

TYPING TUTOR MINI PROJECT

NAME: NEMANI SRIVATHSA

REG NO: RA2111002010025

DEPARTMENT: MECHANICAL

SUBMITTED TO

DR. R. RAJKUMAR

DSBS

SCHOOL OF COMPUTING

SRMIST

JANUARY 2022

**ABSRACT** :

Typing Tutor is a Project developed entirely in C programming language using Graphics Library.

**Functions Used:**

The major user defined functions used in the source code of this project are:

* **void loadFiles(void)**: This function has been used to load .DAT files from the computer which has been stored with the .c file of project. The .DAT files contain the statistics, name of some users etc. of Typing Tutor Project in C. You can change them by editing the .DAT files and the source code of the project.
* **int mainmenu(void)**: This user defined function has been used to print the main menu of the project.
* **int listUser(void)**: There are certain predefined user in the project. This function lists name of them which are stored in ulist.DAT file.
* **int userSelectMenu(void)**: It is for selecting the name of users which are already defined in the source code.
* **void createUser(void)**: If you don’t like the provided name of user, you can create a new user name too. This task is done using this user defined function.
* **int listLesson(void)**: In Typing Tutor Project in C, this function has been used to list the name of lessons for testing the typing speed or learning the typing.
* **void beginSession()**: This function starts the session or lesson which has been selected by the user.

void viewRecords(void), void about(void), void sortSession(session list[],int n), void addRecord(session cstr, void viewStat(void), void DrawMenu(int opt,char \*mi[],int n), void box(int x,int y,int width,int height,int fc,int bc) are some other user defined function in Typing Tutor Project in C. In order to make the source code user friendly, the name of the function simply indicates its use in the project.

**Features:**

* This project provides certain name as user by default and there a facility to create a new user too. You can enter your name as user.
* In the project, you can view the total statistics.
* The record option in the project is for displaying the data of user which has already used the application. you can view typing speed, accuracy etc.
* If you want to terminate the program without any use, you can select exit option in the main menu.

**PROGRAM :**

**#include <stdio.h>**

**#include <conio.h>**

**#include <time.h>**

**#include <string.h>**

**#include <stdlib.h>**

**#include "tt.h"**

**#include "welcome.c"**

**FILE \*fulist; /\*user list file\*/**

**FILE \*fuser; /\*current user data file\*/**

**int nusers;**

**int SELECTING =1;**

**FILE \*fllist; /\*lesson list file\*/**

**FILE \*flesson; /\*current lesson data file\*/**

**/\*define current user and current lesson also\*/**

**user cu; /\*current user\*/**

**session cs; /\*current session\*/**

**lesson cl; /\*current lesson\*/**

**char \*day[7]={"Sunday","Monday","Tuesday","Wednesday","Thursday","Friday","Saturday"};**

**char \*month[12]={"January","February","March","April","May","June","July","August","September","October","November","December"};**

**char \*text;**

**void main(void)**

**{**

**int r1; /\*response 1, main nenu\*/**

**int r2; /\*response 2, user select\*/**

**loadFiles(); /\*initialize\*/**

**WS2(); /\*welcome screen\*/**

**while(RUNNING)**

**{**

**r1=mainmenu();**

**SELECTING=1;**

**switch(r1)**

**{**

**case SELECTUSER:/\*select user\*/**

**while(SELECTING)**

**{**

**r2 = userSelectMenu();**

**switch(r2)**

**{**

**case USERLIST:**

**if(listUser()== -1)break;**

**if(listLesson()==-1)break;**

**beginSession();**

**break;**

**case NEWUSER:**

**createUser();**

**if(listLesson()==-1)break;**

**beginSession();**

**break;**

**case BACK:**

**SELECTING = 0;**

**fclose(fuser); /\*close it\*/**

**break;**

**}/\*end switch(r2)\*/**

**}/\*end while(SELECTING)\*/**

**break;**

**case STAT:/\*view statistics\*/**

**viewStat();**

**/\*CLRSCR**

**printf("Statistics is not complete...");**

**printf("\n\n\n\n...press any key...");**

**getch();\*/**

**break;**

**case VIEWRECORDS:**

**viewRecords();**

**break;**

**case ABOUT:**

**about();**

**break;**

**case QUIT:/\*exit\*/**

**/\*close all open files before exitting\*/**

**fclose(fulist); /\*close the previous TEST open\*/**

**free(text);**

**return; /\*use exit(0); later\*/**

**}/\*end switch(r1)\*/**

**}/\*end while\*/**

**}/\*end main\*/**

**/\*==========================================================\*/**

**void loadFiles(void)**

**{**

**FILE \*fstat;**

**int n=3; lesson l;**

**if((fulist = fopen("userlist.dat","rb+")) == NULL) /\*if file is not created or is missing\*/**

**{**

**printf("\n\nFile does not exist..\n Creating new...");**

**fulist = fopen("userlist.dat","wb"); /\*then create it\*/**

**nusers=0; /\*no users yet\*/**

**fwrite(&nusers,sizeof(int),1,fulist);**

**fclose(fulist);**

**fulist = fopen("userlist.dat","rb+");**

**}**

**else /\*userlist.dat exists,\*/**

**{**

**fread(&nusers,sizeof(int),1,fulist); /\*read\*/**

**}**

**if( (fstat=fopen("stat.dat","rb")) == NULL )**

**{**

**fstat=fopen("stat.dat","wb");**

**fclose(fstat);**

**}**

**else fclose(fstat);**

**fllist=fopen("lessonlist.dat","wb");**

**fwrite(&n,sizeof(int),1,fllist);**

**strcpy(l.title,"atoz");**

**l.length=448;**

**fwrite(&l,sizeof(lesson),1,fllist);**

**strcpy(l.title,"hooks");**

**l.length=1757;**

**fwrite(&l,sizeof(lesson),1,fllist);**

**strcpy(l.title,"radio");**

**l.length=1717;**

**fwrite(&l,sizeof(lesson),1,fllist);**

**fclose(fllist);**

**CLRSCR**

**}**

**int mainmenu(void)**

**{**

**char \*menuitem[5+1]={"\n\n\r (1) Select User",**

**"\n\n\r (2) Statistics",**

**"\n\n\r (3) View User Records",**

**"\n\n\r (4) About",**

**"\n\n\r (5) Quit",**

**" \*\*\*MAIN MENU\*\*\*"};**

**int c; int extended; int option=1;**

**CLRSCR**

**textcolor(LIGHTGREEN);**

**cprintf("\n\r%26s%c T Y P I N G T E S T E R %c","",16,17);**

**textcolor(LIGHTMAGENTA);**

**cprintf("\n\n\r \xaf Press the arrow keys: \x18 or \x19 to navigate the options.\**

**\n\r \xaf Press <ENTER> key to accept an option.\**

**\n\r \xaf [OR] Press the number keys to select your choice");**

**\_setcursortype(\_NOCURSOR);**

**DrawMenu(1,menuitem,5); /\*default select\*/**

**while((c=getch())!='\r')/\*while return key not pressed\*/**

**{**

**if(!c) /\*if extended key\*/**

**{**

**extended=getch();**

**switch(extended)**

**{**

**case UP:**

**option--;**

**if(option<1)option=5;**

**break;**

**case DOWN:**

**option++;**

**if(option>5)option=1;**

**break;**

**}**

**}**

**else**

**{**

**option=c-48;**

**DrawMenu(option,menuitem,5);**

**break;/\*break while loop\*/**

**}**

**DrawMenu(option,menuitem,5);**

**}**

**\_setcursortype(\_NORMALCURSOR);**

**return option;**

**}**

**int userSelectMenu(void)**

**{**

**char \*menuitem[3+1]={"\n\n\r (1)Select from list",**

**"\n\n\r (2)Add new user",**

**"\n\n\r (3)Back",**

**"\*\*\*USER SELECTION MENU\*\*\*"};**

**int c; int extended; int option=1;**

**CLRSCR**

**textcolor(LIGHTGREEN);**

**textcolor(LIGHTGREEN);**

**cprintf("\n\r%26s%c T Y P I N G T E S T E R %c","",16,17);**

**textcolor(LIGHTMAGENTA);**

**cprintf("\n\n\r \xaf Press the arrow keys: \x18 or \x19 to navigate the options.\**

**\n\r \xaf Press <ENTER> key to accept an option.\**

**\n\r \xaf [OR] Press the number keys to select your choice");**

**\_setcursortype(\_NOCURSOR);**

**DrawMenu(1,menuitem,3); /\*default select\*/**

**while((c=getch())!='\r')/\*while return key not pressed\*/**

**{**

**if(!c) /\*if extended key\*/**

**{**

**extended=getch();**

**switch(extended)**

**{**

**case UP:**

**option--;**

**if(option<1)option=3;**

**break;**

**case DOWN:**

**option++;**

**if(option>3)option=1;**

**break;**

**}**

**}**

**else**

**{**

**option=c-48;**

**DrawMenu(option,menuitem,3);**

**break;/\*break while loop\*/**

**}**

**DrawMenu(option,menuitem,3);**

**}**

**\_setcursortype(\_NORMALCURSOR);**

**return option;**

**}**

**int listUser(void)**

**{/\* here the n user name list is displayed from the file ulist.dat**

**if a valid user has been selected, current user is set.\*/**

**int i;**

**char userfilename[N+3+1]; /\*including a period & extension\*/**

**CLRSCR**

**/\*open the user list file and list them all\*/**

**if(nusers==0)**

**{**

**printf("\n\nNo users exist...");**

**printf("\npress any key to continue...");**

**getch();**

**return -1;**

**}**

**printf("\nno. of users: %d",nusers);**

**//rewind(fulist); /\*Sujan : no need\*/**

**fseek(fulist,sizeof(int),SEEK\_SET);**

**for(i=1;i<=nusers;i++) /\*list them\*/**

**{**

**fread(&cu,sizeof(user),1,fulist); /\*read\*/**

**printf("\n(%d) %.8s",i,cu.name);**

**// fseek(fulist,sizeof(user),SEEK\_SET);**

**}**

**/\*now allow the use to select correct user name from the list\*/**

**printf("\nSelect a user (please type the name): ");**

**scanf(" %s",cu.name);**

**sprintf(userfilename,"%s.dat",cu.name);**

**if( (fuser = fopen(userfilename,"r+b")) == NULL ) /\*opened for the beginSession() to write session details\*/**

**{**

**printf("\n\nInvalid filename...");**

**printf("\npress any key to continue...");**

**getch();**

**return -1;**

**}**

**/\*NOTE! if the file cannot be created, the typed username must be incorrect\*/**

**printf("\nSuccess, press any key to continue...");**

**getch();**

**return 0;**

**}**

**void createUser(void)**

**{**

**char userfilename[N+3+1]; /\*including a period & extension\*/**

**/\* char \*dummy;\*/**

**CLRSCR**

**fseek(fulist,0L,SEEK\_SET);**

**nusers++;**

**fwrite(&nusers,sizeof(int),1,fulist); /\*update no of users\*/**

**printf("\nenter your name (8 characters Max): ");**

**scanf(" %8s",cu.name);**

**/\*if a valid name has been entered update n in the userlist.dat**

**and create separate file for new user\*/**

**/\*move pointer at the end\*/**

**fseek(fulist,0L,SEEK\_END);**

**fwrite(&cu,sizeof(user),1,fulist); /\*write a new user entry at the end\*/**

**sprintf(userfilename,"%s.dat",cu.name);**

**fuser = fopen(userfilename,"wb"); /\*create file\*/**

**fclose(fuser);**

**fuser = fopen(userfilename,"r+b");**

**/\*if valid name\*/**

**/\* 1 create a file for the user**

**2 update no of user in the ulist.dat and add the new name at the end**

**\*/**

**printf("\n\n\n\nUser file created\n press any key to continue...");**

**getch();**

**}**

**int listLesson(void)**

**{**

**/\*here the n lesson name is displayed from the file llist.dat**

**>> list.dat is assumed to be already opened**

**if a valid lesson has been selected current lesson is set to it.**

**\*/**

**int n,option; char filename[N+3+1]; /\*including a period & extension\*/**

**lesson l1,l2,l3; int c,i=0;**

**CLRSCR**

**/\*open the lesson list file and list them all\*/**

**fllist=fopen("lessonlist.dat","rb");**

**fread(&n,sizeof(int),1,fllist);**

**printf("\nnumber of lesson: %d",n);/\*3 lessons only\*/**

**fread(&l1,sizeof(lesson),1,fllist); /\*read\*/**

**printf("\n(1) %.8s %ld characters",l1.title,l1.length);**

**fread(&l2,sizeof(lesson),1,fllist); /\*read\*/**

**printf("\n(2) %.8s %ld characters",l2.title,l2.length);**

**fread(&l3,sizeof(lesson),1,fllist); /\*read\*/**

**printf("\n(3) %.8s %ld characters",l3.title,l3.length);**

**fclose(fllist);**

**/\*now allow the use to select correct user name from the list\*/**

**printf("\nSelect your lesson (1,2,3): "); scanf("%d",&option);**

**if(option==1)cl=l1;**

**else if(option==2)cl=l2;**

**else if(option==3)cl=l3;**

**else cl=l1;**

**sprintf(filename,"%s.txt",cl.title);**

**if( (flesson = fopen(filename,"r")) == NULL)**

**{**

**printf("\n\aLesson file not found!!!");**

**printf("\npress any key to continue...");**

**getch();**

**return -1;**

**}**

**if( (text=malloc(sizeof(char)\*cl.length)) == NULL )**

**{printf("\n\n\aOUT OF MEMORY!!!");getch();}**

**while( (c=getc(flesson)) != EOF)**

**{ text[i]=c; i++; }**

**fclose(flesson);**

**printf("\a");**

**return 0;**

**}**

**/\*-------------------------------------------\*/**

**void beginSession()**

**{**

**time\_t t1,t2; struct tm \*area;**

**int txtlen,mistakes,x1,x2,y1,y2,npage,c,currentpage,i,j,nwords;**

**int page=PAGE; float accuracy; int wpm,nchar;**

**CLRSCR**

**\_setcursortype(\_SOLIDCURSOR);**

**currentpage=1; mistakes=0; nwords=0; nchar=0;**

**txtlen=cl.length;**

**npage=txtlen/(PAGE)+1;**

**x1=(SCRWIDTH-PARAWIDTH)/2;**

**x2=x1;**

**y1=(SCRHEIGHT-((NLINES+2)\*2))/2;**

**y2=y1+NLINES+4;**

**i=0; /\*count correct characters\*/**

**j=0; /\*character index in the given part of text\*/**

**box(x1,y1,PARAWIDTH,NLINES,YELLOW,BLACK);**

**LoadPage(&page,txtlen,currentpage,i);**

**window(1,1,80,3);**

**textbackground(WHITE);**

**textcolor(MAGENTA);**

**clrscr();**

**cprintf("page %d of %d",currentpage,npage); /\*display also title username...\*/**

**cprintf("\n\rCurrent User : %10s",cu.name);**

**cprintf("\n\rCurrent Lesson : %10s",cl.title);**

**box(x2,y2,PARAWIDTH,NLINES,11,BLACK);**

**t1=time(NULL);**

**while(i<txtlen)**

**{**

**while(j<page)**

**{**

**c=getch();**

**if(c==text[i])**

**{**

**putch(c);**

**i++;**

**j++;**

**nchar++;**

**if(c==32 || c=='\t' || c== '\r')nwords++;**

**}**

**else if(c==ESC)**

**{ i=txtlen; break; }**

**else**

**{ mistakes++; nchar++; sound(1000);delay(50);nosound(); }**

**}**

**if(c==ESC)break;**

**/\*1 page completed\*/**

**j=0;**

**currentpage++;**

**if(currentpage>npage)break;**

**box(x1,y1,PARAWIDTH,NLINES,YELLOW,BLACK);**

**LoadPage(&page,txtlen,currentpage,i);**

**window(1,1,80,3);**

**textbackground(WHITE);**

**textcolor(MAGENTA);**

**clrscr();**

**cprintf("page %d of %d",currentpage,npage); /\*display also title username...\*/**

**cprintf("\n\rCurrent User : %10s",cu.name);**

**cprintf("\n\rCurrent Lesson : %10s",cl.title);**

**box(x2,y2,PARAWIDTH,NLINES,11,BLACK);**

**gotoxy(1,1);**

**}**

**t2=time(NULL);**

**/\*after a session has ended display result\*/**

**if(difftime(t2,t1)!=0)**

**wpm = (nwords\*60)/difftime(t2,t1); /\*WARNING: time difference may be zero\*/**

**else wpm=0;**

**if(nchar!=0)**

**accuracy=(float)100\*(nchar-mistakes)/nchar;**

**else accuracy=0.0;**

**/\*and write session in the file that has been opened by listUser() in fuser\*/**

**cs.u=cu;**

**area= localtime(&t1); /\*take the start time\*/**

**cs.dtntm = \*area;**

**cs.lsn=cl;**

**cs.wpm = wpm;**

**cs.accuracy = accuracy;**

**CLRSCR**

**textcolor(LIGHTBLUE);**

**cprintf("\n\r [ R E S U L T ]");**

**textcolor(YELLOW);**

**cprintf("\n\rName: %s",cs.u.name);**

**cprintf("\n\rLesson title: %s",cs.lsn.title);**

**cprintf("\n\rDuration: %.2f seconds",difftime(t2,t1));**

**cprintf("\n\rGross Speed : %d WPM",wpm);**

**cprintf("\n\rAccuracy : %.0f%%",accuracy);**

**cprintf("\n\rScore: %.0f",accuracy\*wpm);**

**cprintf("\n\r%s,%s %2d, %d",day[cs.dtntm.tm\_wday],month[cs.dtntm.tm\_mon],cs.dtntm.tm\_mday,cs.dtntm.tm\_year+1900);**

**cprintf("\n\rstart time [hh:mm:ss] = [%02d:%02d:%02d]",cs.dtntm.tm\_hour,cs.dtntm.tm\_min,cs.dtntm.tm\_sec);**

**fseek(fuser,0L,SEEK\_END);**

**fwrite(&cs,sizeof(session),1,fuser); /\*write current seesion\*/**

**fclose(fuser);**

**\_setcursortype(\_NOCURSOR);**

**addRecord(cs);**

**getch();**

**}**

**/\*================================================================\*/**

**void LoadPage(int \*page,int txtlen,int currentpage,int i)**

**{**

**int k;**

**if((txtlen-i)<(PAGE)) /\*if less than a page left to type\*/**

**{**

**for(k= (currentpage-1)\*PAGE;k<txtlen;k++) /\*load only remaining text\*/**

**putch(text[k]);**

**\*page=txtlen-i;**

**}**

**else**

**for(k=(currentpage-1)\*PAGE;k<(PAGE\*currentpage);k++)**

**putch(text[k]);**

**}**

**void box(int x,int y,int width,int height,int fc,int bc)**

**{**

**int i;**

**/\*draw the box\*/**

**x-=1; y-=1;**

**width+=2; height+=3;**

**window(x,y,x+width,y+height);**

**//clrscr();**

**textbackground(bc);**

**textcolor(fc);**

**for(i=2;i<width;i++) /\*draw horizontal lines\*/**

**{**

**gotoxy(i,1);**

**putch(HB);**

**gotoxy(i,height);**

**putch(HB);**

**}**

**for(i=2;i<height;i++) /\*draw vertical lines\*/**

**{**

**gotoxy(1,i);**

**putch(VB);**

**gotoxy(width,i);**

**putch(VB);**

**}**

**/\*draw 4 corners\*/**

**gotoxy(width,1); putch(TRC);**

**gotoxy(1,height); putch(BLC);**

**gotoxy(1,1); putch(TLC);**

**gotoxy(width,height); putch(BRC);**

**/\*restore values\*/**

**x+=1; y+=1;**

**width-=2; height-=2;**

**/\*draw the text window\*/**

**window(x,y,x+width-1,y+height-1);**

**clrscr();**

**}**

**void DrawMenu(int opt,char \*mi[],int n)**

**{**

**int i;**

**int bgcl[2]={BLACK, /\*deselection bg color\*/**

**LIGHTGRAY /\*selection bg color\*/**

**};**

**box(28,10,25,n\*2+2,YELLOW,BLACK);**

**cputs(mi[n]);**

**for(i=0;i<n;i++)**

**{**

**textbackground(bgcl[(opt==(i+1))]);**

**cprintf(mi[i]);**

**}**

**}**

**void viewStat(void)**

**{**

**FILE \*fstat; int i,n;**

**session tmp; char date[12];**

**CLRSCR**

**fstat=fopen("stat.dat","rb");**

**fseek(fstat,0L,SEEK\_END); /\*go to end\*/**

**n=ftell(fstat)/sizeof(session);**

**fseek(fstat,0L,SEEK\_SET); /\*go to beginning\*/**

**printf("\t\t<S T A T I S T I C S>");**

**printf("\n\nsn %10s%10s%7s%10s%8s%12s%10s","Name","Lesson","WPM","Accuracy","Score","Date","Time");**

**printf("\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");**

**for(i=0;i<n;i++)**

**{**

**fread(&tmp,sizeof(session),1,fstat); /\*read one from file\*/**

**sprintf(date,"%02d/%02d/%d",tmp.dtntm.tm\_mday,tmp.dtntm.tm\_mon+1,tmp.dtntm.tm\_year+1900);**

**printf("\n%2d %10s%10s%7d%9.1f%%%8.0f %12s [%02d:%02d:%02d]",i+1,**

**tmp.u.name,**

**tmp.lsn.title,**

**tmp.wpm,**

**tmp.accuracy,**

**tmp.accuracy\*(float)tmp.wpm,**

**date,**

**tmp.dtntm.tm\_hour,tmp.dtntm.tm\_min,tmp.dtntm.tm\_sec);**

**}**

**getch();**

**fclose(fstat);**

**}**

**void addRecord(session cstr)**

**/\*writes the current sessiondetail in the stat.dat file**

**if it is in the top 10 list, based on the score(score=accuracy\*wpm)\*/**

**{**

**int n;**

**FILE \*fstat; /\*file session and a tmp session variable\*/**

**session \*list;**

**fstat=fopen("stat.dat","r+b");**

**fseek(fstat,0L,SEEK\_END); /\*go to end\*/**

**n=ftell(fstat)/sizeof(session);**

**fseek(fstat,0L,SEEK\_SET); /\*go to beginning\*/**

**if(n==0)**

**{**

**fwrite(&cstr,sizeof(session),1,fstat);**

**fclose(fstat);**

**return;**

**}**

**list=calloc((n+1),sizeof(session));**

**if(list == NULL )**

**{ printf("\n\nERROR: Out Of Memory!!!\a"); getch(); fclose(fstat); return;}**

**fread(list,sizeof(session),n,fstat); /\*read all\*/**

**list[n]=cstr; /\*put current at end of list\*/**

**n++;**

**sortSession(list,n);**

**if(n>10)n=10;**

**fseek(fstat,0L,SEEK\_SET);**

**fwrite(list,sizeof(session),n,fstat);**

**fclose(fstat);**

**}**

**void sortSession(session list[],int n)**

**{**

**int j,p;**

**session tmp;**

**for(p=1;p<n;p++)**

**{**

**tmp=list[p];**

**for(j=p;j>0&&( (list[j-1].accuracy\*(float)(list[j-1].wpm))<(tmp.accuracy\*(float)(tmp.wpm)) );j--)**

**list[j] = list[j-1];**

**list[j] = tmp;**

**}**

**return;**

**}**

**void viewRecords(void)**

**{**

**session tmp; int n,i; char date[12];**

**if(listUser()== -1)return;**

**fseek(fuser,0L,SEEK\_END);**

**n=ftell(fuser)/sizeof(session);**

**fseek(fuser,0L,SEEK\_SET);**

**CLRSCR**

**printf("\t\t<U S E R R E C O R D>");**

**printf("\nUser name : %10s",cu.name);**

**printf("\n\nsn %10s%7s%10s%8s%12s%10s","Lesson","WPM","Accuracy","Score","Date","Time");**

**printf("\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");**

**for(i=1;i<=n;i++)**

**{**

**fread(&tmp,sizeof(session),1,fuser); /\*read one from file\*/**

**sprintf(date,"%02d/%02d/%d",tmp.dtntm.tm\_mday,tmp.dtntm.tm\_mon+1,tmp.dtntm.tm\_year+1900);**

**printf("\n%2d %10s%7d%9.1f%%%8.0f %12s [%02d:%02d:%02d]",i,**

**tmp.lsn.title,**

**tmp.wpm,**

**tmp.accuracy,**

**tmp.accuracy\*(float)tmp.wpm,**

**date,**

**tmp.dtntm.tm\_hour,tmp.dtntm.tm\_min,tmp.dtntm.tm\_sec);**

**if((i%18)==0 )**

**{**

**printf("\nPress any key to continue....");**

**getch();**

**}**

**}**

**fclose(fuser);**

**getch();**

**}**

**void about(void)**

**{**

**CLRSCR**

**window(15,5,80,25);**

**textcolor(LIGHTRED);**

**cprintf("\n A B O U T T Y P I N G T E S T E R");**

**textcolor(LIGHTCYAN);**

**getch();**

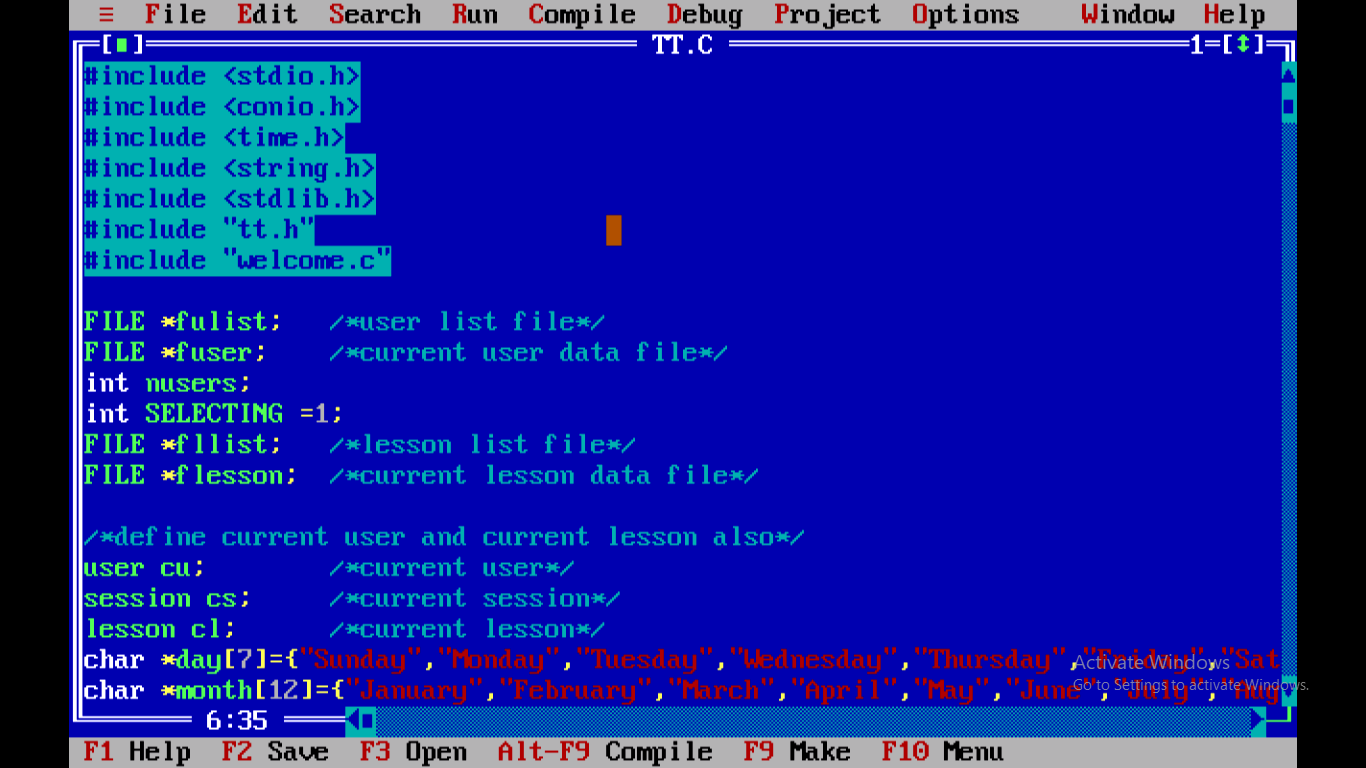
**}**

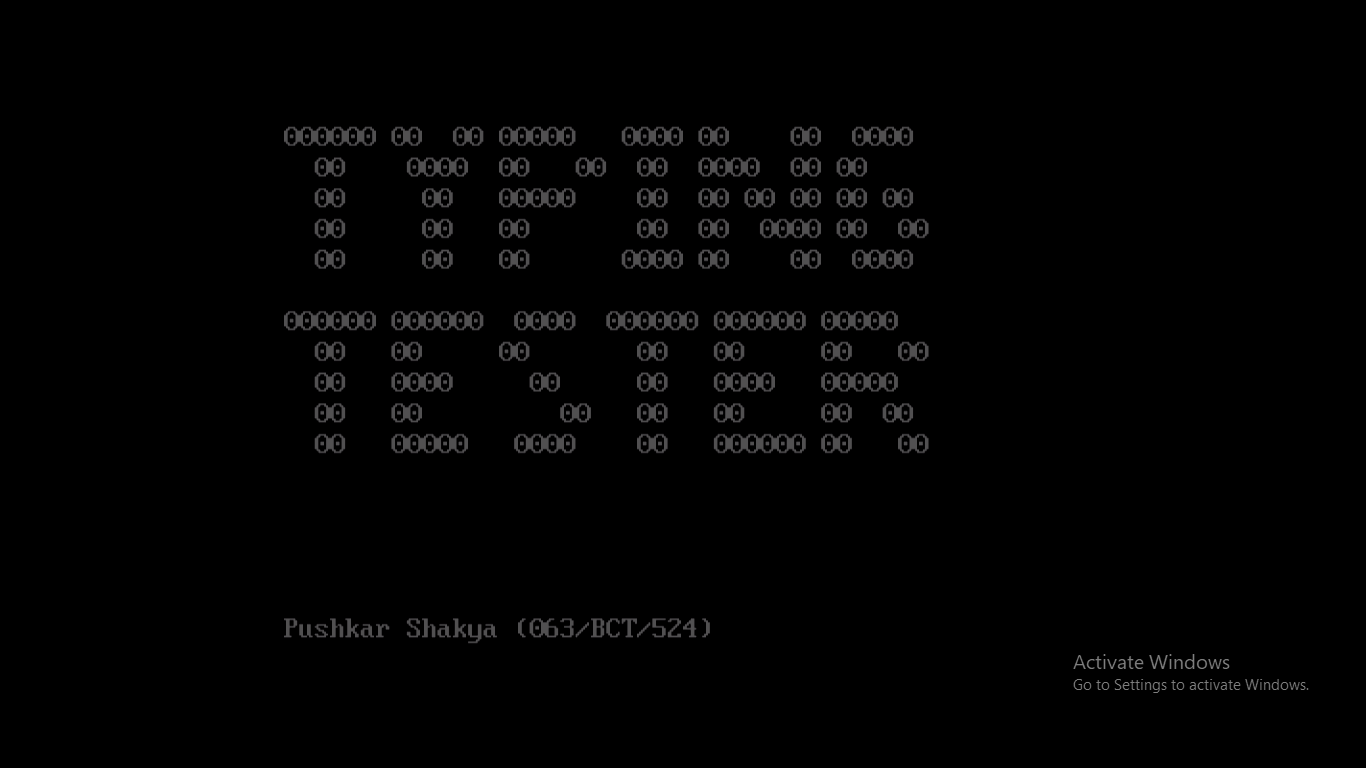
**RESULTS: TYPING**

**TESTER**

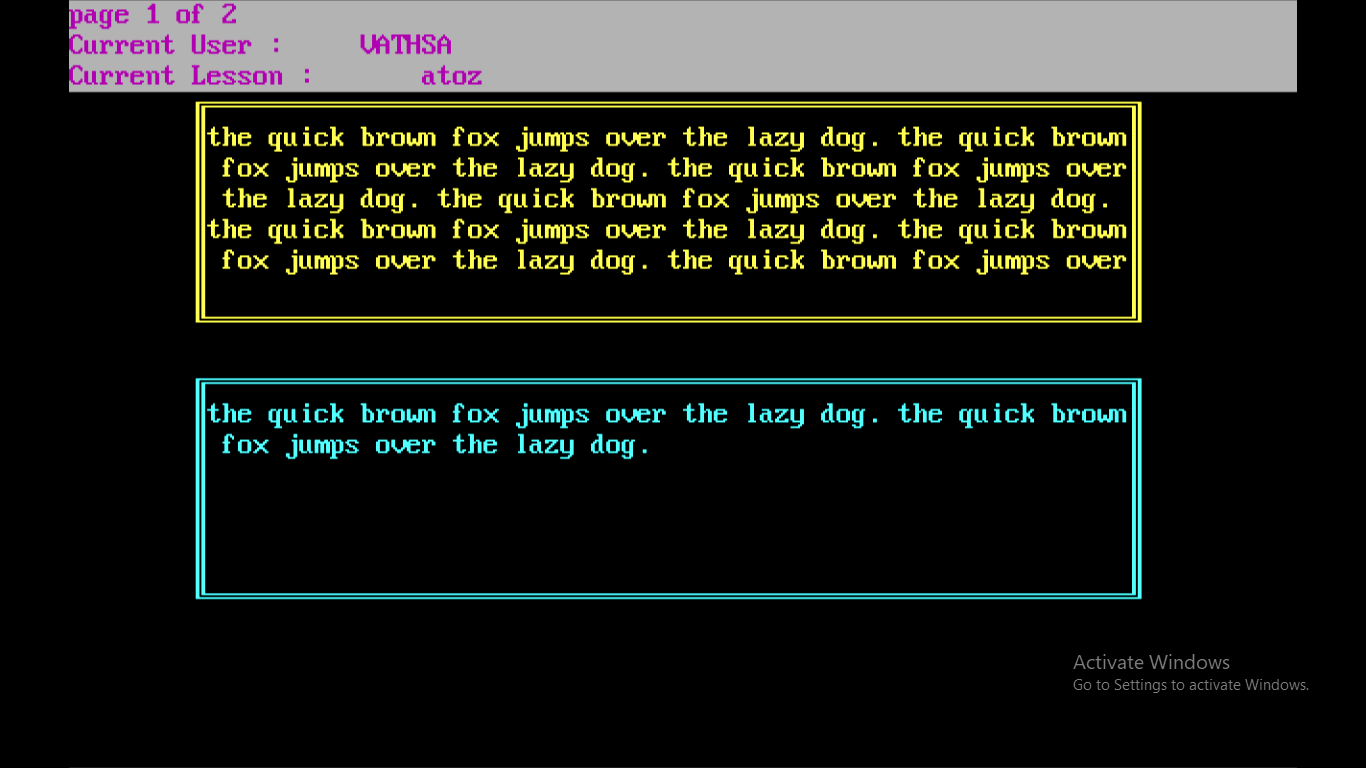
**Pushkar Shakya (063/BCT/524)**

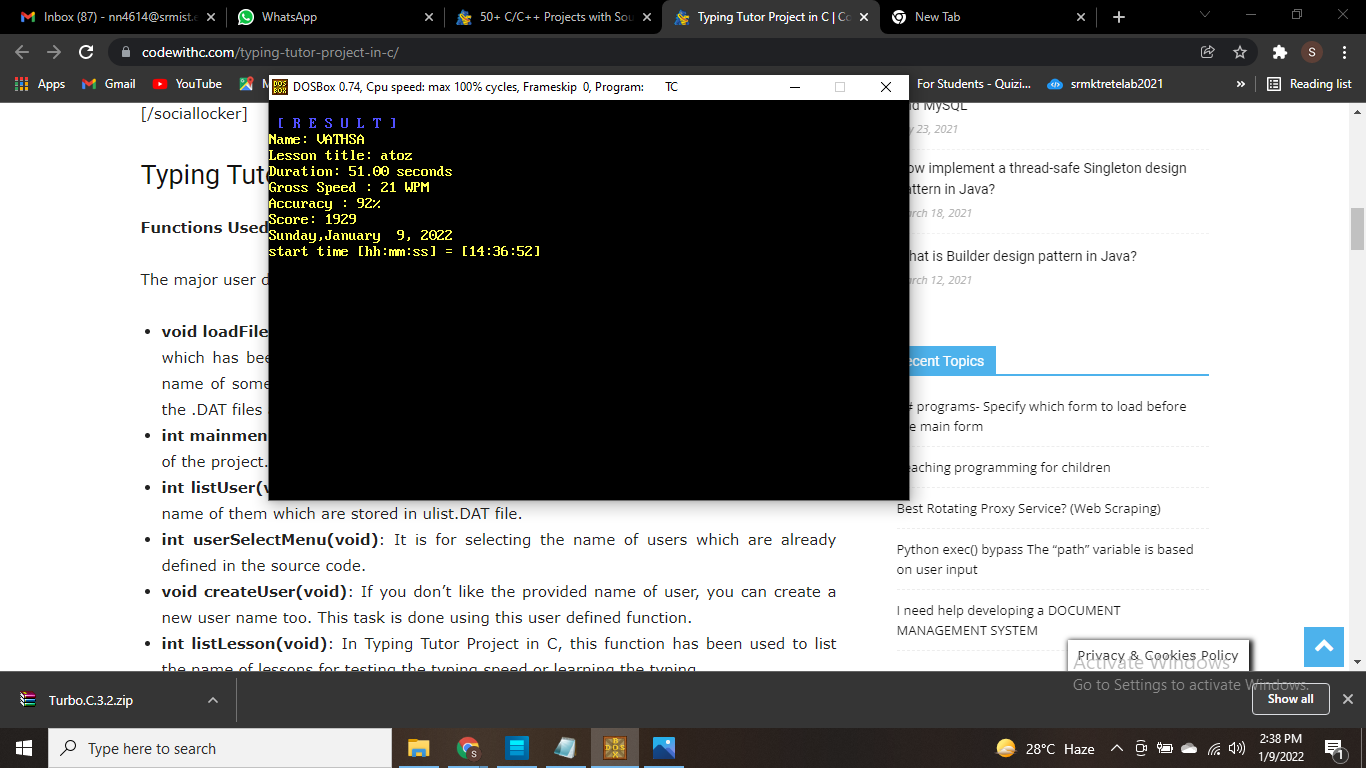
**SCREENSHOTS:**

****

****







**DECLARATION:**

**I would like to thank my SIR “ R.RAJKUMAR” for giving this wonderful opportunity.**

**It is an wonderful opportunity to show my skills.**

**I would like to thank CodewithC website for providing support to do this project.**

**RERERENCES :**

**<https://www.codewithc.com/typing-tutor-project-in-c/>**