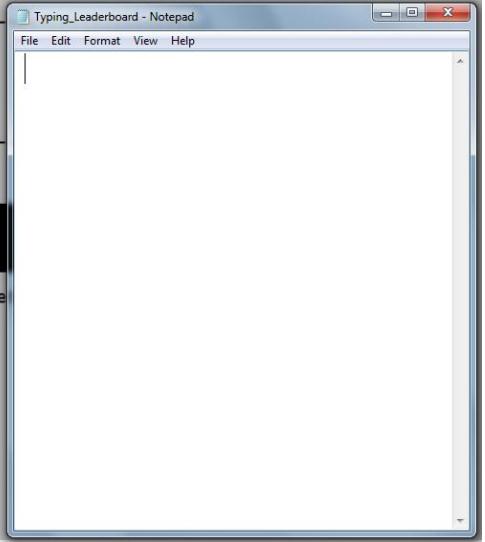




```
C:\Windows\system32\cmd.exe - merge7
[REDACTED]
ADMIN
[REDACTED]
-----
LEADER BOARD
---> DELETE LEADERBOARD
DELETE USER
USER'S CREDENTIALS
EXIT
[REDACTED]
```

```
C:\Windows\system32\cmd.exe - merge7
[REDACTED]
ADMIN
[REDACTED]
-----
LEADER BOARD
---> DELETE LEADERBOARD
DELETE USER
USER'S CREDENTIALS
EXIT
[REDACTED]
Which Leader Board you want to delete?
1.Typing Leader Board
2. Reading Leader Board
3.All
```

```
[REDACTED]  
A D M I N  
[REDACTED]  
-----  
LEADER BOARD  
---> DELETE LEADERBOARD  
DELETE USER  
USER'S CREDENTIALS  
EXIT  
-----  
[REDACTED]  
Which Leader Board you want to delete:  
1.Typing Leader Board  
2. Reading Leader Board  
3.All  
1  
Leaderboard successfully deleted_
```



```
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:_
```

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

[REDACTED]
ADM IN
[REDACTED]

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

[REDACTED]

Enter a valid UserName:eeeeeee
-----User Not Found in usernames.txt-----
-----User Not Found-----
-----User Not Found-----
```

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

[REDACTED]
ADM IN
[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:raaj
Invalid username
```

```
C:\Windows\system32\cmd.exe - merge7
[REDACTED]
ADM IN
[REDACTED]

-----
LEADER BOARD
DELETE LEADERBOARD
DELETE USER
USER'S CREDENTIALS
--> EXIT
[REDACTED]

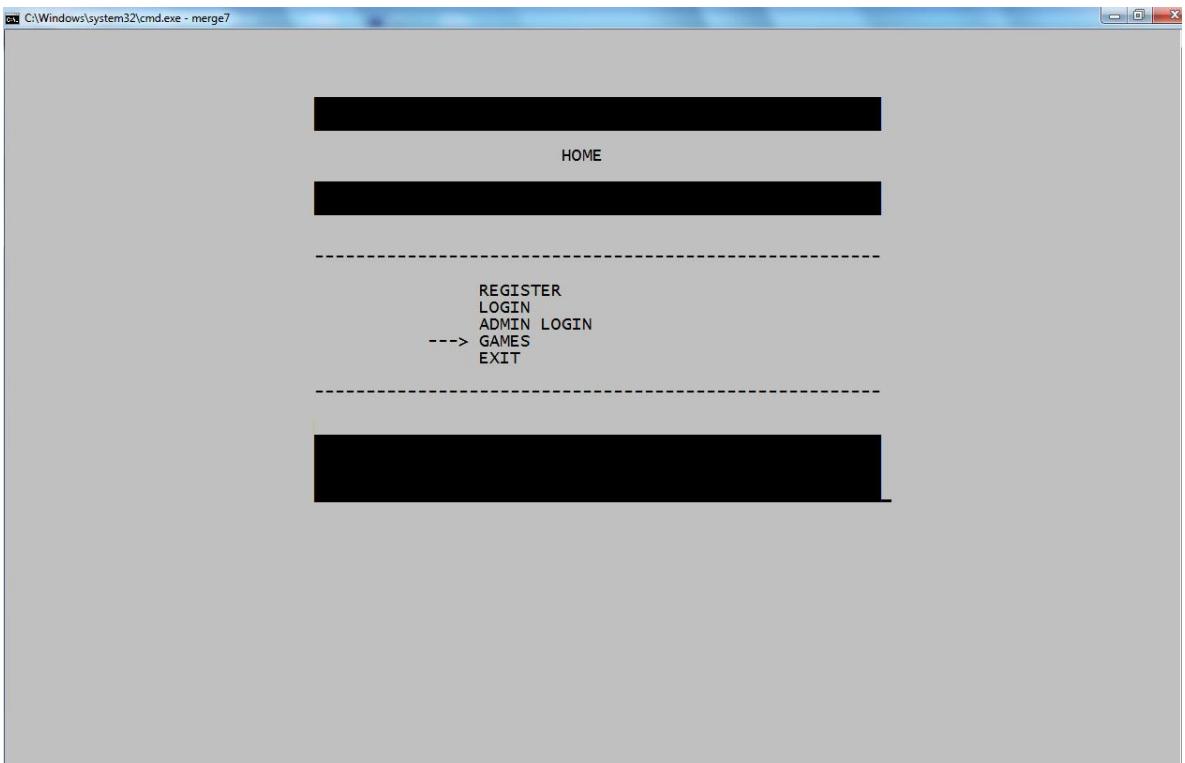
You want to log out?(y/n) _
```

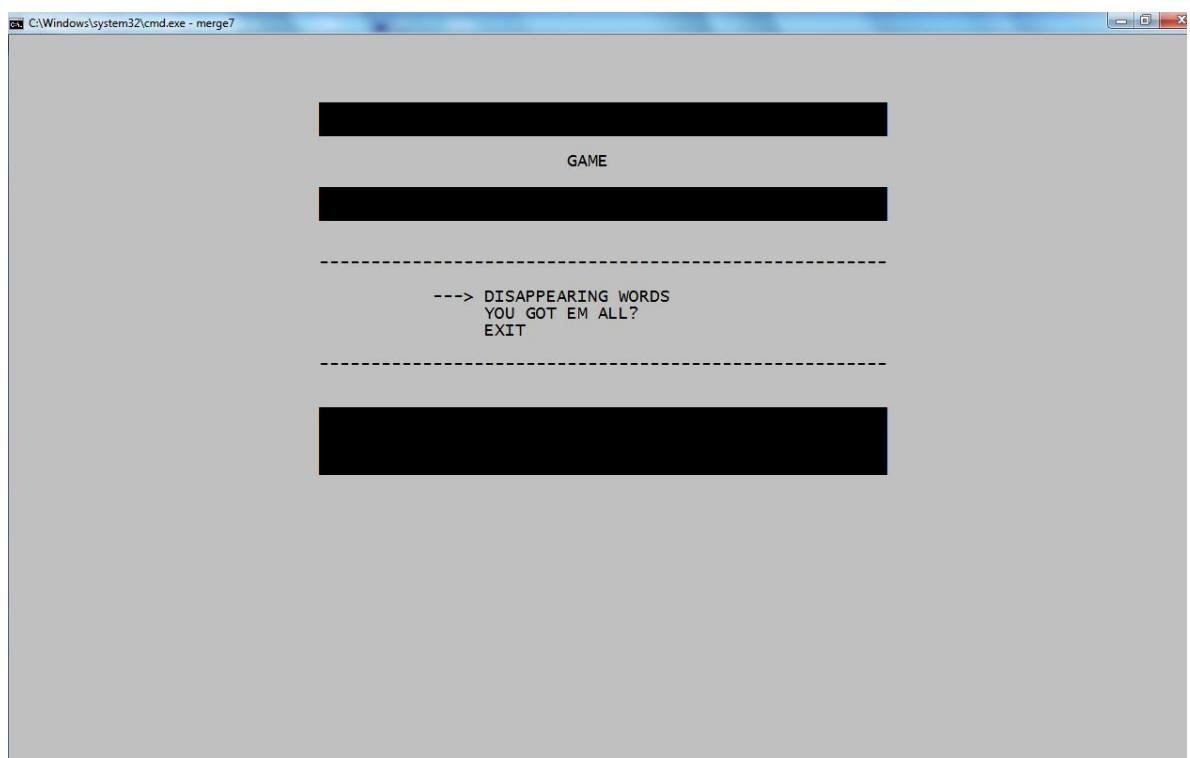
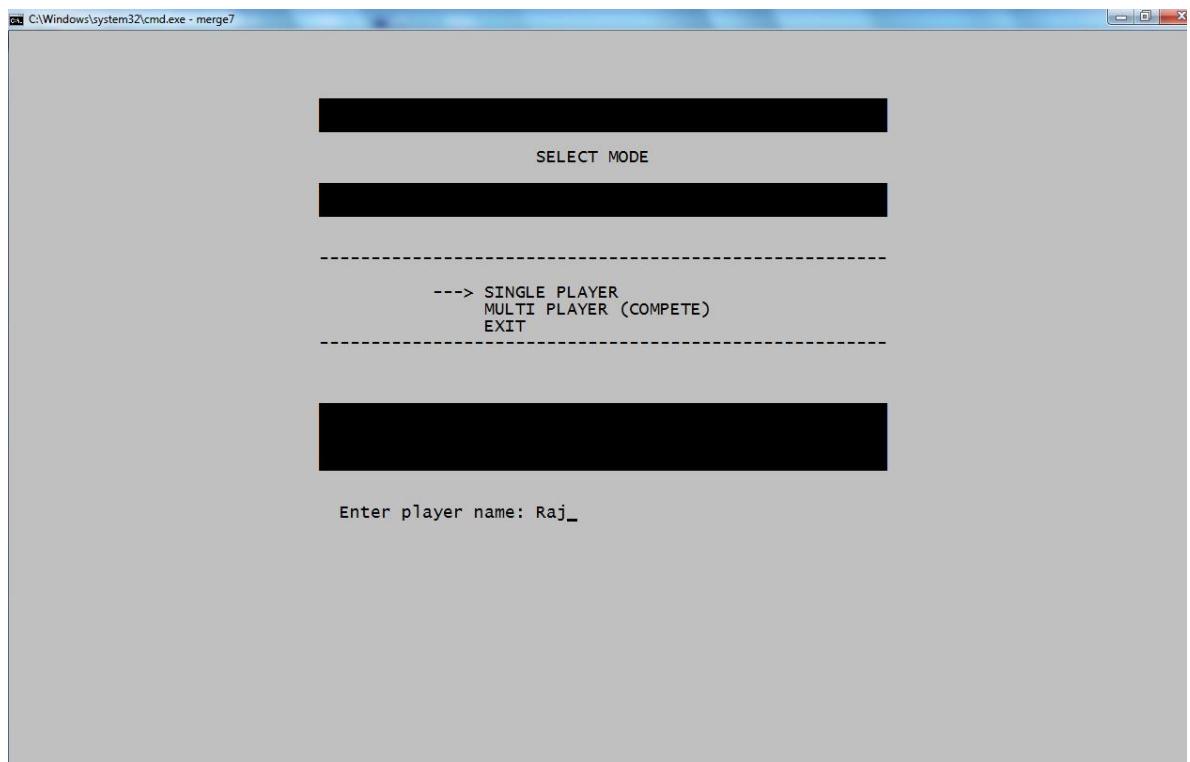
```
C:\Windows\system32\cmd.exe - merge7
[REDACTED]
TYPING TUTOR
[REDACTED]

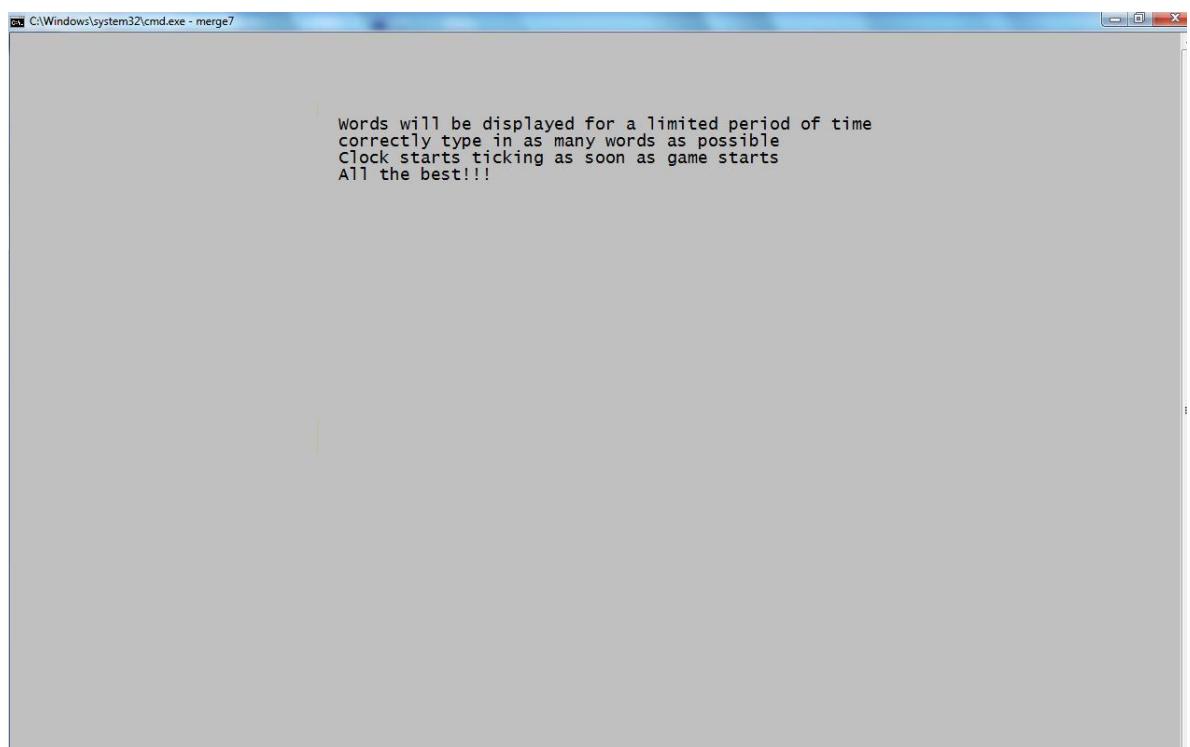
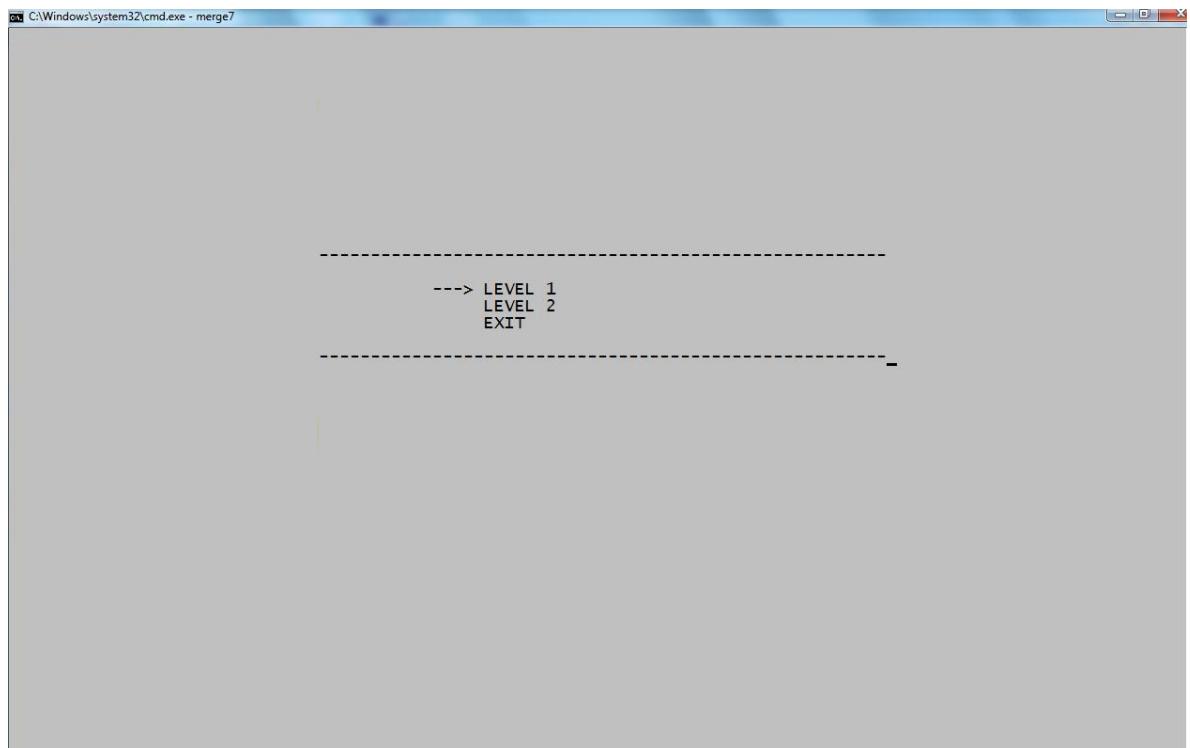
-----
TYPING
READING & COMPREHENSION
LEADER BOARD
--> EXIT
[REDACTED]

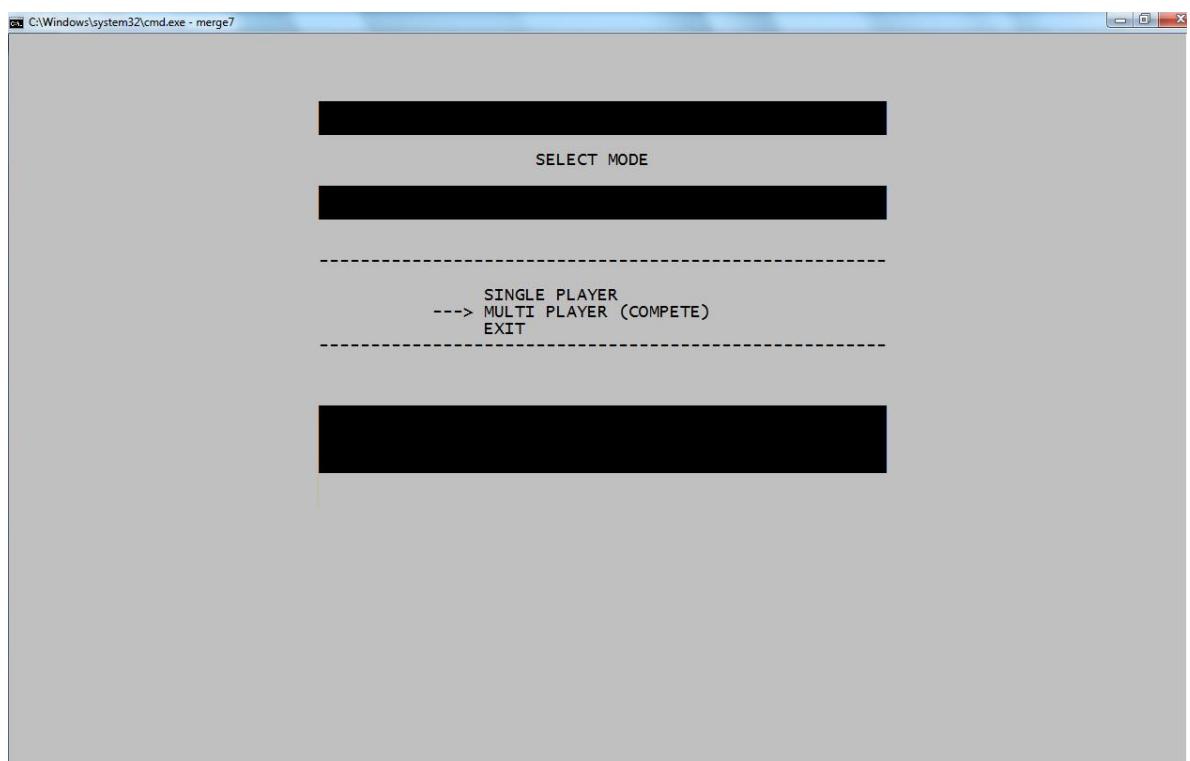
You want to log out?(y/n)
```

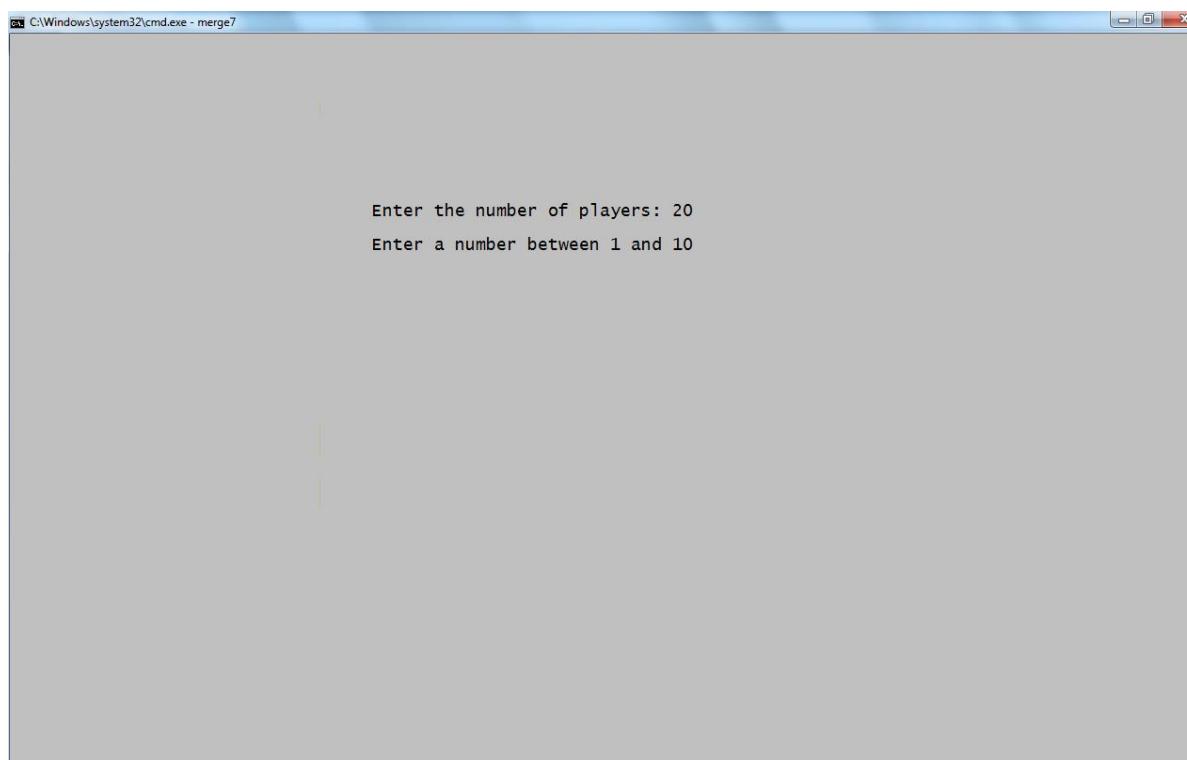
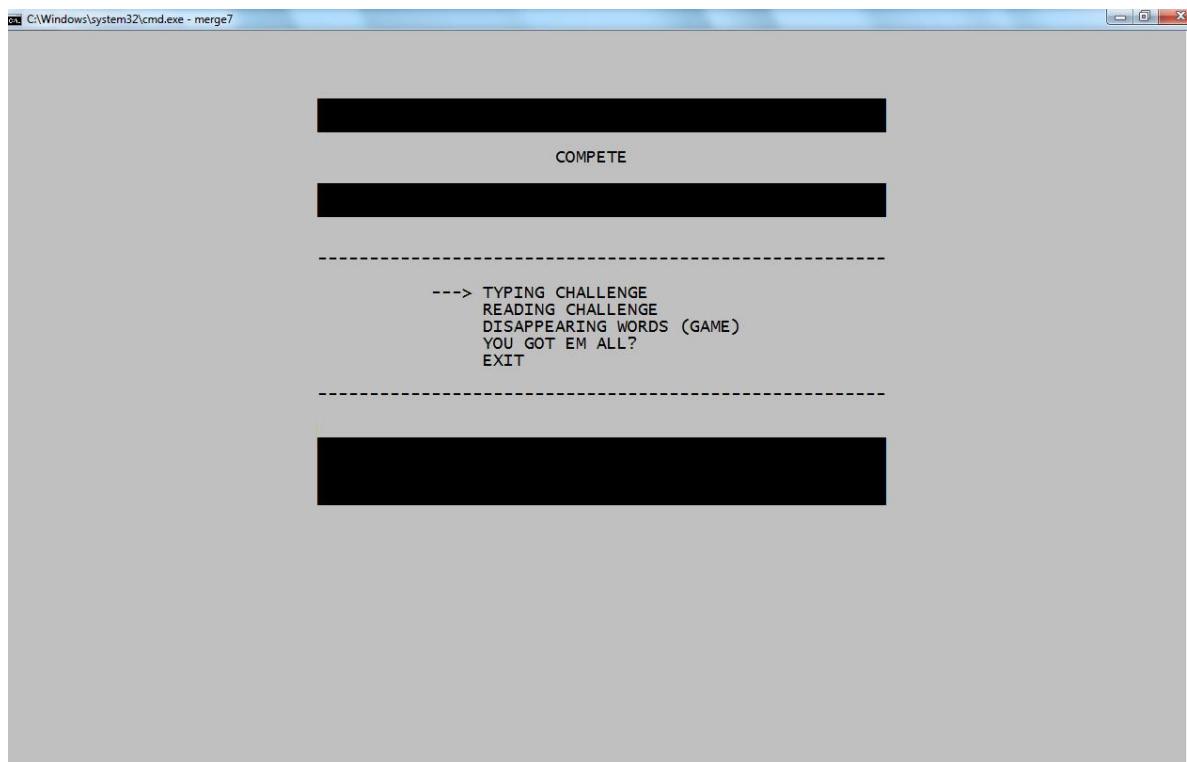
4.4 Competitor Test Cases











```
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 1 st Semester	