

PROFESSIONAL TRAINING REPORT
at
Sathyabama Institute of Science and Technology
(Deemed to be University)

Submitted in partial fulfillment of the requirements for the award
of Bachelor of Technology Degree in Information Technology

By
Keerthiga.N

38120042



DEPARTMENT OF INFORMATION TECHNOLOGY
SCHOOL OF COMPUTING
SATHYABAMA INSTITUTE OF SCIENCE AND
TECHNOLOGY
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119. TAMILNADU

DECEMBER 2020



SCHOOL OF COMPUTING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the Bonafide work of
KEERTHIGA.N(Reg.no. 38120042) who carried out the project entitled "C
CALENDAR" under our supervision from June 2020 to December 2020.

Internal Guide

Mrs. Jithina jose M.E., Ph.D.,

Head of the Department

Dr. R. Subhashini M.E., Ph.D.,

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I KEERTHIGA.N hereby declare that the Project Report entitled “C CALENDAR” done by me under the guidance of Dr. Mary Gladence. L M.E., Ph.D., at Sathyabama Institute of Science and Technology (Deemed to be University), Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai - 600119 is submitted in partial fulfillment of the requirements for the award of Bachelor of Technology degree in Information Technology.

DATE: 23/01/2021

Place: Chennai

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge our sincere thanks to Board of management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey our thanks to **Dr. T. SASIKALA M.E., Ph.D., Dean, School of Computing** and **Dr. R. SUBHASHINI M.E., Ph.D., Head of the Department, Department of Information Technology** for providing us the necessary support and details at the right time during the progressive reviews.

I would like to express our sincere and deep sense of gratitude to our Project **Mrs.Jithina jose M.E., Ph.D.**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express our thanks to all Teaching and Non-teaching staff members of the Department of **INFORMATION TECHNOLOGY** who were helpful in many ways for the completion of the project.

TRAINING CERTIFICATE



VERIFIED ✓

CERTIFICATE OF COMPLETION

THIS IS TO CERTIFY THAT
KEERTHIGA N
HAS SUCCESSFULLY COMPLETED
C PROGRAMMING

DATE: 01/04/2020

CERTIFICATE NUMBER: 7A5351KF850R115s1H

*THIS CERTIFICATE IS AUTO GENERATED



S.P. BALAMURUGAN
CO-FOUNDER AND CEO

ABSTRACT

C CALENDER

Calendar+ is an android application which accesses the calendars of multiple users, for instance members of a family or staff in a particular department and will pull out the events from every calendar. Hence, everyone in the group can see others' events on one screen in customized format. It also helps the user to create, edit, and delete events of his/her calendar. Additionally, this application will use an efficient strategy to update data only if there is any change in the events, making it faster. There are many other applications in the Play store but they cannot get rid of redundant events and reminders of events belonging to other users. This causes a lot of disturbance to the user. On the contrary, Calendar+ application will use an intelligent algorithm to compare the events in many ways based on the time, name, and place of event. Furthermore, this application will only generate reminders for the user events

CHAPTER NO	TITLE NAME	PAGE NO
	ABSTRACT	7
	LIST OF FIGURES	9
	LIST OF ABBREVIATION	10
1	INTRODUCTION	11
	1.1 GENERAL	11
	1.2 OUTLINE OF THE PROJECT	11
2	AIM AND SCOPE	12
	2.1 AIM	12
	2.2 PROBLEM STATEMENT	12
	2.3 SCOPE	12
3	SYSTEM ANALYSIS AND DESIGN	13
	3.1 GENERAL	13
	3.2 SOFTWARES USED	13
	3.3 PROJECT DESCRIPTION	14
4	RESULT AND DISCUSSION	15
	4.1 SYSTEM REQUIREMENTS	15
	4.2 RESULT	15
5	CONCLUSION AND FUTURE WORK	16
	5.1 CONCLUSION	16
	5.2 FUTURE WORK	16
	5.3 REFERENCE	16
	APPENDIX	17
	A. SCREENSHOTS	17
	B. SOURCE CODE	18

FIGURE NO	FIGURE NAME	PAGE NO
5.1	OUTPUT	19

LIST OF ABBREVIATIONS

ABBREVIATIONS	EXPANSION
HTML	HYPERTEXT MARK-UP LANGUAGE
CSS	CASCADING STYLE SHEETS
JS	JAVA SCRIPT

CHAPTER 1

INTRODUCTION

1.1 GENERAL

Smart phones are extremely versatile gadgets, offering unlimited chances to utilize our time efficiently. On average, a smart phone has 41 applications, and every day newer phones are being activated. Perhaps the most startling statistic is that currently, 1.3 million Android devices are activated every day which means that every 24 hours, more than four times as many new smart phones and tablets are set up than babies are born. And once the devices are activated, they are checked on average 150 times a day once every 6.5 minutes. The apps in both Apple's and Google's app store combined is over 2 million. Admittedly, the apps are the purpose behind the smart phone. Furthermore, on average every smart phone contains 41 apps which are used for regular needs.

1.2 OUTLINE OF THE PROJECT

It is necessary to know about others schedule before meeting them so that we don't need to wait. For an instance if a group of 10 people are planning to gettogether, it takes a great amount of time to know each of the group member's calendars and schedule an event. This is the motivation behind this application to store the event details in a single calendar with multiple user access. The other reason behind this project is that few applications like a Calendar application, can support multiple user accounts in a single calendar but they generate alert notifications for unrelated events to the user and also cannot get rid of redundant events. For an instance, in the above example if 10 people have the same event, the present available apps give alert notification 10 times for a single event which creates a lot of disturbance to users. The Calendar+ makes sure that the users are not disturbed by the alert notifications for the events they are not related to and also gets rid of redundant events.

CHAPTER 2

AIM AND SCOPE

2.1 AIM

Android operating system provides more advantages for creating and accessing a single calendar with many users. So, it's most appropriate to develop this project as a calendar application in android platform.

2.2 PROBLEM STATEMENT

One of our problems with existing calendar applications is accessing a calendar by multiple users. Other developers have tried to solve this issue in a different way. However, the existing apps have their own disadvantages. Google calendar is most popular application in the present market. In spite of its popular features, it has some drawbacks too.

2.3 SCOPE

After the entire events are retrieved the user can add, update and delete events in this multiple access calendar. Before using this application by multiple users, one of the users must create a shared password and this can be shared with other users to use this calendar.

CHAPTER 3

SYSTEM ANALYSIS

3.1 GENERAL

3.1.1 USER INTERFACE

The user interface (UI) is the point of human-computer interaction and communication in a device. This can include display screens, keyboards, a mouse and the appearance of a desktop. It is also the way through which a user interacts with an application or a website

3.2 SOFTWARES USED

3.2.1 CODE EDITORS USED

GDB Online Debugger

GDB online is an online compiler and debugger tool for C, C++, Python, PHP, Ruby, C#, VB, Perl, Swift, Prolog, JavaScript, Pascal, HTML, CSS, JS. Code, Compile, Run and Debug online from anywhere in world.

3.2.2 PROGRAMMING LANGUAGES USED

C PROGRAM

C is a general-purpose, procedural computer programming language supporting structured programming, lexical variable scope, and recursion, with a static type system. By design, C provides constructs that map efficiently to typical machine instructions.

3.3 PROJECT DESCRIPTION

This application uses Parse Web Server to store all the users' events and can be accessed in a single calendar. The app synchronizes with the mobile device when it is installed. Synchronization is the process of initiating consistency among the source data to target data storage and vice versa. This process takes several minutes to synchronize the data from the source to target storage. In android mobile operating system, user has facility to synchronize with the account in the device. By doing so, the application will synchronize the data from the mobile device. With the storage once the synchronization is completed, it will automatically collect the data like contacts, groups and calendar resources from the device. Similarly, the Calendar+ application synchronizes with the Google account in the default calendar on the device and stores it on the Parse Web Server. Later, the user can directly add, delete and update event details directly from the Calendar+ application. This application also has many user friendly functionalities like getting rid of redundant events by using an algorithm to merge and unmerge the events. Furthermore, a color scheme is used to differentiate the users' events. More information on color scheme is mentioned in 4th chapter. Adding to that, the app provides customized format to view the calendar in a day, week, month or year mode.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 SYSTEM REQUIREMENTS

- Operating System: Windows/Mac
- RAM: 4GB
- Processor: 64x 1.0Ghz processor
- ROM: 8GB
- Parse Web server to store group events

4.2 RESULT

The outcome of this project is the Calendar+ application. This can be used as regular calendar application to create/update and delete events. Additionally, users can view events of the members in the group. It can be distributed via email for testing and usage purposes.

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 CONCLUSION

Calendar+ application provides a shared password to access and store other users' event details which can be shared by email or by other forms of communication. This application stores all the users' events in Parse Web Server. SQLite is used as a backend to store the information about event and user details in the local database before adding to the Parse Web Server. There is no limit on the number of user accounts for accessing the calendar. User can add, update and delete event details in the application. Related events are merged/unmerged using an algorithm. Events can be seen in a customized view format like day/week/month and year. This application will automatically synchronize phone book contacts from mobile phone.

5.2 FUTURE WORK

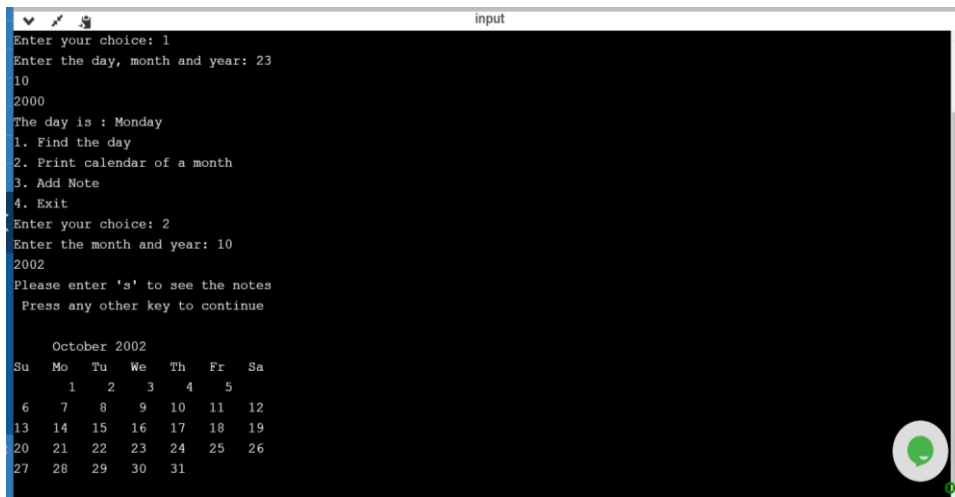
In the future, by using two-step verification more security can be provided for Calendar+ shared password which improves data security levels. Multiple accounts for a user can be implemented. On the other hand, merging/unmerging algorithm which is used to compare the events can use artificial intelligence and thus merging/unmerging can be done on more parameters. Additionally, we can attract more users to use Calendar+ by providing better interface with advanced functionalities. More importantly, this app can be extended to fetch events not only from Google calendar but also from other available calendars as well.

5.3 REFERENCES

1. <https://followtutorials.com>
2. <https://www.w3schools.com/html/default.asp>

APPENDIX

A. SCREENSHOTS



```
input
Enter your choice: 1
Enter the day, month and year: 23
10
2000
The day is : Monday
1. Find the day
2. Print calendar of a month
3. Add Note
4. Exit
Enter your choice: 2
Enter the month and year: 10
2002
Please enter 's' to see the notes
Press any other key to continue

October 2002
Su Mo Tu We Th Fr Sa
1 2 3 4 5
6 7 8 9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31
```



```
input
27 28 29 30 31
s
Here are list of notes for 10 2002
1. Find the day
2. Print calendar of a month
3. Add Note
4. Exit
Enter your choice: 3
Enter the day, month and year: 20
11
2005
Enter the note: hello all
Note added sucessfully
1. Find the day
2. Print calendar of a month
3. Add Note
4. Exit
Enter your choice: 4
Bye!!
...Program finished with exit code 0
Press ENTER to exit console.
```

B. SOURCE CODE

```
#include <stdio.h>
#include <stdlib.h>

int isLeapYear( int year );          /* True if leap year */
int leapYears( int year );          /* The number of leap
year */
int todayof( int y, int m, int d); /* The number of days
since the beginning of the year */
long days( int y, int m, int d);    /* Total number of days
*/
void calendar(int y, int m);        /* display calendar at m
y */
int getDayNumber(int d,int m,int y);
char *getName(int day);

// Copied from
// https://stackoverflow.com/questions/35103745/read-a-string-as-an-input-using-scanf
void flush()
{
    int c;
    while ((c = getchar()) != '\n' && c != EOF);
}
```

```

typedef struct {
    int day;
    int month;
    int year;
    char note[255];
} Note;

int main(int argc, char* argv[]){
    int year,month, day;
    char choice;
    Note note;
    FILE *fp;

    fp = fopen("note.bin", "r");
    if (fp == NULL) {
        fp = fopen("note.bin", "w");
    }
    fclose(fp);

    while(1) {
        printf("1. Find the day\n");
        printf("2. Print calendar of a month\n");
        printf("3. Add Note\n");
        printf("4. Exit\n");
        printf("Enter your choice: ");
        scanf("\n%c", &choice);
    }
}

```

```

switch(choice) {
    case '1':
        printf("Enter the day, month and year: ");
        scanf("%d %d %d", &day, &month, &year);
        printf("The day is : %s\n",
getName(getDayNumber(day, month, year)));
        break;
    case '2':
        printf("Enter the month and year: ");
        scanf("%d %d", &month, &year);
        printf("Please enter 's' to see the notes\n Press
any other key to continue\n");
        calendar(year, month);
        break;
    case '3':
        printf("Enter the day, month and year: ");
        scanf("%d %d %d", &note.day, &note.month,
&note.year);
        flush();
        printf("Enter the note: ");
        fgets(note.note, 255, stdin);
        fp = fopen("note.bin", "a+");
        if (fp == NULL) {
            printf("File note.bin can not be opened\n");
            exit(1);
        }
}

```

```

        fwrite(&note, sizeof(Note), 1, fp);
        printf("Note added sucessfully\n");
        fclose(fp);
        break;
        case '4':
            printf("Bye!!");
            exit(0);
            break;
        default:
            printf("Not a valid option\n");
            break;
    }
}
return 0;
}

int isLeapYear( int y ){
    return(y % 400 == 0) || ((y % 4 == 0) && (y % 100 !=
0));
}

int leapYears( int y ){
    return y/4 - y/100 + y/400;
}

int todayOf( int y, int m, int d) {
    static int DayOfMonth[] =

```

```

        { -1,0,31,59,90,120,151,181,212,243,273,304,334};
    return DayOfMonth[m] + d + ((m>2 && isLeapYear(y)) ? 1 :
0);
}

```

```

long days( int y, int m, int d){
    int lastYear;
    lastYear = y - 1;
    return 365L * lastYear + leapYears(lastYear) +
todayOf(y,m,d);
}

```

```

void calendar(int y, int m){
    FILE *fp;
    Note* notes, note;
    int len, j, hasNote = 0;
    char choice;
    const char *NameOfMonth[] = { NULL/*dummp*/,
        "January", "February", "March", "April", "May",
"June",
        "July", "August", "September", "October",
"November", "December"
    };
    char Week[] = "Su  Mo  Tu  We  Th  Fr  Sa";
    int DayOfMonth[] =
        { -1,31,28,31,30,31,30,31,31,30,31,30,31 };
    int weekOfTopDay;

```

```

int i, day;

weekOfTopDay = days(y, m, 1) % 7;

fp = fopen("note.bin", "rb");
if (fp == NULL) {
    printf("Couldn't read notes\n");
}
len = 0;
while(fread(&note, sizeof(Note), 1, fp)) {
    if (note.year == y && note.month == m) {
        len++;
    }
}
rewind(fp);
j = 0;
notes = (Note*) malloc (sizeof(Note) * len);
while(fread(&note, sizeof(Note), 1, fp)) {
    if (note.year == y && note.month == m) {
        notes[j] = note;
        j++;
    }
}

fclose(fp);

if(isLeapYear(y))

```

```

        DayOfMonth[2] = 29;
printf("\n      %s %d\n%s\n", NameOfMonth[m], y, Week);

for(i=0;i<weekOfTopDay;i++)
    printf("  ");
for(i=weekOfTopDay,day=1;day <=
DayOfMonth[m];i++,day++){
    hasNote = 0;
    for (j = 0; j < len; j++) {
        if (notes[j].day == day) {
            printf("|%2d| ",day);
            hasNote = 1;
            break;
        }
    }
    if (hasNote == 0) {
        printf("%2d  ",day);
    }
    if(i % 7 == 6)
        printf("\n");
}
printf("\n");
scanf("\n%c", &choice);
if (choice == 's') {
    printf("Here are list of notes for %d %d\n", m, y);
    for (j = 0; j < len; j++) {
        printf("%d: %s\n", notes[j].day, notes[j].note);
    }
}

```

```

    }
} else {
    return;
}
}

int getDayNumber(int d, int m, int y){ //returns the day
number
    static int t[] = {0, 3, 2, 5, 0, 3, 5, 1, 4, 6, 2, 4};
    y -= m < 3;
    return (y + y/4 - y/100 + y/400 + t[m-1] + d) % 7;
}

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("Error: Invalid Argument Passed");
    }
}
}

```