**CALENDARING SOFTWARE**

*A*

*Mini Project Report*

*Submitted in partial fulfilment of the*

*Requirements for the award of the Degree of*

**BACHELOR OF ENGINEERING**

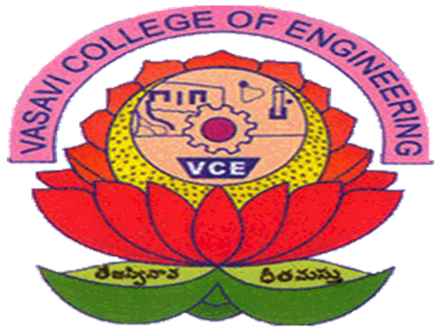
IN

**INFORMATION TECHNOLOGY**

By

**KANDI. NAVYA SHRUTHI 1602-19-737-143**

**BHAVANI SILASAGARAM 1602-19-737-173**

****

**Department of Information Technology**

**Vasavi College of Engineering (Autonomous)**

**(Affiliated to Osmania University)**

**Ibrahimbagh, Hyderabad-31**

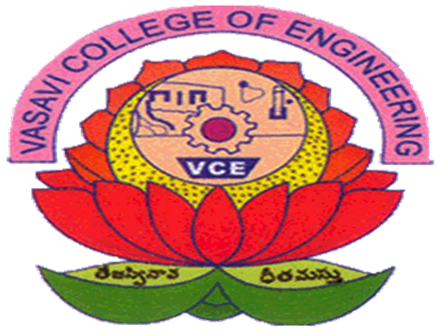
**2020**

**Vasavi College of Engineering (Autonomous)**

**(Affiliated to Osmania University)**

**Hyderabad-500 031**

**Department of Information Technology**

****

**DECLARATION BY THE CANDIDATE**

We, **K. NAVYA SHRUTHI** and **BHAVANI SILASAGARAM**, bearing hall ticket numbers, **1602-19-737-143** and **1602-19-737-173** hereby declare that the project report entitled “CALENDARING SOFTWARE” is submitted in partial fulfilment 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.

**K. NAVYA SHRUTHI**

**1602-19-737-143**

**BHAVANI SILASAGARAM**

**1602-19-737-173**

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

**ACKNOWLEDGEMENT**

We express our sincere indebtedness towards the guide Ms. Divya Lingineni, Information Technology, Vasavi college of engineering for her valuable guidance, suggestions and supervisions throughout the work. Without her kind patronage and guidance the project would not have taken shape. We would also like to express our gratitude and sincere regards for her kind approval of the project, time to time counselling and advice.

We would also like to thank our HOD who have asked us to take up a project in the third semester itself.

We have taken efforts in this project. However, it would have not been possible without the support of faculty and staff. We would like to extend my sincere thanks to all of them.

We would like to express our sincere thanks to GOOGLE which really helped us to sort some tough errors and taught us with some new functions.

Last but not the least, we would also like to thank our parents who have supported us and created a environment during this pandemic situation where we could work and complete the project successfully.

So, with gratitude we acknowledge all those who guided and encouraged us to complete the project.

**ABSTRACT**

In today’s world most of us are dependent on the calendar for various purposes. We have decided to make your work more easier and efficient with our calendaring software. It is a software that minimally provides users with an electronic version of a calendar. This application will be built on the console without using graphic properties; instead, it utilizes many other window properties to give the application a colourful look and feel. Our software would enable the users to add notes , remainders and it enables the user to prepare a to-do list for a particular day with the time slots assigned. The software also enables the user to check for overlapping events etc features.

**Future work**: We will add time blocking-Allows users to organise their days into chunks. Additionally the software may provide an appointment book , address book, and/or contact list.

**CONTENTS**

1. **Introduction…………………………………………. 6**
2. **Technology……………………………………………8**
   1. Software requirements………………………….8
   2. Hardware requirements…………………………8
3. **Proposed work………………………………………..9**
   1. **DESIGN**
      1. Use case diagram and descriptions for all the use cases……………………………………...9
   2. **IMPLEMENTATION**
      1. Module wise code for the entire project…11
      2. Specific code is highlighted in bold……..22
      3. Github links……………………………...34
   3. **TESTING……………………………………………….34**
4. **Results…………………………………………………35**
5. **Additional knowledge gained………………………...57**
6. **Conclusion…………………………………………….57**
7. **Future work…………………………………………..58**
8. **References…………………………………………….58**

**1.0 INTRODUCTION**

We realized the world is becoming fast paced and the need has come to organise and store our data day wise in an efficient manner which can be retrieved easily. It is a tough task to remember all these in this digital era.

In order to solve this problem, we have come up with a software which would minimally provide users to navigate throughout the schedule and organise the chunks both day wise and time wise. Digitalization has made our works and things much effortless where everyone can easily remember everything even on their busy days. The software is a local application designed for individual use i.e., one user for one specific file location. Our application is user friendly and easily navigable. This software allows the user to create a username and password which would keep his/her data safe. This is a modern alternative to paper planners. Aside from providing users with an electronic version of a calendar, the software also helps the user with time management. A person can use the software to manage his/her time effectively, so they can allocate the right time for the right activity.

The software saves the data in the binary files. This facilitates data hiding and also reduces loss of information. The entire information will be maintained in the files and authorized user can retrieve the necessary information which can be easily accessible from the file.

Our software helps users to set a day-to-day plan in the calendar by making use of certain attributes like notes, to-do list, reminders etc.

FEATURES

Calendaring software will contain the following features

* Calendar - a calendar showing dates and days of month and year highlighting the occupied dates.
* Availability checking – this feature checks whether the date is preoccupied or not.
* Notes – this feature allows the user to add any notes
* To-do list – this feature allows the user to make the list of all the forthcoming activities with time, date, month and year.
* Reminders – this feature reminds the user with a beep sound if he/she has any reminders on that particular day.
* Time – this feature allows the user to view the current time.
* The user can also find the day for a given date

**2.0 TECHNOLOGY**

All computer software needs certain hardware components and also other software resources to be present, in order for computer to be used efficiently. These pre requirements are known as System Requirements

SOFTWARE REQUIREMENTS:

* Operating System: Windows 10
* C compiler: GCC Compiler
* Editor: Vim editor and VS code

HARDWARE REQUIREMENTS:

* Processor: Intel Core i5 and above
* Memory: 4 GB RAM and above

1. **PROPOSED WORK**

**3.1 DESIGN**

3.1.1 USE CASE DIAGRAM

**Diagram

Description automatically generated**

Descriptions

* Use Case-1: LOGIN

The login feature allows the user to login into the software with the correct credentials. On entering the wrong password or username, system allows the user to enter again.

* Use Case-2: REGISTER

The register feature allows the user to register in the software with his desired username and password. If the username already exists, system will not take the entered username and asks the user to enter again.

* Use Case-3: NOTES

Notes is a feature available in our software which allows the user to enter the notes.

* Use Case-4: TO-DO LIST

User can ask the system to show the to-do list or can enter an entry to the to-do list

* Use Case-5: DAY

User is asked to enter date, month and year. System checks whether it is leap year or not and prints the day of the week name for the respective date entered.

* Use Case-6: DISPLAYING MONTH AND DAY

User enters date, month and year. System displays the month with all the booked days.

* Use Case-7: REMINDERS

User can enter the reminders he want to remind. When the program compiles and runs reminders are displayed on the screen.

* Use Case-8: OVERLAPPING EVENTS

Shows all overlapping events with names and times.

* 1. **Implementation**

//MINIPROJECT - CALENDARING SOFTWARE

//NAVYA - 1602-19-737-143

//BHAVANI - 1602-19-737-173

#include <stdio.h>

#include <stdlib.h>

#include<time.h>

#include<string.h>

#include<stdbool.h>

#include<Windows.h>

#include<conio.h>

#define LEN 256

#define MAXCHAR 1000

//Functions

void display\_rem(void);//displays all the reminders

void displaytime(void);//displays the time

void writenotes(void);// adds new notes

void readnotes(void);//reads all the notes

void login(void);//logs in

void reg(void);//registers

int isLeapYear( int year ); //to check if given year is a leap year or not

int leapYears( int year ); //to find number of leap years passed

int todayOf( int y, int m, int d);

long days( int y, int m, int d); //total days elapsed

void calendar(int y, int m); //display the calendar

int getDayNumber(int d,int m,int y);//helper functions

void sound();

void sound()

{

FILE \*fp;

char c[1000];

fp=fopen("rem\_test.txt","r");

char ch;

while(1)

{

if(fgets(c,100,fp)==NULL)

Beep(500,750);

break;

}

fclose(fp);

}

char \*getName(int day);//gets the day

void add\_rem(void);//adds a reminder

static int i=0;

//structures and flush

struct web

{

//structure which stores the web registration details

char name[30],pass[30],repass[30];

int phonenumber;

}w[99];

int n;

void flush()

{

int c;

while ((c = getchar()) != '\n' && c != EOF);

}

typedef struct {

//structure - day month and year and note corresponding to that date

int day;

int month;

int year;

char time[15];

char note[255];

} Note;

//main

int main(int argc, char\* argv[]){

int year,month, day;

char choice;

Note note;

FILE \*fp;

system("cls");

printf("\n\n\n\n\n\t\t\t\tWELCOME TO CALENDARING SOFTWARE,ITS NICE TO HAVE YOU HERE");

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

printf("\n\n\n\n\t\t\tPress Enter to proceed...!!");

if(getch()==13)

system("cls");

XY:

printf("\n\n\n\t\t\t1. LOGIN\t\t2. REGISTER");

printf("\n\n\n\t\t\t\tENTER YOUR CHOICE: ");

scanf("%d",&n);

switch(n)

{

case 1: system("cls");

login();

break;

case 2: system("cls");

reg();

break;

default:

printf("\n\n\t\t\t\tNO MATCH FOUND");

printf("\n\n\t\t\tPress Enter to re-Enter the choice");

if(getch()==13)

system("cls");

goto XY;

}

fp = fopen("note.txt", "r");

if (fp == NULL) {

fp = fopen("note.txt", "w");

}

fclose(fp);

while(1) {

sound();

display\_rem(); //shows te reminders at the top of the screen

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

printf("\n\n\n\n----------------------So, what do you want to do today?------------------------ \n\n\n\t\t\t1. Find the day of the week\n\n");

printf("\t\t\t2. Print calendar of a month (showing all the booked days)\n\n");

printf("\t\t\t3. Add An Entry to The To-Do-List\n\n");

printf("\t\t\t4. Read/Edit the notes\n\n");

printf("\t\t\t5. Show the current time\n\n");

printf("\t\t\t6. Add reminder\n\n");

printf("\t\t\t7. Exit\n\n");

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

printf("\t\t Enter your choice here : ");

scanf("\n%c", &choice);

switch(choice) {

case '1':

//Find the day of the week

system("cls");

printf("\n\nYou selected find the day, given the date.\n\n\nEnter the following credentials\n\n\nEnter the day, month and year: ");

scanf("%d %d %d", &day, &month, &year);

printf("\n\n\n^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^\n");

printf("\nThe day is : %s\n", getName(getDayNumber(day, month, year)));

//checks if its a leap year or not

if(isLeapYear(year)){

printf("\n Is it a leap year ? Yes, Its a leap year\n\n");

}

else{

printf("\nIs it a leap year ? No, Its not a leap year\n");

}

printf("\n^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^\n");

break;

case '2':

//view month with booked days

system("cls");

printf("\n===================================================\n");

printf("You have selected to view the month with booked days\n\nEnter the following credentials\n\n\n Enter the month and year: ");

scanf("%d %d", &month, &year);

printf("\n\nPlease enter 's' to see the list in the to-do-list\n\n Press any other key to continue\n");

calendar(year, month);

break;

case '3':

//Appending the to do list

system("cls");

printf("\n===================================================\n\n");

printf("You have picked to append the to-do-list \n\nEnter the following credentials\n\n\nEnter the time,day,month and year: ");

printf("\n===================================================\n\n");

scanf("%s %d %d %d",note.time, &note.day, &note.month, &note.year);

//storing the day month and year to append to in "note" derived from the structure note.

flush();

printf("\n\nEnter the Entry for this day : ");

fgets(note.note, 255, stdin);//stores the entered note into the note string which is part of the structure

fp = fopen("note.txt", "a+");

if (fp == NULL) {

printf("\nFile note.txt can not be opened\n");//if file cannot be opened

exit(1);

}

fwrite(&note, sizeof(Note), 1, fp);//sending all the contents of the "note"(struct point) into the file note.txt

printf("\n\n || To-Do list Appended sucessfully ||\n\n");

//to-do-list is appended and file pointer is closed

fclose(fp);

break;

case '4':

//Reading and adding notes

system("cls");

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

readnotes();//shows notes

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

int v;

printf("\nDo You Want to Add More notes ?\n1 or 0\n");

scanf("%d",&v);

if(v==1){

writenotes();//allows the user to add more notes if 1 is entered.

}

break;

case '5':

//Displays the current time

system("cls");

displaytime();

break;

case '6':

//Adding reminders

system("cls");

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

add\_rem();

break;

case '7':

//Exit

system("cls");

printf("\n\n\n\n\n\t\t\t\tHave a wonderful day! and visit again later");

printf("\n\n\t\t\t\t^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^\n\n\n\n");

exit(0);

break;

default:

printf("Not a valid option\n");

break;

}

}

return 0;

}

//all functions

int isLeapYear( int y ){

if(y%400==0){

return true;

}

else{

if(y%4==0&&y%100!=0){

return true;

}

else{

return false;

}

}

}

int leapYears( int y ){

int a = y/4;

int b = y/100;

int c = y/400;

return a-b+c;

// number of leap years elapsed

//if a number is divisible by 400 it is a leap year

//if a number is divisible by 4 and not by 100 its a leap year

}

int todayOf( int y, int m, int d) {

//total days in each month plus the days in all the months before it

//PREFIX SUM of the total days for each month

static int DayOfMonth[] = { -1,0,31,59,90,120,151,181,212,243,273,304,334};

//finds the 'x'th day from the starting of the year

//and adds 1 if if that year is a leap year AND february is PASSED

int addone = ((m>2 && isLeapYear(y))? 1 : 0);//Conditional operator

return DayOfMonth[m] + d + addone;

//if month >2 i.e if month > february and its a leap year we add one extra day.

}

long days( int y, int m, int d){

int lastYear;

lastYear = y - 1;

//finds the total number of days elapsed from day 0 year 0

int untillastyr = 365 \* lastYear ;

int totalleaps = leapYears(lastYear) ;

int uptilltoday = todayOf(y,m,d);

return untillastyr + totalleaps + uptilltoday;

}

char \*getName(int day){

//returns the name of the day

switch(day){

case 0 :

return(" Sunday ");

case 1 :

return(" Monday ");

case 2 :

return(" Tuesday ");

case 3 :

return(" Wednesday ");

case 4 :

return(" Thursday ");

case 5 :

return(" Friday ");

case 6 :

return(" Saturday ");

default:

return("Enter a VALID ARGUMENT");

}

printf("^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^");

}

int getDayNumber(int d, int m, int y){

//retuns the day number

static int t[] = {0, 3, 2, 5, 0, 3, 5, 1, 4, 6, 2, 4};//starting day name of each month

if(m<3){

y = y - (1);

}

else{

y = y;

}

return (y + y/4 - y/100 + y/400 + t[m-1] + d) % 7;

}

**void calendar(int y, int m){**

**//takes in the year and month**

**FILE \*fp;//takes a file pointer**

**Note\* notes, note;//and pointers to the structure**

**int len, j, hasNote = 0;**

**char choice;**

**//names of the months**

**const char \*NameOfMonth[] = { NULL/\*dummp\*/,**

**"January", "February", "March", "April", "May", "June","July", "August", "September", "October", "November", "December"};**

**//all the weeks that are being printed**

**char Week[] = "Su Mo Tu We Th Fr Sa";**

**//all the last dates of each month**

**int DayOfMonth[] = { -1,31,28,31,30,31,30,31,31,30,31,30,31 };//array of total days in each month**

**int weekOfTopDay;**

**int i,day;**

**weekOfTopDay = days(y, m, 1) % 7;//tells the week in which the first day of the month its located in.**

**fp = fopen("note.txt", "r");**

**if (fp == NULL) {**

**printf("Couldn't read notes\n");**

**}**

**len = 0;**

**while(fread(&note, sizeof(Note), 1, fp)) {//reads all the notes from the file and stores in "note".(point struct)**

**if (note.year == y && note.month == m) {//checking if the month and year is the same as the one we entered.**

**len++;**

**}**

**}**

**// now length contains the total number of notes with the month and year same as the entered month and year**

**rewind(fp);**

**j = 0;**

**//int notess[1000];//rev**

**notes = (Note\*) malloc (sizeof(Note) \* len); //allocating memory the size of the structure for notes that contains the day month year and note**

**int t=0;**

**int overlap[1000];//here**

**while(fread(&note, sizeof(Note), 1, fp)) {//while reading from the file we store the data**

**if (note.year == y && note.month == m) {//getting all the notes from the file which have the same day and store into**

**//getting ALL the events and storing the dates in the array**

**overlap[t] = note.day;//for overlapping event checking**

**notes[j] = note;//the "note" array that stores the 'day month year and note'**

**j++;**

**t++;**

**}**

**}**

**fclose(fp);**

**//edit --- CHECKING OVERLAPPING EVENTS**

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

**for(int ele=0;ele<t+1;ele++){**

**//debug ---> printf("%d th element in the array is - %d \n",ele,overlap[ele]);**

**for(int elee=ele+1;elee<t+1;elee++){**

**if(overlap[ele]==overlap[elee]&&overlap[ele]!=0&&overlap[elee]!=0){**

**printf("Overlapping event on %d\n",overlap[ele]);**

**//checks if the same event occurs more than one time**

**break;**

**}**

**}**

**}**

**//edit**

**if(isLeapYear(y))**

**DayOfMonth[2] = 29;**

**printf("\n %s %d\n%s\n", NameOfMonth[m], y, Week);//heading of the calendar**

**//Calendar gets printed**

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

**printf(" ");//empty spaces**

**for(i=weekOfTopDay,day=1;day <= DayOfMonth[m];i++,day++){**

**int contains\_a\_note = 0;**

**for (j = 0; j < len; j++) {**

**if (notes[j].day == day) {**

**//if the day that we are printing now is the same as one of the days in our note array then**

**//print that day with 2 bars beside it.**

**printf("|%2d| ",day);**

**contains\_a\_note = 1;//and if it already has a note we flag it**

**break;**

**}**

**}**

**if (contains\_a\_note == 0) {//so it doesnt print the same date twice**

**printf("%2d ",day);**

**}**

**if(i % 7 == 6)//and if the week shifts to the next one we change the line**

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

**}**

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

**scanf("\n%c", &choice);**

**//if the user asks to display the to do list for that month**

**if (choice == 's') {**

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

**printf("\n\t\t\t\tHere is the TO DO LIST for %d %d\n", m, y);**

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

**for (j = 0; j < len; j++) {**

**printf("%d %s: %s", notes[j].day,notes[j].time,notes[j].note);**

**//displaying all the notes and thier respective days from the array containg dates and notes.**

**}printf("\n~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~");**

**} else {**

**return;**

**}**

**}**

void add\_rem(){

FILE \* fpp;

int i;

char sentence[1000];

fpp = fopen ("remtest.txt","a+");

if (fpp == NULL) {

printf("Error!");

exit(1);

}

flush();

printf("\n\n...................................................\n");

printf("\t\tEnter a reminder for today:\n");

//entering the reminder into the file

fgets(sentence, sizeof(sentence), stdin);

fprintf (fpp, "%s\n",sentence);

printf("\n\n...................................................\n");

fclose(fpp);

return;

}

void display\_rem(){

FILE \*fpp;

char c[1000];

char\* filename = "remtest.txt";

char cc;

fpp = fopen(filename, "r");

if (fpp == NULL){

printf("Empty\n");

fpp = fopen(filename, "w");

fclose(fpp);

return;

}

printf("\n\n\n\t\t\tREMINDERS FOR TODAY - URGENT !! :");

printf("\n\t\t\t~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\n\n");

while(1)

{

if(fgets(c, 100, fpp) ==NULL)

break;

else

printf("%s", c);

}

fclose(fpp);

return;

}

void displaytime(){

char cur\_time[100];

time\_t t;

struct tm\* ptm;

t=time(NULL);

ptm=localtime(&t);

strftime(cur\_time,128,"%d-%b-%Y %H:%M:%S",ptm);

printf("......................................................................\n\n");

printf("\t\tCURRENT DATE AND TIME:%s\n\n",cur\_time);

printf("......................................................................\n\n");

return;

}

void writenotes()

{

FILE \* fpp;

int i;

char sentence[1000];

fpp = fopen ("notestest.txt","a+");

if (fpp == NULL) {

printf("Error!");

exit(1);

}

flush();

printf("\n\n...................................................\n");

printf("\t\tEnter a sentence:\n");

fgets(sentence, sizeof(sentence), stdin);

fprintf (fpp, "%s\n",sentence);

printf("\n\n...................................................\n");

fclose(fpp);

return;

}

void readnotes() {

FILE \*fpp;

char c[1000];

char\* filename = "notestest.txt";

char cc;

fpp = fopen(filename, "r");

if (fpp == NULL){

printf("Empty\n");

fpp = fopen(filename, "w");

fclose(fpp);

return;

}

printf("\n\t\t\tTHE NOTES ENTERED WERE :\n");

printf("\n\t\t\t^^^^^^^^^^^^^^^^^^^^^^^^^\n\n");

while(1)

{

if(fgets(c, 100, fpp) ==NULL)

break;

else

printf("%s", c);

}

fclose(fpp);

return;

}

void reg()

{

FILE \*fp;

char c,username\_check[40]; int z=0;

fp=fopen("Web\_reg.txt","ab+");

printf("\n\n\t\t\t\tWELCOME TO REGISTER ZONE");

printf("\n\t\t\t\t^^^^^^^^^^^^^^^^^^^^^^^^");

for(i=0;i<100;i++){

printf("\n\n\t\t\t\t ENTER USERNAME: ");

scanf("%s",username\_check);

while(!feof(fp)){

fread(&w[i],sizeof(w[i]),1,fp);

if(strcmp(username\_check,w[i].name)==0){

printf("\n\n\t\t\tUSERNAME ALREDY EXISTS");

system("cls");

reg();

}

else{

strcpy(w[i].name,username\_check);

break;

}

}

back:

z = 0;

printf("\n\n\t\t\t\t DESIRED PASSWORD: ");

//done till here

while((c=getch())!=13)

{

w[i].pass[z++]=c;

printf("%c",'\*');

}

printf("\n\n\t\t\t\tre-enter the password : ");

while((c=getch())!=13)

{

w[i].repass[z++]=c;

printf("%c",'\*');

}

char str1[40] = " ";

char str2[40] = " ";

for(int l=0;l<z;l++){

strncat(str1,&w[i].pass[l],1);

strncat(str2,&w[i].repass[l],1);

}

int result;

result = strcmp(str1,str2);

if(result!=0){

goto back;

}

fwrite(&w[i],sizeof(w[i]),1,fp);

fclose(fp);

printf("\n\n\tPress enter if you agree with Username and Password");

if((c=getch())==13)

{

system("cls");

for(int i=0;i<20;i++){

printf("\t-");

usleep(1);

}

printf("\n\n\t\tYou are successfully registered");

printf("\n\n\t\tDo you want to login into your account??\n\n\t\t ");

printf("> PRESS 1 FOR YES\n\n\t\t > PRESS 2 FOR NO\n\n");

for(int i=0;i<20;i++){

printf("\t-");

usleep(1);

}

scanf("%d",&n);

switch(n)

{

case 1: system("cls");

login();

break;

case 2: system("cls");

printf("\n\n\n\t\t\t\t\tTHANK YOU FOR REGISTERING");

break;

}

}

break;

}

getch();

}

void login()

{

FILE \*fp;

char c,name[30],pass[30]; int z=0;

int checkusername,checkpassword;

fp=fopen("Web\_reg.txt","rb");

printf("\n\n\t\t\t\tWELCOME TO LOG IN ZONE");

printf("\n\t\t\t\t^^^^^^^^^^^^^^^^^^^^^^");

for(i=0;i<1000;i++)

{

printf("\n\n\t\t\t\t ENTER USERNAME: ");

scanf("%s",name);

printf("\n\n\t\t\t\t ENTER PASSWORD: ");

while((c=getch())!=13)

{

pass[z++]=c;

printf("%c",'\*');

}

pass[z]='\0';

while(!feof(fp))

{

fread(&w[i],sizeof(w[i]),1,fp);

checkusername=strcmp(name,w[i].name);

checkpassword=strcmp(pass,w[i].pass);

if(checkusername==0)

{

if(checkpassword==0){

system("cls");

printf("\n\n\n\t\t\tLOGIN IS SUCCESSFULL!!");

printf("\n\n\n\t\t\t\tWELCOME, HAVE A NICE DAY\n Press Enter !");

break;

}

}

else if(checkusername==0)

{

if(checkpassword!=0){

printf("\n\n\n\t\t\tWRONG PASSWORD!! Not %s??",name);

printf("\n\n\t\t\t\t (Press 'A/a' to re-login)\n");

if(getch()=='A'||getch()=='a')

login();

}

}

else if(checkusername!=0)

{

printf("\n\n\n\t\t\tYou are not a Registered User\n \t\t\tPress enter to register yourself");

if(getch()==13)

system("cls");

reg();

}

}

break;

}

getch();

}

**3.2.3 GITHUB LINKS**

<https://github.com/navyashruthi15>

<https://github.com/bhavanisilasagaram>

* 1. **Testing**

**Table

Description automatically generated**

**Table

Description automatically generated**

**Table

Description automatically generated**

**Table

Description automatically generated**

**Table

Description automatically generated**

Empty Empty

No entered notes

**Table

Description automatically generated**

**Table

Description automatically generatedTable

Description automatically generated**

UC04

--- ----

Empty to do ist

**Table

Description automatically generated**

**Table

Description automatically generated**

**Table

Description automatically generated**

**Table

Description automatically generated**

**Table

Description automatically generatedTable

Description automatically generated**

UC07

Shows reminders with sound

Shows reminders with sound

Already added reminders

**Test Case**

Test Case ID: UCO1 Use Case ID:

Test Case Title: OVERLAPPING EVENTS UC08

Description: : Show all overlapping events with name and time

Test Steps: Ask to show overlapping Expected Result Actual Result

Events system display

And goes to main menu

Overlapping events are overlapping events

Displayed with time Displayed withtime

**Test Case**

Test Case ID:UC02 Use Case ID:

Test Case Title: show month no booked days UC08

Test Case Description: Booked days are not present

Test Steps: Enter a month and a year Expected Result Actual Result

Shows the month Shows the month

With no booked days with no booked days

**Test Case Template**

Test Case ID: UC01 Use Case ID:

Test Case Title: Current Time UC09

Test Case Description: current time displays

Test Steps: user prompts the system to Expected Result Actual Result

display the time

Time is displayed at the Time is displayed

Top at the top

**4.0 SCREENSHOTS OF EXECUTION**

As soon as the user opens the software, it displays the following

**Text

Description automatically generated**

**Graphical user interface, text

Description automatically generated**

**Text

Description automatically generated**

**Text

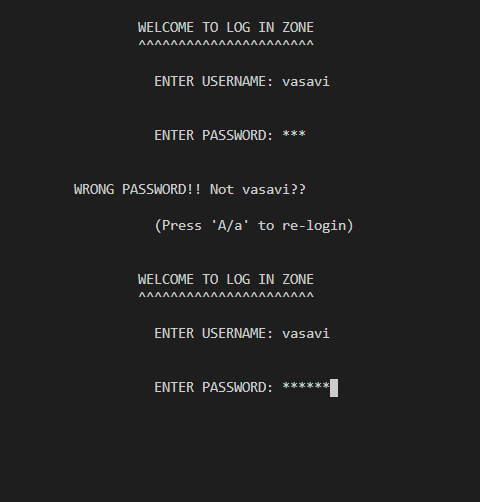
Description automatically generated**

**A picture containing text, night sky

Description automatically generated**

**Text

Description automatically generated**

****

**Text

Description automatically generated**

**Text

Description automatically generatedText

Description automatically generatedText

Description automatically generatedText

Description automatically generatedA picture containing text

Description automatically generatedText

Description automatically generatedText

Description automatically generatedText

Description automatically generatedText

Description automatically generatedText

Description automatically generatedText

Description automatically generatedText

Description automatically generated**

**5.0 ADDITIONAL KNOWLEDGE GAINED:**

Most of our project is actually based on file handling so we have gained lot of improvement in that part. We have also used functions like usleep(), beep(), cls(). We learnt how to write module wise code.

6.0 **CONCLUSION**

Conclusion on overall, the project has achieved its objectives. The project has provided a user/software environment.

We are living in a society where technology plays a major role. It is time to change from paper calendar to the calendaring software. Calendaring software ultimately saves time for the user. We have also provided notes, reminders, to-do list, displaying the current time, checking overlapping events etc features which added more colour to the software.

We have gained immense knowledge in coding and programming throughout the duration of this project. The joy of working and thrill involved while debugging and fixing errors gave us a feel of being an actual software developer. We have improved our collaborating skills. Even though we faced challenges during the coding phase, we helped each other out and finally achieved the desired product. It was a wonderful opportunity for both of us, and will be looking forward to many more such opportunities in the future.

**7.0 FUTURE WORK**

We will add time blocking-Allows users to organise their days into chunks. Additionally the software may provide an appointment book , address book, and/or contact list.

8.0 **REFERENCES**

We referred google and our c textbook for the completion of project successfully.

Some websites which came in handy were :

<https://stackoverflow.com/questions/29493837/how-to-make-a-beep-sound-in-c-on-windows>

<https://en.wikipedia.org/wiki/Calendaring_software>