

# JOBSFOEHER



A Course Project Report in the course

Data Structures

**Computer Science& Engineering**

**By**

18K41A0502

A. DHARANI

18K41A0503

A. NAVYA

18K41A0504

A. VARSHA RAO

18K41A0505

AREEFA

18K41A0506

A. JAHNAVI REDDY

**Under the Guidance of**

Dr. M. Sheshikala  
Assoc. Prof

**Submitted to**

**March, 2019**



**SR**  
**Engineering**  
**College**  
Innovation . Creativity . Entrepreneurship

## **ABSTRACT**

Jobsfoeher would provide a great platform for the women who wanted to start their career so as to fulfill their dream of working in a company. It mainly focuses on the Women Empowerment. There are many women who are educated but still having guilt in them of not yet doing a job due to few circumstances. Though there are many sites like Nowkri.com, internshaala but the thing is, so as to register for jobs in these sites, we need to submit the proofs of our talent through certificates or previous work experience. Thus, the women who newly wanted to start their career are lagging in this and giving up.

Thus, this company would act as a basic step of the career by providing flexible jobs to the woman of short duration span through which women can get work experience and build confidence in them to apply jobs in large platforms and earn the satisfaction that they will.

The Jobsfoeher site that is proposed to be implemented should implement the services for both users like Jobseeker and the admin. Jobseeker must be able register, create profile in the site and must able to opt the job preferences. Whereas admin must be able to see the profiles of the job applicants and send permission to the job seek to opt preferences of job and then must be able to see the opted preferences and allot the jobs accordingly.

Thus, a proper integration and communication must be developed between the admin and job applicants.

# TABLE OF CONTENTS

<b>TITLE</b>	<b>PAGENO.</b>
<b>1. INTRODUCTION</b>	
1.1 ABOUT THE PROJECT	5
1.2 PURPOSE OF THE PROJECT	5
<b>2. PROJECT DOCUMENTATION</b>	
2.1 MODULES	7
2.2 PROJECT REQUIREMENTS	8-9
2.3 APPLICATIONS USED	10-12
<b>3. CODING</b>	14-28
<b>4. RESULTS</b>	30-36
<b>5. CONCLUSION</b>	37

# **Title 1**

## **INTRODUCTION**

## **1.1 INTRODUCTION TO PROJECT**

Jobsfoeher is an online application site that aims in providing the jobs to the woman for short period of span with flexible work hours so that the women can learn the skills and get work experience to build up their career. This application must interface both the services of the job applicant and the admin. Jobseeker must be able register, create profile in the site and must able to opt the job preferences. Whereas admin must be able to see the profiles of the job applicants and send permission to the job seek to opt preferences of job and then must be able to see the opted preferences and allot the jobs accordingly.

Thus, proper integration of the services must be provided on the single platform through which both the users can have access based on the permissions. It must be designed in such a way that the job applicant cannot enter into the section of the admin.

## **1.2 PURPOSE OF THE PROJECT**

Jobsfoeher mainly focuses on the Women Empowerment. There are many women who are educated but still having guilt in them of not yet doing a job due to few circumstances. Though there are many sites like Nowkri.com, internshaala but the thing is, so as to register for jobs in these sites, we need to submit the proofs of our talent through certificates or previous work experience. Thus, the women who newly wanted to start their career are lagging in this and giving up.

Jobsfoeher, the online site must create a proper and good functioning interface between the job applicant and the admin. Thus, before developing the application, it is important to have a clear picture of the working and roles of both the services mentioned. Thus, to demonstrate the whole functioning of the site including the roles of admin and job seek through two different sections including their permissions, we need to implement it using the concepts of data structures in C.

## **Title 2**

# **PROJECT DOCUMENTATION**

## 2.1 MODULES

As told earlier, proper integration of services between both the admin and the job applicant must be created.

### **Roles of Job applicant:**

- Register to apply job :  
The job seek so as to apply for the jobs, she should register in the online site by entering name and mail id.
- Create profile :  
Once after registering in online website, in addition to name, mail id she should enter her age, qualification and address details to create her profile.
- Receive OTP :  
After creating profile, based on her registration order, she will receive the OTP from admin. Receiving OTP in the sense, she is eligible to opt the job preferences.
- Able to choose the job preferences :  
She can place at most four job preferences by accessing through OTP.

### **Roles of the admin:**

- View profiles :  
Once after all the registrations are closed, Admin can view all the profiles so as to send the OTP.
- Send OTP :  
Admin based on the order of registration numbers of the job applicants, send the OTP (NOTE: At a time only one OTP to one applicant must be sent and again after allotting the job to her only another OTP can be sent to another applicant)
- See job preferences opted by job seek :  
Here Admin can view the job preferences given by the applicant so as to allot the jobs according to preferences.
- Allot the jobs :  
After viewing the preferences, Admin allot the most preferred job by looking whether the job is allotted to other applicant or not.

## 2.2 PROJECT REQUIREMENTS

### HARDWARE

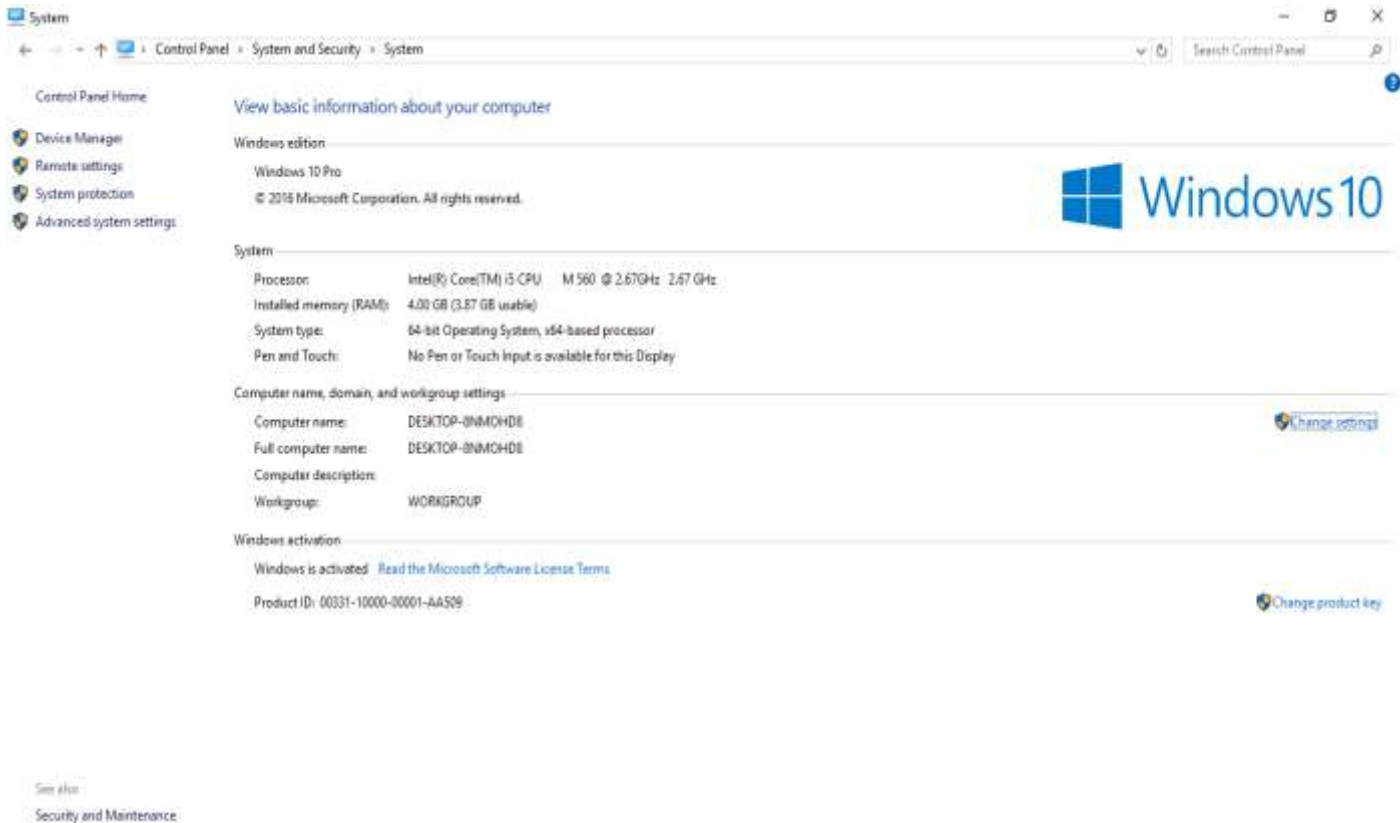
#### Windows Edition

Windows 10 Pro

**Processor** : Intel(R) Core™ & CPU

**RAM** : 4.00 GB

**System Type** : 64-bit Operating System





## SOFTWARE: DEV C++



Dev-C++ is a feature rich IDE, which comes complete with compiler for creating software in C++. It is a fork of the original Bloodshed Dev-C++ environment.

Key Features include:

- TDM-GCC 4.9.2 32/64bit.
- Syntax highlighting.
- Code completion.
- Code insight.
- Editable shortcuts.
- GPROF profiling.
- GDB debugging.
- AStyle code formatting.
- Devpak IDE extensions.
- External tools.

Included in the Dev-C++ environment are all of the standard features that are necessary for writing, compiling, debugging, and executing programs written in C. Dev-C++ has been designed for the hard-core C++ programmer as it allows

you to compose all of your source code without the simple features, which are common with most programming environments for beginners.

## 2.3 APPLICATIONS USED

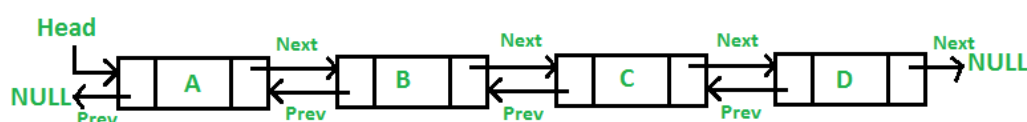
So as to implement the modules and generate proper interface between the Admin and the job applicant, we need to have a clear cut about the functionality. For this we are including the programming concepts of C like given below.

### 1. DOUBLE LINKED LIST:

A **Doubly Linked List (DLL)** contains an extra pointer, typically called previous pointer, together with next pointer and data which are there in singly linked list.

#### Advantages

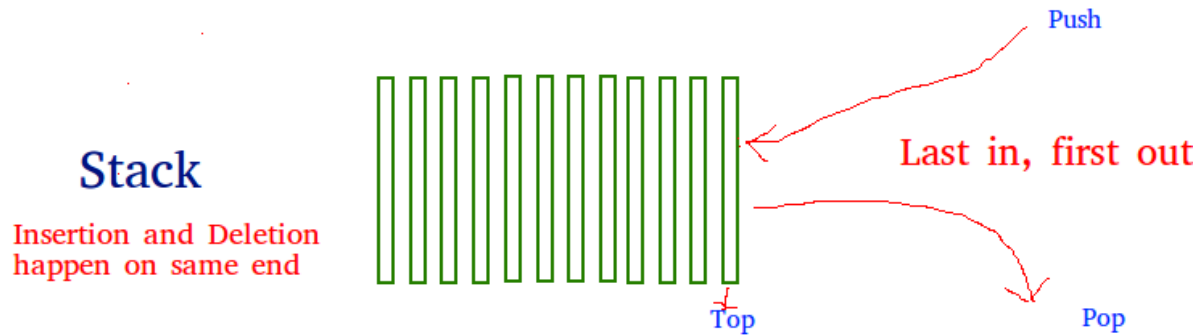
- 1) A DLL can be traversed in both forward and backward direction.
  - 2) The delete operation in DLL is more efficient if pointer to the node to be deleted is given.
  - 3) We can quickly insert a new node before a given node.
- In singly linked list, to delete a node, pointer to the previous node is needed. To get this previous node, sometimes the list is traversed. In DLL, we can get the previous node using previous pointer.



**In our project we used double linked list in admin section so as to view the profiles one by one.**

### 2. Stack:

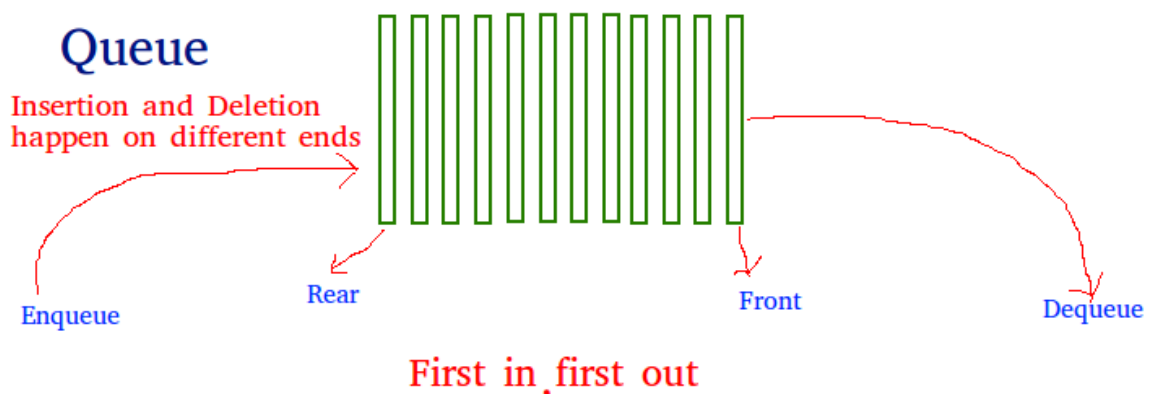
Stack is a linear data structure which follows a particular order in which the operations are performed. The order may be LIFO (Last In First Out) or FILO(First In Last Out).



This concept is used to store the job preferences in order. So that the job applicant wanted to delete the recently opted job, she can easily delete by stack concept (Last in first out).

### 3. Queue:

A Queue is a linear structure which follows a particular order in which the operations are performed. The order is First In First Out (FIFO). A good example of a queue is any queue of consumers for a resource where the consumer that came first is served first. The difference between stacks and queues is in removing. In a stack we remove the item the most recently added; in a queue, we remove the item the least recently added.



Admin when wanted to allot the jobs, he should allot the first preferred job. In this the queue concept is used (First in first out).

### 4. ARRAY:

An array is a collection of items stored at contiguous memory locations. The idea is to store multiple items of the same type together. This makes it easier to calculate the position of each element by simply adding an offset to a base

value, i.e., the memory location of the first element of the array (generally denoted by the name of the array).

Memory Location									
200	201	202	203	204	205	206	▪	▪	▪
U	B	F	D	A	E	C	▪	▪	▪
0	1	2	3	4	5	6	▪	▪	▪
Index									

Thus, a vast usage of arrays is done in our project for storing of variables and implementing all the above different applications.

## 5. Strings:

Strings are defined as an array of characters. The difference between a character array and a string is the string is terminated with a special character '\0'.

	0	1	2	3	4	5
str	G	e	e	k	s	\0
Address	0x23452	0x23453	0x23454	0x23455	0x23456	0x23457

Strings are used for the profile data and different string handling functions are used in case of passwords and all.

## 6. Files:

Different file operations are used for different purpose in the project.

Creation of a new file (**fopen with attributes as “a” or “a+” or “w” or “w+”**). Opening an existing file (**fopen**). Reading from file (**fscanf or fgetc**). Writing to a file (**fprintf or fputs**). Closing a file (**fclose**).

**Tile 3**

**CODING**

## MAIN SECTION:

In main, a function is called Permit().

This function is meant for the permissions. That means as there are 2 sections, one is admin and other is job seek, the job seek should not be allowed in the admin section. For this we placed a password for admin to access the admin section. That password is resided in a file password.txt.

### Code:

```
void Permit()
{
int ch;
char psswd[8];
char str[5];
int count1=1;
do{
printf("Let us know who you are\n");
printf("1.Job Seeker\n");
printf("2. Admin\n");

printf("Enter the selection\n");
scanf("%d",&ch);
switch(ch) //For permitting to respective roles
{
case 1:Jobseek();
count1=0;
break;
case 2:printf("Enter the admin password!\n");
//To allow only admin
scanf("%s",&psswd);
FILE *f;int c;
f=fopen("D://password.txt","r");
int i=0,count=0;
while(((c=fgetc(f))!=EOF) && (i<5))
{
str[i]=c;
if(psswd[i]==str[i])
{
count++;
}
i++;
}
}
```

```

if(count==5)
{
printf("Welcome Admin\n");
Admin();
}
else
printf("Sorry you donot have permission into to    this\n");
count1=0;
fclose(f);
break;
default: return 0;
}
}while(count1);
}
int main()
{
Permit();
}

```

Thus by this code, we have separated the 2 interfaces.

**Let's look into the implementation of the roles respectively.**

## **JOB SEEKS SECTION:**

Through this the Job seekers can register themselves for applying of jobs.

**NOTE:** For demonstration we took only 2 registrations as show.

```

/** Whole Jobseek fuction - Main function */
void Jobseek()
{
int ch1,enter;
char psswd1[10];
char str1[10];
printf("\n1.To register for jobs\n");
printf("2.Opt the job preferences (Terms-only if you got the OTP)\n");
printf("Enter the choice\n");
scanf("%d",&ch1);
switch(ch1)
{
case 1:
/* For demo purpose we have considered only 2 registrations */

```

```

printf("Registrations are opened\n");
int n=1;
do{
registration(n);
n++;
}while(n!=3);
printf("Registrations are closed (Applicants exceeded)!\n");
printf("Do wait till the release of new jobs\n");
Permit();
break;
/** To not allow job seekers who didn't recieved otp and preferences permission **/
case 2:
printf("Enter psswd of jobseek\n");
scanf("%s",&psswd1);
FILE *f1;int c1;
f1=fopen("C:\\Users\\AREEFA\\Desktop\\pass2.txt","r");
int j=0,co=0;
while(((c1=fgetc(f1))!=EOF) && (j<5))
{
str1[j]=c1;
if(psswd1[j]==str1[j])
{
co++;
}
j++;
}
fclose(f1);
if(co==5)
{
printf("Welcome\n");
preferences();
}
else
{
printf("Sorry wrong OTP...\n");
printf("Enter the correct OTP which you recieved\n");
printf("Press 1 to re enter the otp\n");
printf("Press 2 to exit\n");
scanf("%d",&enter);
if(enter==1)
{
printf("Enter psswd of jobseek\n");

```



```

scanf("%s",&psswd1);
f1=fopen("C:\\Users\\AREEFA\\Desktop\\pass2.txt","r");
int j=0,co=0;
while(((c1=fgetc(f1))!=EOF) && (j<5))
{
    str1[j]=c1;
    if(psswd1[j]==str1[j])
    {
        co++;
    }
    j++;
}
if(co==5)
{
    printf("Welcome\n");
    preferences();
}
else{
    printf("Await for OTP\n");
    Permit();
}
}
else if(enter==2){
    printf("If you didn't receive OTP! Await.... till you get");
    Permit();
}
}
break;
default: Permit();
}

}

```

**If the Job seek wanted to register, he need to choose 1. Once she chosen 1, enter into registration() function.**

```

/** ----- Registration Process ----- */
void registration(int n)
{
    int c;
    FILE *fp1;
    profile();
}

```

```
fp1=fopen("C:\\Users\\AREEFA\\Desktop\\profile.txt","a");
fprintf(fp1,"%s %s %d %s %d\n",p.name,p.mail,p.age,p.q,n);
fclose(fp1);
}
```

In registration itself we are calling profile() function.

**/\*\*\*\* ---- PROFILE CREATION---- \*\*\*\*\*/**

```
void profile()
{
printf("Enter your Name\n");
scanf("%s",p.name);
printf("Enter the mail id\n");
mail();
printf("Enter your age\n");
scanf("%d",&p.age);
printf("Enter your qualifications\n");
scanf("%s",p.q);
}
```

**If the duplicate mail ID is given, it should not be accepted.**

**For this code is:**

**/\*\*\*\* For checking whether the mail id entered by the applicant exist or not \*\*\*\*\*/**

```
void mail()
{
FILE *fp;
char str[256];
int f=-991;
fp = fopen("C:\\Users\\AREEFA\\Desktop\\test.txt", "r");
//printf("Enter a mail id\n");
scanf("%s",p.mail);
while(!feof(fp) && f)
{
fscanf(fp, "%s\n", str);
if(strcmp(str,p.mail)==0)
{
puts("already exists");
printf("Use other mail\n");
mail(); /*To allow the applicant to register with other mail ID */
f=0;
}
}
fclose(fp);
```

```

if(f) /** If the mail Id entered donot exists... it will be stored in file for
reference **/
{
fp = fopen("C:\\Users\\AREEFA\\Desktop\\test.txt","a");
fprintf(fp,"%s\n",p.mail);
}
fclose(fp);
}

```

**Once profiles are created, now Job applicant waits for the OTP to be sent by admin.**

## **ADMIN SECTION**

```

void Admin()    /*Admin----*/
{
    int ch2;
    int regno;
    int otp;
    int reg1;
    do{
        printf("\n1.View profile\n");
        printf("2.Send OTP to job seek for preferences\n");
        printf("3.See preferences by applicant to which otp sent recently\n");
        printf("Enter your choice\n");
        scanf("%d",&ch2);
        switch(ch2)
        {
            case 1:Viewprofile();
            break;
            case 2:
                printf("\n enter reg no");
                scanf("%d",&regno);
                printf("Send the OTP to job applicant %d\n",regno);
                scanf("%d",&otp);
                FILE *fp2;
                fp2=fopen("C:\\Users\\AREEFA\\Desktop\\pass2.txt","w");
                fprintf(fp2,"%d",otp);
                fclose(fp2);
                printf("OTP sent...!!!");
                Message();
                break;
            case 3:
                printf("Enter the reg no., whose job preferences you want to view\n");

```

```

printf("TERMS AND CONDITIONS: To which you have sent otp only\n");
scanf("%d",&reg1);
job(reg1);
exit (0);
break;
default: exit(0);
}
}while(1);
}

```

**First admin views the profiles of all job applicants soon once the registrations are closed.**

**Double linked list concept is used**

```

struct dll
{
char Name[30];
char mail[30];
char q[50];
int regn;
int age;
struct dll *next;
struct dll *prev;
}s;
typedef struct dll node;
node *start=NULL;
node *temp,*newnode;
node *prev1;
node *getnode(char *Name, char *mail,int *age,char *q,int *regn)
{
newnode=(node *)malloc(sizeof(node));
strcpy(newnode->Name,Name);
strcpy(newnode->mail,mail);
newnode->age=age;
strcpy(newnode->q,q);
newnode->regn=regn;
newnode->next=NULL;
newnode->prev=NULL;
}
void read()
{
FILE *fp;

```

```

fp=fopen("C:\\Users\\AREEFA\\Desktop\\profile.txt","r");
while(!feof(fp)){
fscanf(fp,"%s %s %d %s %d\n",s.Name,s.mail,&s.age,s.q,&s.regn);
newnode = getnode(s.Name, s.mail,s.age,s.q,s.regn);
if(start==NULL)
start = newnode;
else
{
temp = start;
while(temp->next!=NULL)
temp = temp->next;
temp->next = newnode;
newnode->prev = temp;
}
}
fclose(fp);
}
void Viewprofile()
{
read();
temp=start;
while(temp->next!=NULL)
{
printf("\nName = %s\n",temp->Name);
printf("eMail = %s\n",temp->mail);
printf("Age = %d\n",temp->age);
printf("Qualification = %s\n",temp->q);
printf("Registration no. = %d\n-----\n",temp->regn);
temp=temp->next;
}
printf("Name = %s\n",temp->Name);
printf("eMail = %s\n",temp->mail);
printf("Age = %d\n",temp->age);
printf("Qualification = %s\n",temp->q);
printf("Registration no. = %d\n-----",temp->regn);
}

```

**After viewing the profiles, the admin will send OTP to a job applicant in the order of their registration numbers. (NOTE: One OTP at once. After allotting job only another OTP to other job seek is sent)**

```

void Message()
{

```

```

int otp;
if(ra==0)
{
printf("-----\n");
printf("\n\n----- Message - INBOX-----\n");
printf("ID-Jobsforher\n");
printf("Congratulations you have been shortlisted\n");
FILE *fp3;
fp3=fopen("C:\\Users\\Office\\Desktop\\pass2.txt","r");
while(!feof(fp3)){
fscanf(fp3,"%d",&otp);
}
printf("%d is the OTP and donot disclose with anyone\n",otp);
}
int n;
printf("Enter 1 to proceed to put job preferences\n");
scanf("%d",&n);
if(n==1)
{
Jobseek();
}
else{
ra=1;
Message();
}

}

```

## **OTP RECEIVED BY APPLICANT**

### **Job seek role to opt the preferences through OTP**

**/\*\* To not allow job seekers who didn't recieved otp and preferences permission \*\*/**

```

Case 2: printf("Enter psswd of jobseek\n");
scanf("%s",&psswd1);
FILE *f1;int c1;
f1=fopen("C:\\Users\\AREEFA\\Desktop\\pass2.txt","r");
int j=0,co=0;
while(((c1=fgetc(f1))!=EOF) && (j<5))
{
str1[j]=c1;
if(psswd1[j]==str1[j])

```

```

{
co++;
}
j++;
}
fclose(f1);
if(co==5)
{
printf("Welcome\n");
preferences();
}
else
{
printf("Sorry wrong OTP...\n");
printf("Enter the correct OTP which you recieved\n");
printf("Press 1 to re enter the otp\n");
printf("Press 2 to exit\n");
scanf("%d",&enter);
if(enter==1)
{
printf("Enter psswd of jobseek\n");
scanf("%s",&psswd1);
f1=fopen("C:\\Users\\AREEFA\\Desktop\\pass2.txt","r");
int j=0,co=0;
while(((c1=fgetc(f1))!=EOF) && (j<5))
{
str1[j]=c1;
if(psswd1[j]==str1[j])
{
co++;
}
j++;
}
if(co==5)
{
printf("Welcome\n");
preferences();
}
else{
printf("Await for OTP\n");
Permit();
}
}
}

```

```

else if(enter==2){
printf("If you didn't recieve OTP! Await.... till you get");
Permit();
}
}
break;

```

If OTP entered is correct then, enter into preferences() function

```

void preferences() /* Placing job preferences options */
{
do{
printf("\nAvailable jobs are:\n");
printf("A.IT Technician\n");
printf("B.Marketting\n");
printf("C.Management\n");
printf("D.Content Writing\n");
printf("E.Event manager and hosting\n");
printf("\n\n1.Opt your selection according to preferences order\n");
printf("2.To delete recently opted one\n");
printf("3.Display all the preferences opted\n");
printf("4.Press 4 to confirm preferences and halt!!!\n");
printf("enter ur choice\n");
scanf("%d",&ch);
switch(ch)
{
case 1:
printf("Enter your preference\n");
scanf(" %c",&ele);
insert(ele);
break;
case 2:delete();
break;
case 3:display2();
break;
case 4:printf("You ended the opting of preferences\n");
printf("Selected preferences are:\n");
display2();
exit(0);
break;
default:exit(0);
}
}while(1);

```



```
}
```

**So as to enter the preferences the stack concept is used.**

```
int top=-1,n,ch;
char s1[Max];
void insert(int ele)
{
if(top==Max-1)
printf("Stack overflow\n");
else{
s1[++top]=ele;
}
}
```

**If by mistake you wanted to give preference B but gave A, then delete last preference as shown (STACK- last I n first out)**

```
void delete()
{
if(top== -1)
{
printf("Stack underflow\n");
}
else
{
printf("Element deleted is %c\n",s1[top]);
s1[top--];
}
}
void display2()
{
int i=0;
if(top== -1)
{
printf("Underflow\n");
}
else{
for(i=0;i<=top;i++)
{
printf("%c ",s1[i]);
}
}
int j;
```

```

printf("Final preferences are\n");
FILE *fp5;
fp5=fopen("C:\\Users\\AREEFA\\Desktop\\we.txt","w");
fprintf(fp5,"%s",s1);
for(j=0;j<i;j++)
{
if(j==(i-1))
{
printf("%c",s1[j]);
}
else{
printf("%c->",s1[j]);
}
}
fclose(fp5);
givejob();
}

}

```

**By this the role of Job applicant is completed. She will wait for her selection.**

## **ADMIN ROLE AGAIN**

**Job preferences are stored in a file.**

**See the job preferences opted and to allot the jobs**

## **QUEUE CONCEPT**

Thus this code is about allotting jobs. Job already allotted cannot be allotted again. For this also we included logic as shown above.

First preferred job is most probably allocated if it is not allocated to any others. Thus, first in first out – Queue concept.

```

void pop(int reg1)
{
char job;
if(rear==NULL)
printf("queue is empty\n");
else
{

```

```

printf("Allot job\n");
scanf(" %c",&job);
printf("%c job is allotted to %d\n",job,reg1);
Admin();
}
}
void display()      //See job preferences
{
if(rear==NULL)
printf("\n queue is empty");
else
{
temp3=front;
while(temp3->next3!=NULL)
{
printf("%c",temp3->data3);
temp3=temp3->next3;
}
printf("%c\n",temp3->data3);
}
//int k=rear-1;
//printf("\nJob %c is alloted!!\n",q[k]);
}
void job(int reg1)
{
do{
printf("\n 1. See the job preferences");
printf("\n 2. pop and allot jobs\n");
printf("3. exit\n");
printf("\n Enter ur choice");
scanf("%d",&ch3);
switch(ch3)
{
case 1:      fp6=fopen("C:\\Users\\AREEFA\\Desktop\\we.txt","r");
while(((z=fgetc(fp6))!=EOF) )
{
if(isalpha(z)){
q[i]=z;
printf("%c ",q[i]);
i++;  }
}
printf("\n are Job preferences\n");
if(i>3){

```

```

for(j=0;j<3;j++)
{
printf("%c",q[j]);
if(j==0)
{
printf("%c already allotted... So don't allot\n",q[j]);
}
break;
}
printf("\n");
for(j=3;j<i;j++)
printf("%c ",q[j]);
}
else if(i<3){
for(j=0;j<i;j++)
printf("%c ",q[j]);
}
for(j=i-1;j>=0;j--)
{
push(q[j]);
}
break;
case 2:pop(reg1); break;
case 3:exit(0);
default: exit(0);
}
}while(1);
}

```

**Tile 4**

**RESULTS**

## PERMISSIONS

### Jobseek for registration

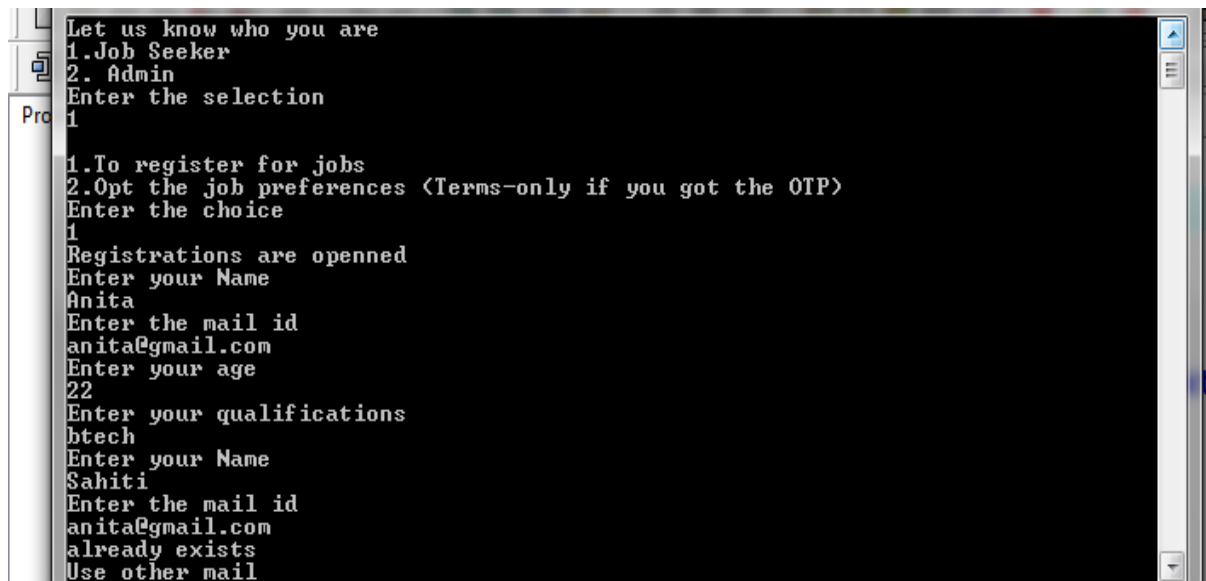
```
Let us know who you are
1.Job Seeker
2. Admin
Enter the selection
2
Enter the admin password!
abcd
Sorry you donot have permission into to this

-----
Process exited after 11.54 seconds with return value 0
Press any key to continue . . . _
```

```
Let us know who you are
1.Job Seeker
2. Admin
Enter the selection
2
Enter the admin password!
abcdef
Welcome Admin

1.View profile
2.Send OTP to job seek for preferences
3.See preferences by applicant to which otp sent recently
Enter your choice
```

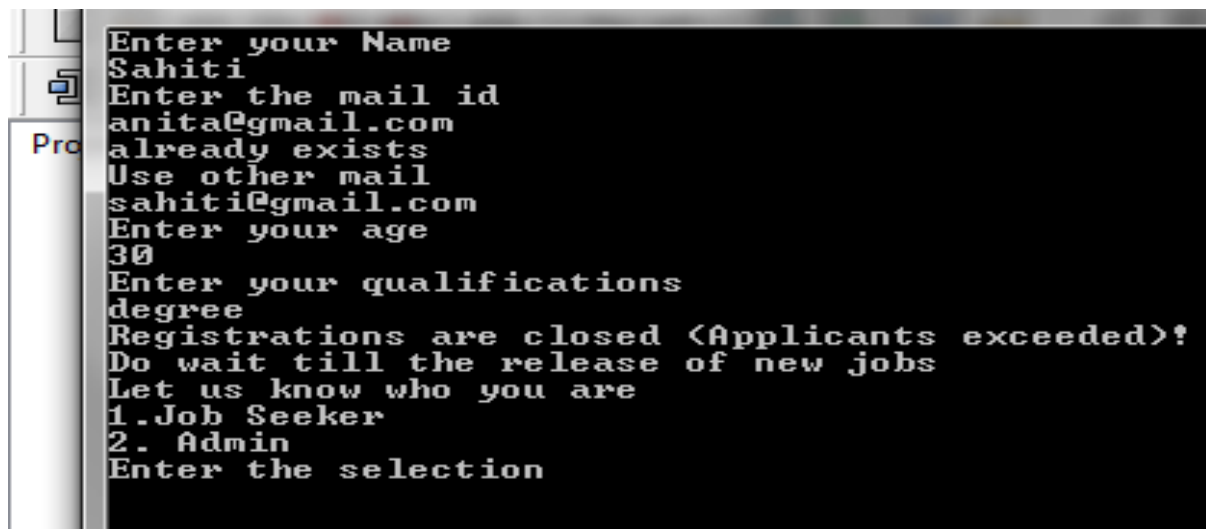
## REGISTRATIONS:



```
Let us know who you are
1.Job Seeker
2. Admin
Enter the selection
1
1.To register for jobs
2.Opt the job preferences <Terms-only if you got the OTP>
Enter the choice
1
Registrations are openned
Enter your Name
Anita
Enter the mail id
anita@gmail.com
Enter your age
22
Enter your qualifications
btech
Enter your Name
Sahiti
Enter the mail id
anita@gmail.com
already exists
Use other mail
```

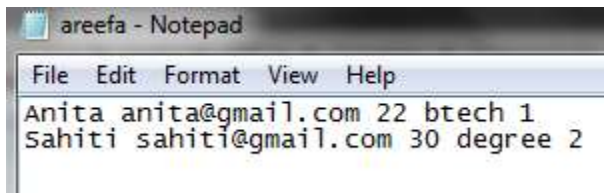
If mail already exists

## REGISTRATIONS CLOSED

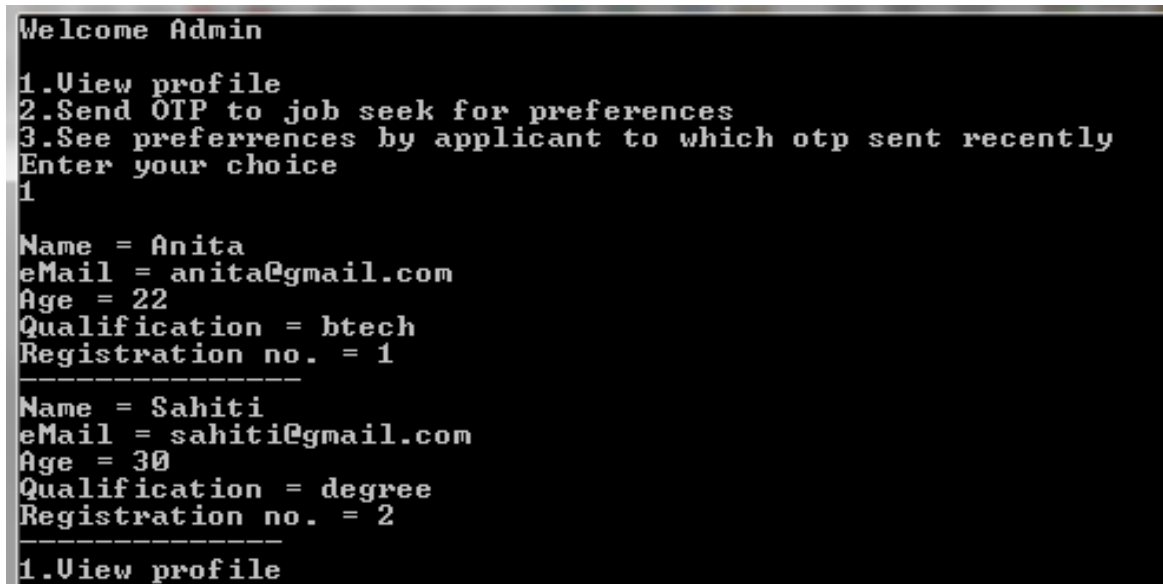


```
Enter your Name
Sahiti
Enter the mail id
anita@gmail.com
already exists
Use other mail
sahiti@gmail.com
Enter your age
30
Enter your qualifications
degree
Registrations are closed <Applicants exceeded>?
Do wait till the release of new jobs
Let us know who you are
1.Job Seeker
2. Admin
Enter the selection
```

## ADMIN PART (View profiles)



```
areefa - Notepad
File Edit Format View Help
Anita anita@gmail.com 22 btech 1
Sahiti sahiti@gmail.com 30 degree 2
```



```
Welcome Admin
1.View profile
2.Send OTP to job seek for preferences
3.See preferences by applicant to which otp sent recently
Enter your choice
1
Name = Anita
eMail = anita@gmail.com
Age = 22
Qualification = btech
Registration no. = 1
-----
Name = Sahiti
eMail = sahiti@gmail.com
Age = 30
Qualification = degree
Registration no. = 2
-----
1.View profile
```



```
-----
Name = Sahiti
eMail = sahiti@gmail.com
Age = 30
Qualification = degree
Registration no. = 2
-----
1.View profile
2.Send OTP to job seek for preferences
3.See preferences by applicant to which otp sent recently
Enter your choice
2
enter reg no1
Send the 5 digit OTP to job applicant 1
34567
OTP sent...!!!-----
```

SEND  
OTP



## MESSAGE SENT BY ADMIN

```
2.Send OTP to job seek for preferences
3.See preferences by applicant to which otp sent recently
Enter your choice
2

enter reg no1
Send the 5 digit OTP to job applicant 1
34567
OTP sent...!!!-----

----- Message - INBOX-----
ID-Jobsforher
Congratulations you have been shortlisted
34567 is the OTP and donot disclose with anyone
Enter 1 to proceed to put job preferences
```

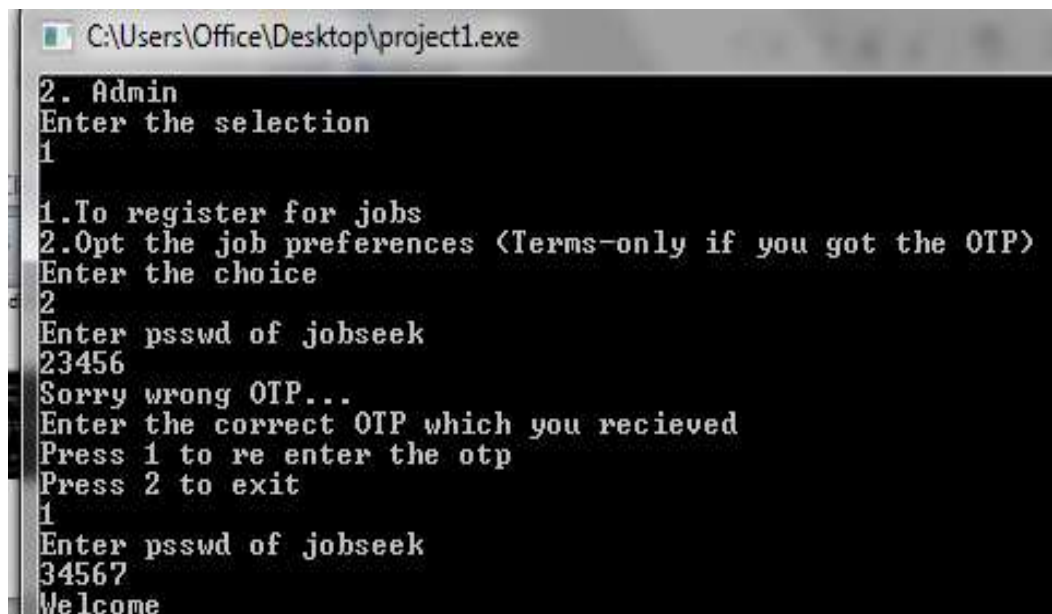
## MESSAGE RECEIVED BY APPLICANT

```
Project ----- Message - INBOX-----
ID-Jobsforher
Congratulations you have been shortlisted
34567 is the OTP and donot disclose with anyone
Enter 1 to proceed to put job preferences
1

1.To register for jobs
2.Opt the job preferences <Terms-only if you got the OTP>
Enter the choice
2
Enter psswd of jobseek
34567
Welcome

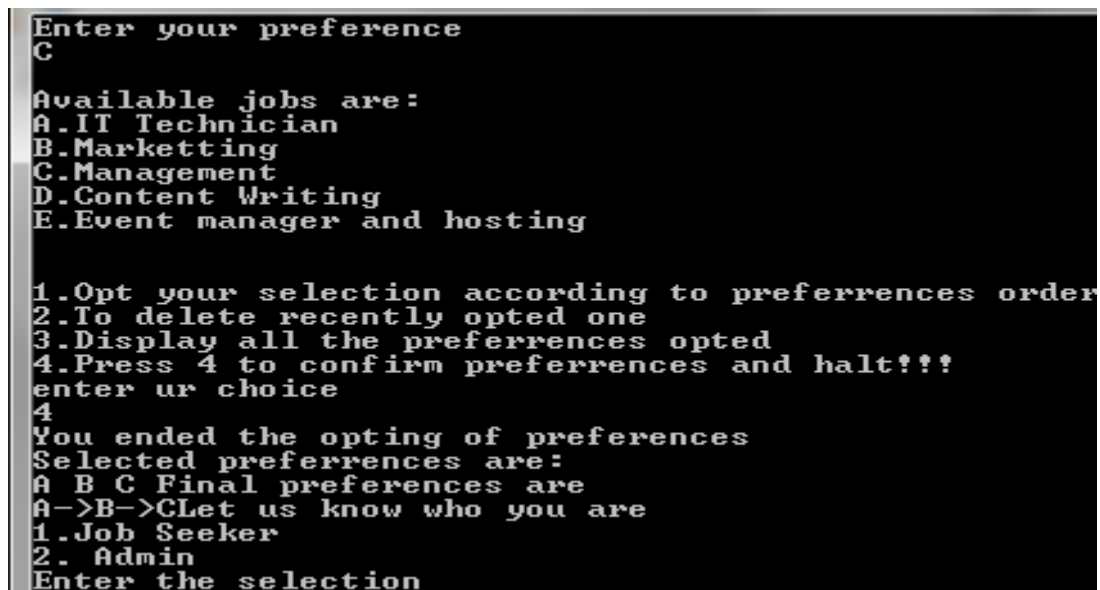
Available jobs are:
A.IT Technician
B.Marketting
C.Management
D.Content Writing
E.Event manager and hosting
```

## OPT JOB PREFERENCES BY JOB SEEK



```
C:\Users\Office\Desktop\project1.exe
2. Admin
Enter the selection
1
1.To register for jobs
2.Opt the job preferences <Terms-only if you got the OTP>
Enter the choice
2
Enter psswd of jobseek
23456
Sorry wrong OTP...
Enter the correct OTP which you recieved
Press 1 to re enter the otp
Press 2 to exit
1
Enter psswd of jobseek
34567
Welcome
```

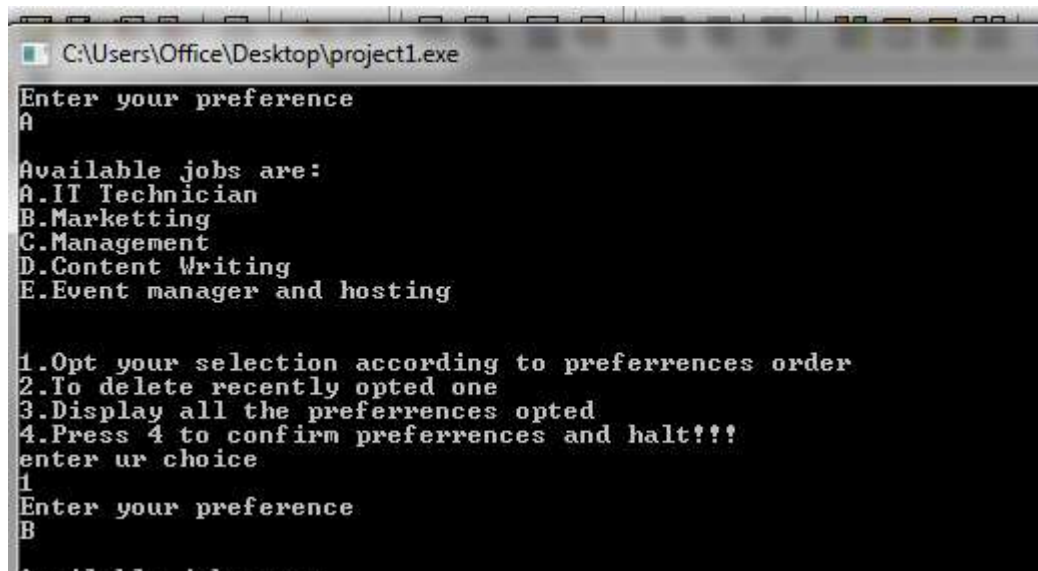
## JOB PREFERENCES BY STACK



```
Enter your preference
C
Available jobs are:
A.IT Technician
B.Marketting
C.Management
D.Content Writing
E.Event manager and hosting

1.Opt your selection according to preferences order
2.To delete recently opted one
3.Display all the preferences opted
4.Press 4 to confirm preferences and halt!!!
enter ur choice
4
You ended the opting of preferences
Selected preferences are:
A B C Final preferences are
A->B->CLet us know who you are
1.Job Seeker
2. Admin
Enter the selection
```

## INSERT IN STACK

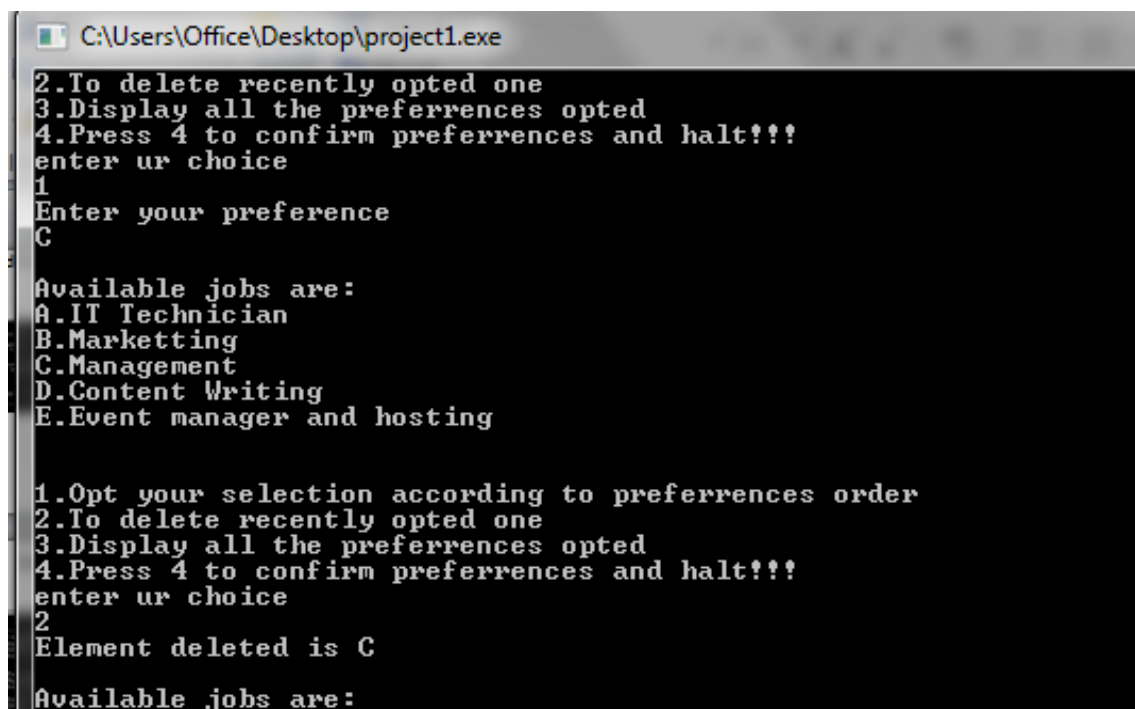


```
C:\Users\Office\Desktop\project1.exe
Enter your preference
A

Available jobs are:
A.IT Technician
B.Marketting
C.Management
D.Content Writing
E.Event manager and hosting

1.Opt your selection according to preferences order
2.To delete recently opted one
3.Display all the preferences opted
4.Press 4 to confirm preferences and halt!!!
enter ur choice
1
Enter your preference
B
Available jobs are:
```

## DELETE RECENTLY OPTED PREFERENCE



```
C:\Users\Office\Desktop\project1.exe
2.To delete recently opted one
3.Display all the preferences opted
4.Press 4 to confirm preferences and halt!!!
enter ur choice
1
Enter your preference
C

Available jobs are:
A.IT Technician
B.Marketting
C.Management
D.Content Writing
E.Event manager and hosting

1.Opt your selection according to preferences order
2.To delete recently opted one
3.Display all the preferences opted
4.Press 4 to confirm preferences and halt!!!
enter ur choice
2
Element deleted is C
Available jobs are:
```

Admin sees the job preference.

## QUEUE CONCEPT

```
1.View profile
2.Send OTP to job seek for preferences
3.See preferences by applicant to which otp sent recently
Enter your choice
3
Enter the reg no., whose job preferences you want to view
TERMS AND CONDITIONS: To which you have sent otp only
1

1.See the job preferences
2.pop and allot jobs
3.exit

Enter ur choice1
A B C are Job preferences

1.See the job preferences
2.pop and allot jobs
3.exit

Enter ur choice2
Allot job
A
A job is allotted to 1
```

Next job allotted.. If job already allotted is preferred

```
3
Enter the reg no., whose job preferences you want to view
TERMS AND CONDITIONS: To which you have sent otp only
2

1.See the job preferences
2.pop and allot jobs
3.exit

Enter ur choice1
A C B are Job preferences
AA already allotted... So don't allot

A C B
1.See the job preferences
2.pop and allot jobs
3.exit

Enter ur choice2
Allot job
C
C job is allotted to 2

1.View profile
2.Send OTP to job seek for preferences
```

Thus preferred jobs are allotted.

- Once after the span of that job... New jobs are released!!!
- Again permissions and registrations.

## **Title 5:**

### **Conclusion:**

This project is helpful to create jobs by developing a website and allotting jobs for women. It also helps women to show enthusiasm to do job and it is especially to build confidence in them. Based on the roles depicted in the project we will further try to implement and use it for creating the front end and backend interfaces respectively.

Thus, we would raise a startup by the concept depicted in the project further and empower women to follow their passions.