

JAVA SWING BASED - InternHub - SQL CONNECTIVITY USING JDBC

A

Report

*Submitted in partial fulfillment of the
Requirements for the award of the Degree of*

BACHELOR OF ENGINEERING IN INFORMATION TECHNOLOGY By

N Taruni <1602-19-737-120>

Under the guidance of Ms B. Leelavathy



**Department of Information Technology
Vasavi College of Engineering (Autonomous)
(Affiliated to Osmania University)
Ibrahimbagh, Hyderabad-31**

2020-2021

BONAFIDE CERTIFICATE

This is to certify that this project report titled
‘InternHub’
is a project work of **N Taruni** bearing
roll no. 1602-19-737-120 who carried out this
project under my supervision in the IV semester
for the academic year 2020- 2021

Signature
External Examiner

Signature
Internal Examiner

ABSTRACT

InternHub is a console-based project that connects students looking for internships with project managers from various companies. Every user, either student or project manager has to login or register first. The students can surf through the internships available and apply for the internships they are interested in. Project managers post about openings for interns and select candidates based on their skills.

Requirement Analysis

List of Tables:

- ProjectManagers
- ManagerProfile
- Internships
- Students
- StudentProfile
- InternshipsApplied

List of Attributes with their Domain Types:

ProjectManagers

- Manager_id varchar2(15)
- FirstName varchar2(20)
- LastName varchar2(15)

ManagerProfile

- Manager_id varchar2(15)
- CompanyName varchar2(20)
- Designation varchar2(20)
- PhoneNumber number(10)

Internships

- Manager_id varchar2(15)
- Internship_id number(5)
- Position varchar2(20)
- Salary number(8)
- Location varchar2(15)
- Skills_required varchar2(25)

Students

- Student_id varchar2(15)
- FirstName varchar2(20)
- LastName varchar2(15)

StudentProfile

- Student_id varchar2(15)
- DOB date
- CollegeName varchar2(25)
- Skills varchar2(50)
- PhoneNumber number(10)
- CGPA number(5)

InternshipsApplied

- Student_id varchar2(15)
- Internship_id number(5)
- SelectedYesNo varchar2(3)

AIM AND PRIORITY OF THE PROJECT

To create a **Java GUI-based** desktop application that connects students looking for internships with project managers looking for interns. It takes values like student name, username, DOB, College name, etc through forms which are then updated in the database using JDBC connectivity.

ARCHITECTURE AND TECHNOLOGY

Software used:

Java Eclipse, Oracle 11g Database, Java SE version 13, SQL*Plus.

Java SWING:

Java SWING is a GUI widget toolkit for Java. It is part of Oracle's Java Foundation Classes (JFC) - an API for providing a graphical user interface (GUI) for Java programs.

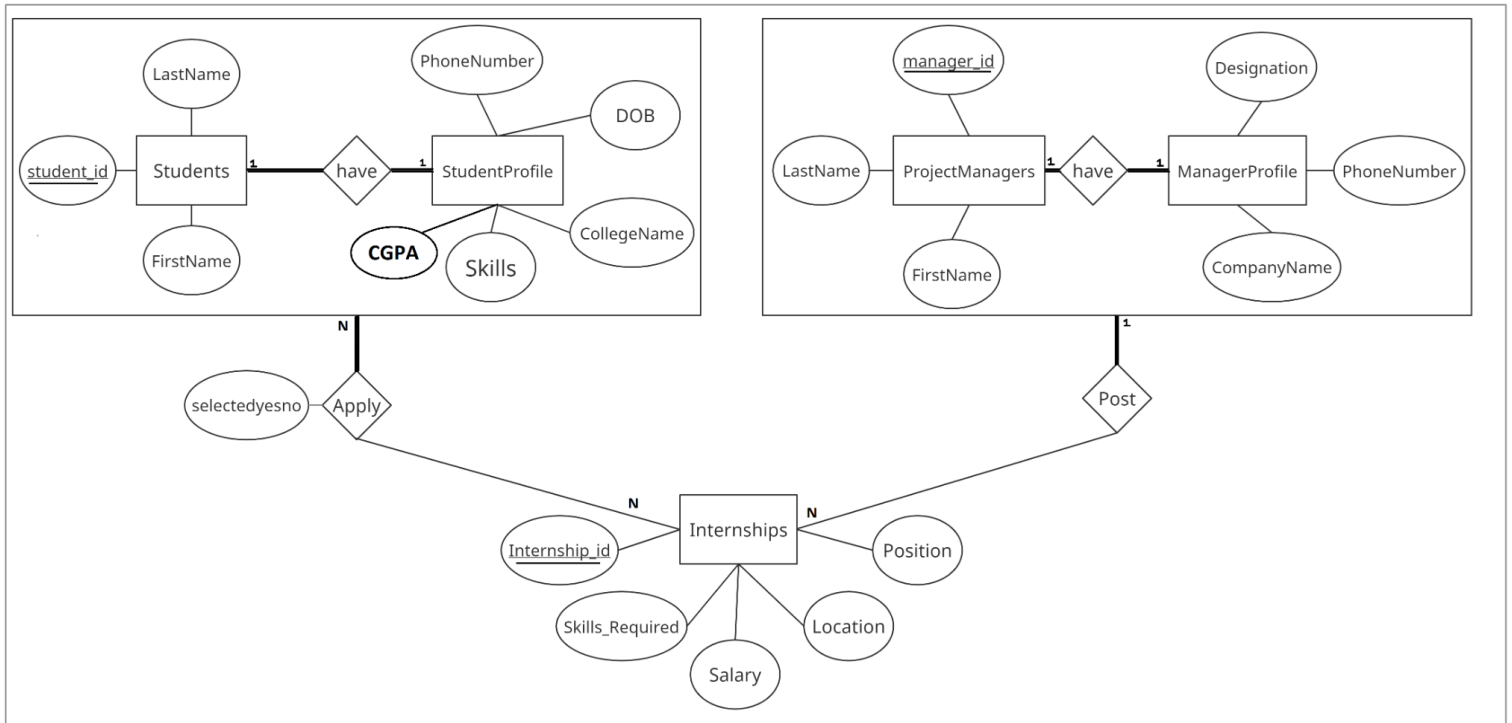
Swing was developed to provide a more sophisticated set of GUI components than the earlier AWT. Swing provides a look and feel that emulates the look and feel of several platforms, and also supports a pluggable look and feel that allows applications to have a look and feel unrelated to the underlying platform. It has more powerful and flexible components than AWT. In addition to familiar components such as buttons, check boxes and labels, Swing provides several advanced components such as tabbed panel, scroll panes, trees, tables, and lists.

SQL:

Structure Query Language(SQL) is a database query language used for storing and managing data in **Relational** DBMS. SQL was the first commercial language introduced for E.F Codd's Relational model of database. Today almost all RDBMS (MySQL, Oracle, Infomix, Sybase, MS Access) use **SQL** as the standard database query language. SQL is used to perform all types of data operations in RDBMS.

DESIGN

Entity Relationship Diagram



DATABASE DESIGN

Mapping Cardinality and Participation Constraints

Every user registered as a student has to have one (only one) student profile mandatorily in order to apply for internships. Therefore, this is a total participation and one-to-one relationship.

Similarly, every user registered as a project manager has to have one (only one) manager profile mandatorily in order to be able to post the internship openings. Therefore, this is a total participation and one-to-one relationship.

A student can apply to any number of different internships he/she wants and at the same time one internship can be applied to by many students, therefore, this is a many-to-many (or N-N) relationship.

Also, every student applies to at least one internship, because that is the reason he/she registered in the first place. Therefore, it is a total participation here from the students' side.

Similarly, a project manager can post many internship openings, therefore, this is a one-to-many (or 1-N) relationship.

Also, every manager posts at least one internship, because that is the purpose of this application, therefore, it is a total participation from the project managers' side.

DDL Operations

```
1. create table projectmanagers(  
  manager_id varchar2(15),  
  firstname varchar2(20) NOT NULL,  
  lastname varchar2(15),  
  CONSTRAINT pkey_projectmanagers_mid PRIMARY KEY (manager_id));
```

```
SQL> create table projectmanagers(  
  2  manager_id varchar2(15),  
  3  firstname varchar2(20) NOT NULL,  
  4  lastname varchar2(15),  
  5  CONSTRAINT pkey_projectmanagers_mid PRIMARY KEY (manager_id));  
  
Table created.
```

```
SQL> desc projectmanagers;
Name                               Null?    Type
-----
MANAGER_ID                         NOT NULL VARCHAR2(15)
FIRSTNAME                         NOT NULL VARCHAR2(20)
LASTNAME                           VARCHAR2(15)
```

2. create table managerprofile(
 manager_id varchar2(15),
 companyname varchar2(20) NOT NULL,
 designation varchar2(30) NOT NULL,
 phonenumber number(10) NOT NULL,
 CONSTRAINT fkey_managerprofile_mid FOREIGN KEY (manager_id)
 REFERENCES projectmanagers(manager_id) ON DELETE CASCADE,
 CONSTRAINT pkey_managerprofile_mid PRIMARY KEY (manager_id));

```
SQL> create table managerprofile(
  2  manager_id varchar2(15),
  3  companyname varchar2(20) NOT NULL,
  4  designation varchar2(20) NOT NULL,
  5  phonenumber number(10) NOT NULL,
  6  CONSTRAINT fkey_managerprofile_mid FOREIGN KEY (manager_id) REFERENCES
projectmanagers(manager_id) ON DELETE CASCADE);

Table created.
```

```
SQL> alter table managerprofile modify(designation varchar2(30));

Table altered.
```

```
SQL> alter table managerprofile add CONSTRAINT pkey_managerprofile_mid
PRIMARY KEY (manager_id);

Table altered.
```

```
SQL> desc managerprofile;
Name                               Null?    Type
-----
MANAGER_ID                         NOT NULL VARCHAR2(15)
COMPANYNAME                       NOT NULL VARCHAR2(20)
DESIGNATION                       NOT NULL VARCHAR2(30)
PHONENUMBER                       NOT NULL NUMBER(10)
```



```

3. create table internships(
manager_id varchar2(15),
internship_id number(5),
position varchar2(30) NOT NULL,
salary number(8),
location varchar2(15) NOT NULL,
skills_required varchar2(50) NOT NULL,
CONSTRAINT fkey_internships_mid FOREIGN KEY (manager_id)
REFERENCES projectmanagers(manager_id) ON DELETE CASCADE,
CONSTRAINT pkey_internships_iid PRIMARY KEY (internship_id));

```

```

SQL> create table internships(
  2  manager_id varchar2(15),
  3  internship_id number(5),
  4  position varchar2(2) NOT NULL,
  5  salary number(8),
  6  location varchar2(15) NOT NULL,
  7  CONSTRAINT fkey_internships_mid FOREIGN KEY (manager_id) REFERENCES
projectmanagers(manager_id) ON DELETE CASCADE,
  8  CONSTRAINT pkey_internships_iid PRIMARY KEY (internship_id));

Table created.

```

```

SQL> alter table internships add(skills_required varchar2(50));

```

```

Table altered.

```

```

SQL> alter table internships modify(position varchar2(30));

```

```

Table altered.

```

```

SQL>

```

```

SQL> desc internships;

```

Name	Null?	Type
-----	-----	-----
MANAGER_ID		VARCHAR2(15)
INTERNSHIP_ID	NOT NULL	NUMBER(5)
POSITION	NOT NULL	VARCHAR2(30)
SALARY		NUMBER(8)
LOCATION	NOT NULL	VARCHAR2(15)
SKILLS_REQUIRED	NOT NULL	VARCHAR2(50)

4. create table students(
 student_id varchar2(15),
 firstname varchar2(20) NOT NULL,
 lastname varchar2(15),
 CONSTRAINT pkey_students_sid PRIMARY KEY (student_id));

```
SQL> create table students(
  2  student_id varchar2(15),
  3  firstname varchar2(20) NOT NULL,
  4  lastname varchar2(15),
  5  CONSTRAINT pkey_students_sid PRIMARY KEY (student_id));

Table created.
```

```
SQL> desc students;
Name                                Null?    Type
-----
STUDENT_ID                          NOT NULL VARCHAR2(15)
FIRSTNAME                           NOT NULL VARCHAR2(20)
LASTNAME                             VARCHAR2(15)
```

5. create table studentprofile(
 student_id varchar2(15),
 dob date NOT NULL,
 collegename varchar2(25) NOT NULL,
 skills varchar2(50) NOT NULL,
 phonenumber number(10) NOT NULL,
 cgpa number(5) NOT NULL,
 CONSTRAINT fkey_studentprofile_sid FOREIGN KEY (student_id)
 REFERENCES students(student_id) ON DELETE CASCADE
 CONSTRAINT pkey_studentprofile_sid PRIMARY KEY (student_id));

```
SQL> create table studentprofile(
  2  student_id varchar2(15),
  3  dob date NOT NULL,
  4  collegename varchar2(25) NOT NULL,
  5  skills varchar2(50) NOT NULL,
  6  phonenumber number(10) NOT NULL,
  7  cgpa number(5) NOT NULL,
  8  CONSTRAINT fkey_studentprofile_sid FOREIGN KEY (student_id) REFERENCES
students(student_id) ON DELETE CASCADE);
```

Table created.

```
SQL> alter table studentprofile add CONSTRAINT pkey_studentprofile_sid
PRIMARY KEY (student_id);
```

Table altered.

```
SQL> desc studentprofile;
```

Name	Null?	Type
STUDENT_ID	NOT NULL	VARCHAR2(15)
DOB	NOT NULL	DATE
COLLEGENAME	NOT NULL	VARCHAR2(25)
SKILLS	NOT NULL	VARCHAR2(50)
PHONENUMBER	NOT NULL	NUMBER(10)
CGPA	NOT NULL	NUMBER(5)

6. create table internshipsapplied(
 student_id varchar2(15),
 internship_id number(5),
 selectedyesno varchar2(3) NOT NULL,
 CONSTRAINT fkey_internshipsapplied_sid FOREIGN KEY (student_id)
 REFERENCES students(student_id) ON DELETE CASCADE,
 CONSTRAINT fkey_internshipsapplied_iid FOREIGN KEY (internship_id)
 REFERENCES internships(internship_id) ON DELETE CASCADE,
 CONSTRAINT pkey_sid_iid PRIMARY KEY (student_id, internship_id));

```
SQL> create table internshipsapplied(
  2  student_id varchar2(15),
  3  internship_id number(5),
  4  selectedyesno varchar2(3) NOT NULL,
  5  CONSTRAINT fkey_internshipsapplied_sid FOREIGN KEY (student_id) REFERENCES students(student_id) ON DELETE
CASCADE,
  6  CONSTRAINT fkey_internshipsapplied_iid FOREIGN KEY (internship_id) REFERENCES internships(internship_id)
ON DELETE CASCADE,
  7  CONSTRAINT pkey_sid_iid PRIMARY KEY (student_id, internship_id));
```

Table created.

```
SQL> desc internshipsapplied;
  Name                               Null?     Type
-----
STUDENT_ID                          NOT NULL  VARCHAR2(15)
INTERNSHIP_ID                       NOT NULL  NUMBER(5)
SELECTEDYESNO                       NOT NULL  VARCHAR2(3)
```

DML Operations

1. insert into projectmanagers values('&manager_id', '&firstname', '&lastname');

```
SQL> insert into projectmanagers values('&manager_id', '&firstname', '&lastname');
Enter value for manager_id: john123
Enter value for firstname: john
Enter value for lastname: ambrose
old 1: insert into projectmanagers values('&manager_id', '&firstname', '&lastname')
new 1: insert into projectmanagers values('john123', 'john', 'ambrose')

1 row created.
```

```
SQL> /
Enter value for manager_id: da_vid
Enter value for firstname: david
Enter value for lastname: rose
old 1: insert into projectmanagers values('&manager_id', '&firstname', '&lastname')
new 1: insert into projectmanagers values('da_vid', 'david', 'rose')

1 row created.
```

```
SQL> /
Enter value for manager_id: joce123
Enter value for firstname: jocelyn
Enter value for lastname:
old 1: insert into projectmanagers values('&manager_id', '&firstname', '&lastname')
new 1: insert into projectmanagers values('joce123', 'jocelyn', '')

1 row created.
```

```
SQL> /
Enter value for manager_id: kat4all
Enter value for firstname: kat
Enter value for lastname: graham
old 1: insert into projectmanagers values('&manager_id', '&firstname', '&lastname')
new 1: insert into projectmanagers values('kat4all', 'kat', 'graham')

1 row created.
```

```
SQL> /
Enter value for manager_id: harryp
Enter value for firstname: harry
Enter value for lastname: potter
old 1: insert into projectmanagers values('&manager_id', '&firstname', '&lastname')
new 1: insert into projectmanagers values('harryp', 'harry', 'potter')

1 row created.
```

```
SQL> /
Enter value for manager_id: hlucy
Enter value for firstname: lucy
Enter value for lastname: hale
old 1: insert into projectmanagers values('&manager_id', '&firstname', '&lastname')
new 1: insert into projectmanagers values('hlucy', 'lucy', 'hale')

1 row created.
```

```
SQL> /
Enter value for manager_id: b_bennet
Enter value for firstname: bonnie
Enter value for lastname: bennet
old 1: insert into projectmanagers values('&manager_id', '&firstname', '&lastname')
new 1: insert into projectmanagers values('b_bennet', 'bonnie', 'bennet')

1 row created.
```

```
SQL> select * from projectmanagers;
```

MANAGER_ID	FIRSTNAME	LASTNAME
john123	john	ambrose
da_vid	david	rose
joce123	jocelyn	
kat4all	kat	graham
harryp	harry	potter
hlucy	lucy	hale
b_bennet	bonnie	bennet

```
7 rows selected.
```

2. insert into managerprofile values('&manager_id', '&companyname', '&designation', &phonenumber);

```

SQL> insert into managerprofile values('&manager_id', '&companyname',
'&designation', &phonenumbers);
Enter value for manager_id: john123
Enter value for companyname: warner bros
Enter value for designation: assistant hr
Enter value for phonenumbers: 123456789
old 1: insert into managerprofile values('&manager_id', '&companyname',
'&designation', &phonenumbers)
new 1: insert into managerprofile values('john123', 'warner bros',
'assistant hr', 123456789)

1 row created.

```

```

SQL> /
Enter value for manager_id: da_vid
Enter value for companyname: fission labs
Enter value for designation: big data lead
Enter value for phonenumbers: 2345678901
old 1: insert into managerprofile values('&manager_id', '&companyname',
'&designation', &phonenumbers)
new 1: insert into managerprofile values('da_vid', 'fission labs',
'big data lead', 2345678901)

1 row created.

```

```

SQL> /
Enter value for manager_id: joce123
Enter value for companyname: AtoZ Store
Enter value for designation: marketing manager
Enter value for phonenumbers: 3456789120
old 1: insert into managerprofile values('&manager_id', '&companyname', '&designation',
&phonenumbers)
new 1: insert into managerprofile values('joce123', 'AtoZ Store', 'marketing manager',
3456789120)

1 row created.

```

```

SQL> /
Enter value for manager_id: kat4all
Enter value for companyname: facebook
Enter value for designation: media lead
Enter value for phonenumbers: 1357924680
old 1: insert into managerprofile values('&manager_id', '&companyname', '&designa
tion', &phonenumbers)
new 1: insert into managerprofile values('kat4all', 'facebook', 'media lead
', 1357924680)

1 row created.

```

```
SQL> /
Enter value for manager_id: harryp
Enter value for companyname: bookworld
Enter value for designation: manager
Enter value for phonenum: 2468013579
old 1: insert into managerprofile values('&manager_id', '&companyname', '&d
esignation', &phonenum)
new 1: insert into managerprofile values('harryp', 'bookworld', 'manager',
2468013579)

1 row created.
```

```
SQL> /
Enter value for manager_id: hlucy
Enter value for companyname: scarlet magazine
Enter value for designation: ad sales lead
Enter value for phonenum: 1111122222
old 1: insert into managerprofile values('&manager_id', '&companyname'
, '&designation', &phonenum)
new 1: insert into managerprofile values('hlucy', 'scarlet magazine',
'ad sales lead', 1111122222)

1 row created.
```

```
SQL> /
Enter value for manager_id: b_bennet
Enter value for companyname: google
Enter value for designation: QandA Lead
Enter value for phonenum: 2222233333
old 1: insert into managerprofile values('&manager_id', '&companyname'
, '&designation', &phonenum)
new 1: insert into managerprofile values('b_bennet', 'google', 'QandA
Lead', 2222233333)

1 row created.
```

```
SQL> select * from managerprofile;
```

MANAGER_ID	COMPANYNAME	DESIGNATION	PHONENUMBER
john123	warner bros	assistant hr	123456789
da_vid	fission labs	big data lead	2345678901
jocel23	AtoZ Store	marketing manager	3456789120
kat4all	facebook	media lead	1357924680
harryp	bookworld	manager	2468013579
hlucy	scarlet magazine	ad sales lead	1111122222
b_bennet	google	QandA Lead	2222233333

```
7 rows selected.
```

3. insert into internships values('&manager_id', &internship_id, '&position', &salary, '&location', '&skills_required');

```
SQL> insert into internships values('&manager_id', &internship_id, '&position',
, &salary, '&location', '&skills_required');
Enter value for manager_id: john123
Enter value for internship_id: 101
Enter value for position: assistant director
Enter value for salary: 8000
Enter value for location: mumbai
Enter value for skills_required: videography, photography
old 1: insert into internships values('&manager_id', &internship_id, '&posit
ion', &salary, '&location', '&skills_required')
new 1: insert into internships values('john123', 101, 'assistant director',
8000, 'mumbai', 'videography, photography')

1 row created.
```

```
SQL> /
Enter value for manager_id: da_vid
Enter value for internship_id: 102
Enter value for position: data analyst
Enter value for salary: 12000
Enter value for location: banglore
Enter value for skills_required: data science, python
old 1: insert into internships values('&manager_id', &internship_id, '&posit
ion', &salary, '&location', '&skills_required')
new 1: insert into internships values('da_vid', 102, 'data analyst', 12000,
'banglore', 'data science, python')

1 row created.
```

```
SQL> /
Enter value for manager_id: joce123
Enter value for internship_id: 103
Enter value for position: salesperson
Enter value for salary: 4000
Enter value for location: gujarat
Enter value for skills_required: communication, patience
old 1: insert into internships values('&manager_id', &internship_id, '&posit
ion', &salary, '&location', '&skills_required')
new 1: insert into internships values('joce123', 103, 'salesperson', 4000, '
gujarat', 'communication, patience')

1 row created.
```



```

SQL> /
Enter value for manager_id: kat4all
Enter value for internship_id: 109
Enter value for position: content writer
Enter value for salary: 7000
Enter value for location: hyderabad
Enter value for skills_required: communication, writing skills
old 1: insert into internships values('&manager_id', &internship_id, '&posit
ion', &salary, '&location', '&skills_required')
new 1: insert into internships values('kat4all', 109, 'content writer', 7000
, 'hyderabad', 'communication, writing skills')

1 row created.

```

```

SQL> /
Enter value for manager_id: harryp
Enter value for internship_id: 104
Enter value for position: receptionist
Enter value for salary: 6000
Enter value for location: kolkata
Enter value for skills_required: communication skills
old 1: insert into internships values('&manager_id', &internship_id, '&posit
ion', &salary, '&location', '&skills_required')
new 1: insert into internships values('harryp', 104, 'receptionist', 6000, '
kolkata', 'communication skills')

1 row created.

```

```

SQL> /
Enter value for manager_id: hlucy
Enter value for internship_id: 105
Enter value for position: assistant
Enter value for salary: 4000
Enter value for location: elmdale
Enter value for skills_required: passion to learn, patience
old 1: insert into internships values('&manager_id', &internship_id, '&posit
ion', &salary, '&location', '&skills_required')
new 1: insert into internships values('hlucy', 105, 'assistant', 4000, 'elmd
ale', 'passion to learn, patience')

1 row created.

```

```

SQL> /
Enter value for manager_id: b_bennet
Enter value for internship_id: 106
Enter value for position: software testing
Enter value for salary: 9000
Enter value for location: hyderabad
Enter value for skills_required: java, python
old 1: insert into internships values('&manager_id', &internship_id, '&posit
ion', &salary, '&location', '&skills_required')
new 1: insert into internships values('b_bennet', 106, 'software testing', 9
000, 'hyderabad', 'java, python')

1 row created.

```

```
SQL> select * from internships;
```

MANAGER_ID	INTERNSHIP_ID	POSITION	SALARY
john123	101	assistant director	8000
mumbai		videography, photography	
da_vid	102	data analyst	12000
bangalore		data science, python	
joc123	103	salesperson	4000
gujarat		communication, patience	

MANAGER_ID	INTERNSHIP_ID	POSITION	SALARY
kat4all	109	content writer	7000
hyderabad		communication, writing skills	
harryp	104	receptionist	6000
kolkata		communication skills	
hlucy	105	assistant	4000
elmdale		passion to learn, patience	

MANAGER_ID	INTERNSHIP_ID	POSITION	SALARY
b_bennet	106	software testing	9000
hyderabad		java, python	

```
7 rows selected.
```

4. insert into students values('&student_id', '&firstname', '&lastname');

```
SQL> insert into students values('&student_id', '&firstname', '&lastname');
Enter value for student_id: pk123
Enter value for firstname: peter
Enter value for lastname: kavinsky
old 1: insert into students values('&student_id', '&firstname', '&lastname')
new 1: insert into students values('pk123', 'peter', 'kavinsky')
1 row created.
```

```
SQL> /
Enter value for student_id: laraj
Enter value for firstname: lara
Enter value for lastname: jean
old 1: insert into students values('&student_id', '&firstname',
'&lastname')
new 1: insert into students values('laraj', 'lara', 'jean')

1 row created.
```

```
SQL> /
Enter value for student_id: knowitall
Enter value for firstname: hermione
Enter value for lastname: granger
old 1: insert into students values('&student_id', '&firstname',
'&lastname')
new 1: insert into students values('knowitall', 'hermione', 'granger')

1 row created.
```

```
SQL> /
Enter value for student_id: parth4u
Enter value for firstname: parth
Enter value for lastname: samthaan
old 1: insert into students values('&student_id', '&firstname', '&
lastname')
new 1: insert into students values('parth4u', 'parth', 'samthaan')

1 row created.
```

```
SQL> /
Enter value for student_id: emmaw
Enter value for firstname: emma
Enter value for lastname: watson
old 1: insert into students values('&student_id', '&firstname', '&
lastname')
new 1: insert into students values('emmaw', 'emma', 'watson')

1 row created.
```

```
SQL> select * from students;
```

STUDENT_ID	FIRSTNAME	LASTNAME
pk123	peter	kavinsky
laraj	lara	jean
knowitall	hermione	granger
parth4u	parth	samthaan
emmaw	emma	watson

5. insert into studentprofile values('&student_id', '&dob', '&collegename', '&skills', &phonenum, &cgpa);

```
SQL> insert into studentprofile values('&student_id', '&dob', '&collegename', '&skills', &phonenum, &cgpa);
Enter value for student_id: pk123
Enter value for dob: 01-jan-2001
Enter value for collegename: yale university
Enter value for skills: photography, videography
Enter value for phonenum: 1231231231
Enter value for cgpa: 8.7
old 1: insert into studentprofile values('&student_id', '&dob', '&collegename', '&skills', &phonenum, &cgpa)
new 1: insert into studentprofile values('pk123', '01-jan-2001', 'yale university', 'photography, videography', 1231231231, 8.7)

1 row created.
```

```
SQL> /
Enter value for student_id: laraj
Enter value for dob: 02-feb-2002
Enter value for collegename: cbit
Enter value for skills: writing skills, communication
Enter value for phonenum: 4564564564
Enter value for cgpa: 8.3
old 1: insert into studentprofile values('&student_id', '&dob', '&collegename', '&skills', &phonenum, &cgpa)
new 1: insert into studentprofile values('laraj', '02-feb-2002', 'cbit', 'writing skills, communication', 4564564564, 8.3)

1 row created.
```

```
SQL> /
Enter value for student_id: knowitall
Enter value for dob: 03-mar-2000
Enter value for collegename: harvard
Enter value for skills: python, datascience
Enter value for phonenum: 3453453453
Enter value for cgpa: 9.6
old 1: insert into studentprofile values('&student_id', '&dob', '&collegename', '&skills', &phonenum, &cgpa)
new 1: insert into studentprofile values('knowitall', '03-mar-2000', 'harvard', 'python, datascience', 3453453453, 9.6)

1 row created.
```

```
SQL> /
Enter value for student_id: parth4u
Enter value for dob: 11-mar-2000
Enter value for collegename: iit roorkee
Enter value for skills: python, java
Enter value for phonenumber: 7897897897
Enter value for cgpa: 9.5
old 1: insert into studentprofile values('&student_id', '&dob', '&collegename', '&sk
ills', &phonenumber, &cgpa)
new 1: insert into studentprofile values('parth4u', '11-mar-2000', 'iit roorkee', 'p
ython, java', 7897897897, 9.5)

1 row created.
```

```
SQL> /
Enter value for student_id: emmaw
Enter value for dob: 04-apr-2002
Enter value for collegename: stanford
Enter value for skills: communication
Enter value for phonenumber: 1234123412
Enter value for cgpa: 8.7
old 1: insert into studentprofile values('&student_id', '&dob', '&collegename', '&sk
ills', &phonenumber, &cgpa)
new 1: insert into studentprofile values('emmaw', '04-apr-2002', 'stanford', 'commun
ication', 1234123412, 8.7)

1 row created.
```

```
SQL> select * from studentprofile;
```

STUDENT_ID	DOB	COLLEGENAME	SKILLS	PHONENUMBER	CGPA
laraj	02-FEB-02	cbit	writing skills, communication	4564564564	8
knowitall	03-MAR-00	harvard	python, datascience	3453453453	10
parth4u	11-MAR-00	iit roorkee	python, java	7897897897	10
emmaw	04-APR-02	stanford	communication	1234123412	9
pk123	01-JAN-01	yale university	photography, videography	1231231231	9

6. insert into internshipsapplied values('&student_id', &internship_id, 'no');

```
SQL> insert into internshipsapplied values('&student_id', &internship_id,
'no');
Enter value for student_id: pk123
Enter value for internship_id: 101
old 1: insert into internshipsapplied values('&student_id', &internship_id, 'no')
new 1: insert into internshipsapplied values('pk123', 101, 'no')

1 row created.
```

```
SQL> /
Enter value for student_id: laraj
Enter value for internship_id: 109
old 1: insert into internshipsapplied values('&student_id', &internship_id, 'no')
new 1: insert into internshipsapplied values('laraj', 109, 'no')

1 row created.
```

```
SQL> /
Enter value for student_id: knowitall
Enter value for internship_id: 102
old 1: insert into internshipsapplied values('&student_id', &internship_id, 'no')
new 1: insert into internshipsapplied values('knowitall', 102, 'no')

1 row created.
```

```
SQL> /
Enter value for student_id: parth4u
Enter value for internship_id: 106
old 1: insert into internshipsapplied values('&student_id', &internship_id, 'no')
new 1: insert into internshipsapplied values('parth4u', 106, 'no')

1 row created.
```

```
SQL> /
Enter value for student_id: emmaw
Enter value for internship_id: 103
old 1: insert into internshipsapplied values('&student_id', &internship_id, 'no')
new 1: insert into internshipsapplied values('emmaw', 103, 'no')

1 row created.
```

```
SQL> /
Enter value for student_id: emmaw
Enter value for internship_id: 104
old 1: insert into internshipsapplied values('&student_id', &internship_
id, 'no')
new 1: insert into internshipsapplied values('emmaw', 104, 'no')

1 row created.
```

```
SQL> select * from internshipsapplied;

STUDENT_ID      INTERNSHIP_ID SEL
-----
pk123           101 no
laraj           109 no
knowitall       102 no
parth4u         106 no
emmaw           103 no
emmaw           104 no

6 rows selected.
```

IMPLEMENTATION

JAVA-SQL Connectivity using JDBC:

Java Database Connectivity (JDBC) is an application programming interface (API) for the programming language Java, which defines how a client may access a database. It is a Java-based data access technology used for Java database connectivity. It is part of the Java Standard Edition platform, from Oracle Corporation. It provides methods to query and update data in a database and is oriented towards relational databases.

The connection to the database can be performed using Java programming (JDBC API) as:

```
package main;
import java.sql.*;

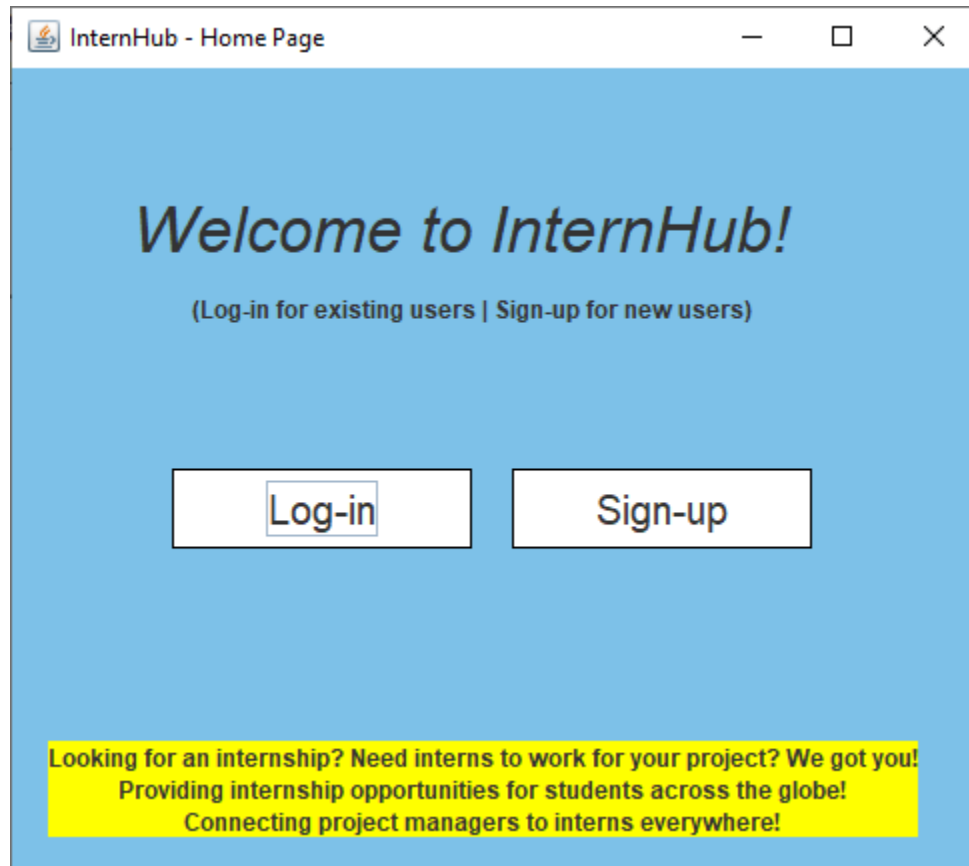
public class ConnectionManager {

    private static String url = "jdbc:oracle:thin:@localhost:1521:xe";
    private static String username = "it19737120";
    private static String password = "123";
    private static Connection con;

    public static Connection getConnection() throws Exception {
        con = DriverManager.getConnection(url, username, password);
        return con;
    }
}
```


Front-end Programs (User Interfaces)

Home Page



```
package UI;

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class HomePage {
    public JFrame frame;
    public JLabel title, prompt, marketing;
    public JButton login, signup;

    public HomePage(JFrame frame) {
        this.frame = frame;
        frame.setTitle("InternHub - Home Page");
        frame.getContentPane().setBackground(new Color(125, 193, 232));
    }
}
```

InternHub

```
title = new JLabel("Welcome to InternHub!");
title.setBounds(60, 50, 350, 60);
Font titleFont = title.getFont();
int stringWidth =
title.getFontMetrics(titleFont).stringWidth("Welcome to InternHub!");
double widthRatio = (double)350 / (double)stringWidth;
int newFontSize = (int)(titleFont.getSize() * widthRatio);
title.setFont(new Font("Welcome to InternHub!", Font.ITALIC,
Math.min(newFontSize, 60)));
frame.add(title);

prompt = new JLabel("(Log-in for existing users | Sign-up for new
users)");
prompt.setBounds(90, 100, 350, 40);
frame.add(prompt);

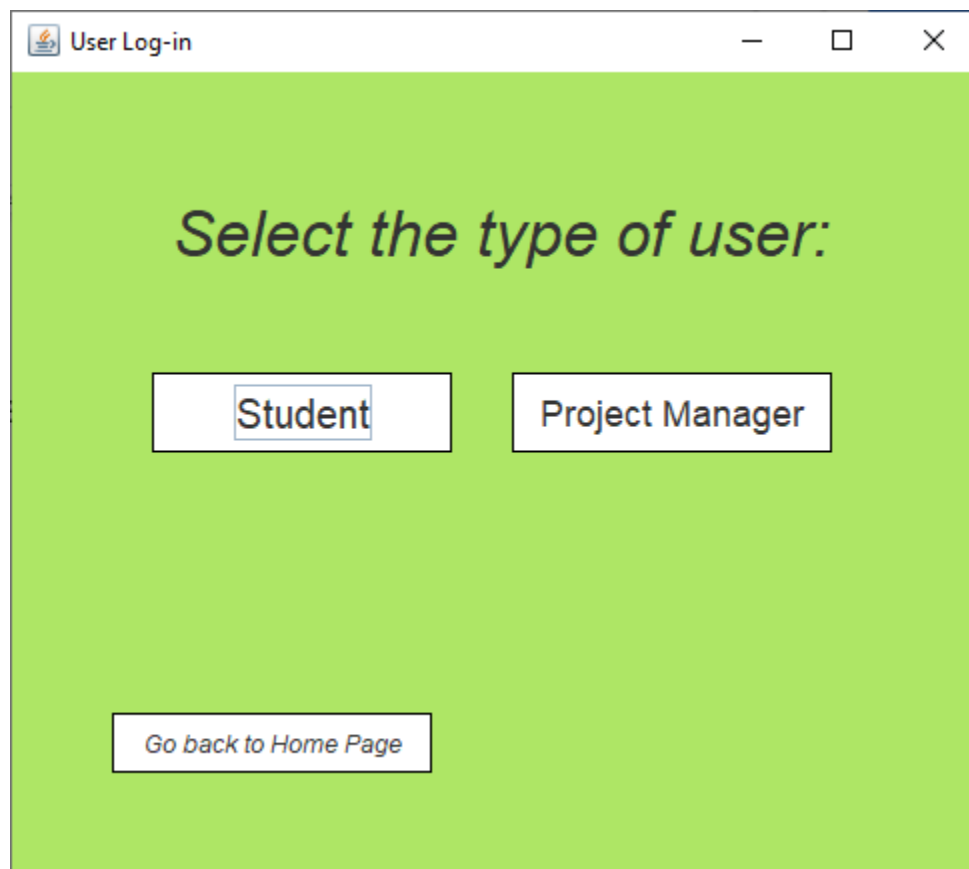
login = new JButton("Log-in");
login.setBounds(80,200,150, 40);
login.setFont(new Font("Log-in", Font.PLAIN, 20));
login.setBackground(Color.white);
login.setBorder(BorderFactory.createLineBorder(Color.BLACK));
frame.add(login);
login.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        frame.getContentPane().removeAll();
        new UserLoginSignup(frame, "login");
    }
});

signup = new JButton("Sign-up");
signup.setBounds(250,200,150, 40);
signup.setFont(new Font("Sign-up", Font.PLAIN, 20));
signup.setBackground(Color.white);
signup.setBorder(BorderFactory.createLineBorder(Color.BLACK));
frame.add(signup);
signup.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        frame.getContentPane().removeAll();
        new UserLoginSignup(frame, "signup" );
    }
});
```

InternHub

```
marketing = new JLabel( "<html><p style=\"text-align:center;\"  
bgcolor=\"yellow\">Looking for an internship? Need interns to work for  
your project? We got you! <br>Providing internship opportunities for  
students across the globe!<br> Connecting project managers to interns  
everywhere!</p></html>" );  
  
marketing.setBounds(18, 320, 600, 80);  
frame.add(marketing);  
  
frame.setSize(500, 440);  
frame.setLayout(null);  
frame.setVisible(true);  
}  
}
```

Type of User - student or project manager



```
package UI;  
import javax.swing.*.*;  
import java.awt.*.*;  
import java.awt.event.*;
```

```

public class UserLoginSignup {
    public JFrame frame;
    public JLabel prompt;
    public JButton student, manager, back;

    public UserLoginSignup(JFrame frame, String userType) {
        this.frame = frame;
        if (userType.equals("login")) {
            frame.setTitle("User Log-in");
        }
        else {
            frame.setTitle("User Sign-up");
        }
        frame.getContentPane().setBackground(new Color(174, 230, 101));

        prompt = new JLabel("Select the type of user:");
        prompt.setBounds(80, 50, 350, 60);
        Font promptFont = prompt.getFont();
        int stringWidth =
prompt.getFontMetrics(promptFont).stringWidth("Select the type of user:");
        double widthRatio = (double)350 / (double)stringWidth;
        int newFontSize = (int)(promptFont.getSize() * widthRatio);
        prompt.setFont(new Font("Select the type of user:", Font.ITALIC,
Math.min(newFontSize, 60)));
        frame.add(prompt);

        student = new JButton("Student");
        student.setBounds(70,150,150, 40);
        student.setFont(new Font("Student", Font.PLAIN, 20));
        student.setBackground(Color.white);
        student.setBorder(BorderFactory.createLineBorder(Color.BLACK));
        frame.add(student);
        student.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                frame.getContentPane().removeAll();
                if (userType.equals("login")) {
                    new LoginWithUsername(frame, "student");
                }
                else {
                    new NewUserSignup(frame, "student");
                }
            }
        })
    }
}

```

```

    });

    manager = new JButton("Project Manager");
    manager.setBounds(250,150,160, 40);
    manager.setFont(new Font("Project Manager", Font.PLAIN, 18));
    manager.setBackground(Color.white);
    manager.setBorder(BorderFactory.createLineBorder(Color.BLACK));
    frame.add(manager);
    manager.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            frame.getContentPane().removeAll();
            if (userType.equals("login")) {
                new LoginWithUsername(frame, "manager");
            }
            else {
                new NewUserSignup(frame, "manager");
            }
        }
    });

    back = new JButton("Go back to Home Page");
    back.setBounds(50, 320, 160, 30);
    back.setFont(new Font("Go back to Home Page",Font.ITALIC, 12));
    back.setBackground(Color.white);
    back.setBorder(BorderFactory.createLineBorder(Color.BLACK));
    frame.add(back);
    back.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            frame.getContentPane().removeAll();
            new HomePage(frame);
        }
    });

    frame.setSize(500, 440);
    frame.setLayout(null);
    frame.setVisible(true);
}
}

```

Sign-up page for new user

```
package UI;
```

1602-19-737-120

N.Taruni

InternHub

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
import main.*;

public class NewUserSignup {
    public JFrame frame;
    public JLabel prompt, userNameLabel, firstNameLabel, secondNameLabel;
    public JTextField userName, firstName, secondName;
    public JButton back, submit;
    public String userType;

    public NewUserSignup(JFrame frame, String userType) {
        this.frame = frame;
        this.userType = userType;
        if (userType.equals("student")) {
            frame.setTitle("New Student User");
        }
        else {
            frame.setTitle("New Project Manager User");
        }
        frame.getContentPane().setBackground(new Color(125, 193, 232));

        JLabel prompt = new JLabel("Please Enter Your Details");
        prompt.setBounds(50, 56, 400, 40);
        Font promptFont = prompt.getFont();
        int stringWidth =
prompt.getFontMetrics(promptFont).stringWidth("Please Enter Your
Details");
        double widthRatio = (double)400 / (double)stringWidth;
        int newFontSize = (int)(promptFont.getSize() * widthRatio);
        prompt.setFont(new Font("Please Enter Your Details", Font.ITALIC,
Math.min(newFontSize, 40)));
        frame.add(prompt);

        userNameLabel = new JLabel("User name *");
        userNameLabel.setFont(new Font("User name", Font.PLAIN, 20));
        userNameLabel.setBounds(50, 127, 114, 29);
        frame.add(userNameLabel);

        userName = new JTextField();
        userName.setBorder(BorderFactory.createLineBorder(Color.BLACK));
```

```

userName.setBounds(240, 127, 173, 28);
frame.add(userName);

firstNameLabel = new JLabel("First name *");
firstNameLabel.setFont(new Font("First name", Font.PLAIN, 20));
firstNameLabel.setBounds(50, 197, 114, 29);
frame.add(firstNameLabel);

firstName = new JTextField("");
firstName.setBorder(BorderFactory.createLineBorder(Color.BLACK));
firstName.setBounds(240, 197, 173, 28);
frame.add(firstName);

secondNameLabel = new JLabel("Second name ");
secondNameLabel.setFont(new Font("Second name", Font.PLAIN, 20));
secondNameLabel.setBounds(50, 268, 139, 29);
frame.add(secondNameLabel);

secondName = new JTextField();
secondName.setBorder(BorderFactory.createLineBorder(Color.BLACK));
secondName.setBounds(240, 268, 173, 28);
frame.add(secondName);

submit = new JButton("Submit");
submit.setBorder(BorderFactory.createLineBorder(Color.BLACK));
submit.setBackground(Color.WHITE);
submit.setFont(new Font("Submit", Font.ITALIC, 15));
submit.setBounds(190, 340, 89, 23);
frame.add(submit);
submit.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        validate(userName.getText(), firstName.getText(),
secondName.getText());
    }
});

back = new JButton("Go back to Home Page");
back.setBorder(BorderFactory.createLineBorder(Color.BLACK));
back.setBackground(Color.WHITE);
back.setFont(new Font("Go back to Home Page", Font.ITALIC, 11));
back.setBounds(10, 11, 154, 23);
frame.add(back);
back.addActionListener(new ActionListener() {

```

```

        public void actionPerformed(ActionEvent e) {
            frame.getContentPane().removeAll();
            frame.repaint();
            new HomePage(frame);
        }
    });

    frame.setSize(500, 440);
    frame.setLayout(null);
    frame.setVisible(true);
}

public void validate(String uname, String fname, String lname) {
    String message = "";
    if (uname.length() == 0) {
        message = "Username cannot be empty!\n";
    }
    else {
        Connection con;
        Statement s;
        ResultSet rs;
        try {
            con = ConnectionManager.getConnection();
            s = con.createStatement();
            rs = s.executeQuery("select student_id from students");
            while (rs.next()) {
                if (rs.getString(1).equals(uname)) {
                    message += "The username already exists. Please
choose a different username!\n";
                    break;
                }
            }
            rs = s.executeQuery("select manager_id from
projectmanagers");
            while (rs.next()) {
                if (rs.getString(1).equals(uname)) {
                    message += "The username already exists. Please
choose a different username!\n";
                    break;
                }
            }
            s.close();
            con.close();

```



```

    }

    catch (Exception e1) {
        e1.printStackTrace();
    }

}

if (fname.length() == 0) {
    message = message + "First name cannot be empty!\n";
}
else {
    for(char c : fname.toCharArray()) {
        if (!Character.isAlphabetic(c)) {
            message = message + "First name cannot have digits or
special characters!\n";
            break;
        }
    }
}

if (message.length() != 0) {
    JOptionPane.showMessageDialog(new JFrame(), message, "error",
JOptionPane.ERROR_MESSAGE);
}
else {
    Connection con;
    Statement s;
    try {
        con = DriverManager.getConnection();
        s = con.createStatement();
        frame.getContentPane().removeAll();
        if (userType.equals("student")){
            s.executeQuery("Insert into students values('" + uname
+ "', '" + fname + "', '" + lname + "')");
            new NewStudentProfile(frame, userName.getText());
        }
        else {
            s.executeQuery("Insert into projectmanagers values('"
+ uname + "', '" + fname + "', '" + lname + "')");
            new NewManagerProfile(frame, userName.getText());
        }
        s.executeQuery("commit");
        s.close();
        con.close();
    }
}

```

```

    }

    catch (Exception e1) {
        e1.printStackTrace();
    }
}
}
}

```

Log-in page for existing users

```

package UI;

import javax.swing.*;
import main.ConnectionManager;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;

public class LoginWithUsername {
    public JFrame frame;
    public String typeOfUser;
    public JLabel prompt;
    public JButton submit, back;
    public JTextField username;

    public LoginWithUsername(JFrame frame, String typeOfUser ) {
        this.typeOfUser = typeOfUser;
        this.frame = frame;
        frame.setTitle("Login as a " + typeOfUser + "!");
        frame.getContentPane().setBackground(new Color(125, 193, 232));

        prompt = new JLabel("Enter your user name:");
        prompt.setBounds(50, 56, 400, 40);
        Font promptFont = prompt.getFont();
        int stringWidth =
prompt.getFontMetrics(promptFont).stringWidth("Enter your user name");
        double widthRatio = (double)400 / (double)stringWidth;
        int newFontSize = (int)(promptFont.getSize() * widthRatio);
        prompt.setFont(new Font("Enter your user name", Font.ITALIC,
Math.min(newFontSize, 40)));
        frame.add(prompt);
    }
}

```

```

submit = new JButton("Submit");
submit.setBorder(BorderFactory.createLineBorder(Color.BLACK));
submit.setBackground(Color.WHITE);
submit.setFont(new Font("Submit", Font.ITALIC, 15));
submit.setBounds(190, 240, 89, 23);
frame.add(submit);

submit.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        validate(username.getText());
    }
});

username = new JTextField();
username.setBounds(130, 150, 200, 30);
username.addKeyListener(new KeyListener() {
    public void keyPressed(KeyEvent e) {
        if(e.getKeyCode() == KeyEvent.VK_ENTER) {
            submit.doClick();
        }
    }
    public void keyReleased(KeyEvent e) {}
    public void keyTyped(KeyEvent e) {}
});
frame.add(username);

back = new JButton("Go back to Home Page");
back.setBorder(BorderFactory.createLineBorder(Color.BLACK));
back.setBackground(Color.WHITE);
back.setFont(new Font("Go back to Home Page", Font.ITALIC, 11));
back.setBounds(10, 350, 154, 23);
frame.add(back);
back.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        frame.getContentPane().removeAll();
        frame.repaint();
        new HomePage(frame);
    }
});

frame.setSize(500, 440);
frame.setLayout(null);
frame.setVisible(true);
}

```

```

public void validate(String username) {
    Connection con;
    Statement s;
    ResultSet rs;
    try {
        con = ConnectionManager.getConnection();
        s = con.createStatement();
        boolean flag = false;
        if (typeOfUser.equals("student")) {
            rs = s.executeQuery("select student_id from students");
        }
        else {
            rs = s.executeQuery("select manager_id from
projectmanagers");
        }
        while(rs.next()) {
            String user = rs.getString(1);
            if (user.equals(username)) {
                flag = true;
                break;
            }
        }
        if (flag) {
            frame.getContentPane().removeAll();
            if (typeOfUser.equals("student")) {
                new Student(frame, username);
            }
            else {
                new Manager(frame, username);
            }
        }
        else {
            JOptionPane.showMessageDialog(new JFrame(), "Username
doesnot exist! Please enter a valid username!", "error",
JOptionPane.ERROR_MESSAGE);
        }
        s.close();
        con.close();
    }
    catch (Exception e1) {
        e1.printStackTrace();
    }
}

```

```
}  
}
```

Student profile for a new student user

```
package UI;  
  
import javax.swing.*;  
  
import main.ConnectionManager;  
  
import java.awt.*;  
import java.awt.event.*;  
import java.sql.*;  
  
public class NewStudentProfile {  
    public JFrame frame;  
    public JButton submit, back;  
    public JLabel dobLabel, collegeLabel, skillsLabel, phoneLabel,  
gpaLabel, prompt;  
    public JTextField dob, college, phone, cgpa;  
    public JTextArea skills;  
    public String sid;  
  
    public NewStudentProfile(JFrame frame, String sid) {  
        this.frame = frame;  
        frame.setTitle("New Student Profile");  
        this.sid = sid;  
        frame.getContentPane().setBackground(new Color(174, 230, 101));  
  
        dobLabel = new JLabel("Date of Birth (Ex:05-dec-2001)*");  
        dobLabel.setBounds(50, 127, 200, 29);  
        frame.add(dobLabel);  
  
        dob = new JTextField("dd-mmm-yyyy");  
        dob.setBorder(BorderFactory.createLineBorder(Color.BLACK));  
        dob.setBounds(250, 127, 150, 28);  
        frame.add(dob);  
  
        phoneLabel = new JLabel("Phone Number *");
```

```
phoneLabel.setBounds(50, 167, 114, 29);
frame.add(phoneLabel);

phone = new JTextField();
phone.setBorder(BorderFactory.createLineBorder(Color.BLACK));
phone.setBounds(250, 167, 150, 28);
frame.add(phone);

collegeLabel = new JLabel("College Name *");
collegeLabel.setBounds(50, 207, 114, 29);
frame.add(collegeLabel);

college = new JTextField();
college.setBorder(BorderFactory.createLineBorder(Color.BLACK));
college.setBounds(250, 207, 150, 28);
frame.add(college);

gpaLabel = new JLabel("CGPA *");
gpaLabel.setBounds(50, 247, 114, 29);
frame.add(gpaLabel);

cgpa = new JTextField();
cgpa.setBorder(BorderFactory.createLineBorder(Color.BLACK));
cgpa.setBounds(250, 247, 150, 28);
frame.add(cgpa);

skillsLabel = new JLabel("Skills *");
skillsLabel.setBounds(50, 287, 114, 29);
frame.add(skillsLabel);

skills = new JTextArea();
skills.setBorder(BorderFactory.createLineBorder(Color.BLACK));
skills.setBounds(250, 287, 150, 80);
frame.add(skills);

String name = " name";
Connection con;
Statement s;
ResultSet rs;
try {
    con = ConnectionManager.getConnection();
    s = con.createStatement();
```

```

        rs = s.executeQuery("select firstname from students where
student_id = '" + sid + "'");
        rs.next();
        name = rs.getString(1);
        s.close();
        con.close();
    }
    catch (Exception e1) {
        e1.printStackTrace();
    }

    JLabel prompt = new JLabel("Welcome " + name + ", Please Enter
Your Profile Details");
    prompt.setBounds(50, 50, 400, 35);
    Font promptFont = prompt.getFont();
    int stringWidth =
prompt.getFontMetrics(promptFont).stringWidth("Welcome ____ Please Enter
Your Profile Details");
    double widthRatio = (double) 400 / (double) stringWidth;
    int newFontSize = (int) (promptFont.getSize() * widthRatio);
    prompt.setFont(new Font("Please Enter Your Profile Details",
Font.ITALIC, Math.min(newFontSize, 30)));
    frame.add(prompt);

    submit = new JButton("Submit");
    submit.setBorder(BorderFactory.createLineBorder(Color.BLACK));
    submit.setBackground(Color.WHITE);
    submit.setFont(new Font("Submit", Font.ITALIC, 18));
    submit.setBounds(190, 400, 89, 23);
    frame.add(submit);
    submit.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            validate();
        }
    });

    back = new JButton("Go back to Home Page");
    back.setBorder(BorderFactory.createLineBorder(Color.BLACK));
    back.setBackground(Color.WHITE);
    back.setFont(new Font("Go back to Home Page", Font.ITALIC, 11));
    back.setBounds(10, 11, 154, 23);
    frame.add(back);
    back.addActionListener(new ActionListener() {

```

```

        public void actionPerformed(ActionEvent e) {
            frame.getContentPane().removeAll();
            new HomePage(frame);
        }
    });

    frame.setSize(500, 500);
    frame.setLayout(null);
    frame.setVisible(true);
}

public void validate() {
    String message = "";
    String dob = this.dob.getText(), college = this.college.getText(),
    phone = this.phone.getText(), gpa = this.cgpa.getText(), skills =
    this.skills.getText();
    if (dob.length() == 0 || dob.equals("dd-mmm-yyyy")) {
        message = message + "Date of Birth cannot be empty!\n";
    }
    else if (dob.length() != 11) {
        message += "Enter a valid Date of Birth!\n";
    }

    if (phone.length() == 0) {
        message = message + "Phone number cannot be empty!\n";
    }
    else {
        for(char c : phone.toCharArray()) {
            if (!Character.isDigit(c)) {
                message = message + "Phone number cannot have
alphabets or special characters!\n";
                break;
            }
        }
        if (phone.length() != 10) {
            message += "Phone number must have exactly 10
digits!\n";
        }
    }

    if (college.length() == 0) {
        message = message + "College name cannot be empty!\n";
    }
}

```



```

        else {
            for(char c : college.replaceAll(" ", "").toCharArray()) {
                if (!Character.isAlphabetic(c) ) {
                    message = message + "College name cannot have digits
or special characters!\n";
                    break;
                }
            }
        }

        if (gpa.length() == 0) {
            message = message + "CGPA cannot be empty!\n";
        }
        else {
            try {
                float f = Float.parseFloat(gpa);
                if (f > 10 || f < 0) {
                    message += "CGPA must lie between 0 to 10!\n";
                }
            }
            catch (Exception e) {
                message += "CGPA must be a valid Number!\n";
            }
        }

        if (skills.length() == 0) {
            message = message + "Skills cannot be empty!\n";
        }

        if (message.length() != 0) {
            JOptionPane.showMessageDialog(new JFrame(), message, "error",
JOptionPane.ERROR_MESSAGE);
        }
        else {
            Connection con;
            Statement s;
            try {
                con = ConnectionManager.getConnection();
                s = con.createStatement();
                s.executeQuery("insert into studentprofile values('" + sid
+ "', '" + dob + "', '" + college + "', '" + skills + "', " + phone + ", "
+ gpa + ")");
                s.executeQuery("commit");
            }
        }
    }
}

```

```

        s.close();
        con.close();
    }
    catch (Exception e1) {
        e1.printStackTrace();
    }
    frame.getContentPane().removeAll();
    new Student(frame, sid);
}
}
}

```

Manager profile for a new project manager user

```

package UI;

import javax.swing.*;

import main.ConnectionManager;

import java.awt.*;
import java.awt.event.*;
import java.sql.*;

public class NewManagerProfile {
    public JFrame frame;
    public JButton submit, back;
    public JLabel companyLabel, designationLabel, phoneLabel, prompt;
    public JTextField company, designation, phone;
    public String mid;

    public NewManagerProfile(JFrame frame, String mid) {
        this.frame = frame;
        frame.setTitle("New Project Manager Profile");
        this.mid = mid;

        frame.getContentPane().setBackground(new Color(174, 230, 101));

        companyLabel = new JLabel("Company Name *");
        companyLabel.setBounds(50, 127, 200, 30);
        companyLabel.setFont(new Font("", Font.PLAIN, 20));
        frame.add(companyLabel);
    }
}

```

```

company = new JTextField();
company.setBorder(BorderFactory.createLineBorder(Color.BLACK));
company.setBounds(250, 127, 150, 28);
frame.add(company);

phoneLabel = new JLabel("Phone Number *");
phoneLabel.setBounds(50, 167, 150, 29);
phoneLabel.setFont(new Font("", Font.PLAIN, 20));
frame.add(phoneLabel);

phone = new JTextField();
phone.setBorder(BorderFactory.createLineBorder(Color.BLACK));
phone.setBounds(250, 167, 150, 28);
frame.add(phone);

designationLabel = new JLabel("Designation *");
designationLabel.setBounds(50, 207, 150, 29);
designationLabel.setFont(new Font("", Font.PLAIN, 20));
frame.add(designationLabel);

designation = new JTextField();

designation.setBorder(BorderFactory.createLineBorder(Color.BLACK));
designation.setBounds(250, 207, 150, 28);
frame.add(designation);

String name = " name";
Connection con;
Statement s;
ResultSet rs;
try {
    con = ConnectionManager.getConnection();
    s = con.createStatement();
    rs = s.executeQuery("select firstname from projectmanagers
where manager_id = '" + mid + "'");
    rs.next();
    name = rs.getString(1);
    s.close();
    con.close();
}
catch (Exception e1) {
    e1.printStackTrace();
}

```

```

        JLabel prompt = new JLabel("Welcome " + name + ", Please Enter
Your Profile Details");
        prompt.setBounds(50, 50, 400, 35);
        Font promptFont = prompt.getFont();
        int stringWidth =
prompt.getFontMetrics(promptFont).stringWidth("Welcome " + name + " Please
Enter Your Profile Details");
        double widthRatio = (double) 400 / (double) stringWidth;
        int newFontSize = (int) (promptFont.getSize() * widthRatio);
        prompt.setFont(new Font("Please Enter Your Profile Details",
Font.ITALIC, Math.min(newFontSize, 30)));
        frame.add(prompt);

        submit = new JButton("Submit");
        submit.setBorder(BorderFactory.createLineBorder(Color.BLACK));
        submit.setBackground(Color.WHITE);
        submit.setFont(new Font("Submit", Font.ITALIC, 18));
        submit.setBounds(190, 280, 89, 23);
        frame.add(submit);
        submit.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                validate();
            }
        });

        back = new JButton("Go back to Home Page");
        back.setBorder(BorderFactory.createLineBorder(Color.BLACK));
        back.setBackground(Color.WHITE);
        back.setFont(new Font("Go back to Home Page", Font.ITALIC, 11));
        back.setBounds(10, 11, 154, 23);
        frame.add(back);
        back.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                frame.getContentPane().removeAll();
                new HomePage(frame);
            }
        });

        frame.setSize(500, 400);
        frame.setLayout(null);
        frame.setVisible(true);
    }

```

```

public void validate() {
    String message = "";
    String company = this.company.getText(), phone =
this.phone.getText(), designation = this.designation.getText();
    if (company.length() == 0) {
        message = message + "Company name cannot be empty!\n";
    }
    else {
        for(char c : company.replaceAll(" ", "").toCharArray()) {
            if (!Character.isAlphabetic(c)) {
                message = message + "Company name cannot have digits
or special characters!\n";
                break;
            }
        }
    }

    if (phone.length() == 0) {
        message = message + "Phone number cannot be empty!\n";
    }
    else {
        for(char c : phone.toCharArray()) {
            if (!Character.isDigit(c)) {
                message = message + "Phone number cannot have
alphabets or special characters!\n";
                break;
            }
        }
        if (phone.length() != 10) {
            message += "Phone number must have exactly 10
digits!\n";
        }
    }

    if (designation.length() == 0) {
        message = message + "Company name cannot be empty!\n";
    }
    else {
        for(char c : company.replaceAll(" ", "").toCharArray()) {
            if (!Character.isAlphabetic(c)) {
                message = message + "Company name cannot have digits
or special characters!\n";

```

```

        break;
    }
}

if (message.length() != 0) {
    JOptionPane.showMessageDialog(new JFrame(), message, "error",
JOptionPane.ERROR_MESSAGE);
}
else {
    Connection con;
    Statement s;
    try {
        con = ConnectionManager.getConnection();
        s = con.createStatement();
        s.executeQuery("insert into managerprofile values('" + mid
+ "', '" + company + "', '" + designation + "', '" + phone + ")");
        s.executeQuery("commit");
        s.close();
        con.close();
    }
    catch (Exception e1) {
        e1.printStackTrace();
    }
    frame.getContentPane().removeAll();
    new Manager(frame, mid);
}
}
}

```

User interface for students

```

package UI;

import javax.swing.*;
import javax.swing.table.DefaultTableModel;

import main.ConnectionManager;
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
import java.text.SimpleDateFormat;

```

InternHub

```
public class Student {
    public JFrame frame;
    public String student_id;
    public JMenuBar menuBar;
    public JMenu profileMenu, internshipsApplied, internships;
    public JMenuItem viewProfile, editProfile, viewInternshipsApplied,
deleteInternshipsApplied, viewInternships, applyInternships;
    public JLabel welcome;
    public JButton back;

    public Student(JFrame frame, String username) {

        this.student_id = username;

        String name = " name";
        Connection con;
        Statement s;
        ResultSet rs;
        try {
            con = ConnectionManager.getConnection();
            s = con.createStatement();
            rs = s.executeQuery("select firstname from students where
student_id = '" + username + "'");
            rs.next();
            name = rs.getString(1);
            s.close();
            con.close();
        }
        catch (Exception e1) {
            e1.printStackTrace();
        }

        frame.setTitle("Student: " + name);
        frame.getContentPane().setBackground(new Color(174, 230, 101));

        back = new JButton("Go back to Home Page");
        back.setBorder(BorderFactory.createLineBorder(Color.BLACK));
        back.setBackground(Color.WHITE);
        back.setFont(new Font("Go back to Home Page", Font.ITALIC, 11));
        back.setBounds(20, 320, 154, 23);
        frame.add(back);
        back.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
```

```

        frame.getContentPane().removeAll();
        frame.setJMenuBar(null);
        new HomePage(frame);
    }
});

welcome = new JLabel("Welcome " + name + "!");
welcome.setBounds(60, 120, 400, 40);
Font promptFont = welcome.getFont();
int stringWidth =
welcome.getFontMetrics(promptFont).stringWidth("Welcome namelong!");
double widthRatio = (double)400 / (double)stringWidth;
int newFontSize = (int)(promptFont.getSize() * widthRatio);
welcome.setFont(new Font("Welcome", Font.ITALIC,
Math.min(newFontSize, 40)));
frame.add(welcome);

menuBar = new JMenuBar();
frame.add(menuBar);

profileMenu = new JMenu("Profile");

profileMenu.setBorder(BorderFactory.createLineBorder(Color.BLACK));
menuBar.add(profileMenu);

internshipsApplied = new JMenu("Internships applied by me");

internshipsApplied.setBorder(BorderFactory.createLineBorder(Color.BLACK));
menuBar.add(internshipsApplied);

internships = new JMenu("Internships available");

internships.setBorder(BorderFactory.createLineBorder(Color.BLACK));
menuBar.add(internships);

viewProfile = new JMenuItem("View profile");
viewProfile.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        welcome.setText("");
        frame.getContentPane().removeAll();
        frame.repaint();
        frame.add(back);
        profileView(frame);
    }
});

```



```

    }
    });

    editProfile = new JMenuItem("Edit profile");
    editProfile.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            welcome.setText("");
            frame.getContentPane().removeAll();
            frame.repaint();
            frame.add(back);
            profileEdit(frame);
        }
    });
    profileMenu.add(viewProfile);
    profileMenu.add(editProfile);

    viewInternshipsApplied = new JMenuItem("View internships
applied");
    viewInternshipsApplied.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            welcome.setText("");
            frame.getContentPane().removeAll();
            frame.repaint();
            frame.add(back);
            internshipsAppliedView(frame);
        }
    });
    deleteInternshipsApplied = new JMenuItem("Delete internships
applied");
    deleteInternshipsApplied.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            welcome.setText("");
            frame.getContentPane().removeAll();
            frame.repaint();
            frame.add(back);
            internshipsAppliedDelete(frame);
        }
    });

    internshipsApplied.add(viewInternshipsApplied);
    internshipsApplied.add(deleteInternshipsApplied);

    viewInternships = new JMenuItem("View available internships");

```

```

viewInternships.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        welcome.setText("");
        frame.getContentPane().removeAll();
        frame.repaint();
        frame.add(back);
        internshipsView(frame);
    }
});

applyInternships = new JMenuItem("Apply for new internships");
applyInternships.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        welcome.setText("");
        frame.getContentPane().removeAll();
        frame.repaint();
        frame.add(back);
        internshipsApply(frame);
    }
});
internships.add(viewInternships);
internships.add(applyInternships);

frame.setJMenuBar(menuBar);
frame.setSize(500, 440);
frame.setLayout(null);
frame.setVisible(true);
this.frame = frame;
}

public void profileView(JFrame frame) {
    JLabel unameF, unameV, fnameF, fnameV = null, lnameF, lnameV =
null, dobF, dobV = null, collegeF, collegeV = null, skillsF, skillsV =
null, phoneF, phoneV = null, gpaF, gpaV = null;
    unameF = new JLabel("Username:");
    unameV = new JLabel(student_id);
    fnameF = new JLabel("First name:");
    lnameF = new JLabel("Last name:");
    dobF = new JLabel("Date of Birth:");
    phoneF = new JLabel("Phone number:");
    collegeF = new JLabel("College name:");
    gpaF = new JLabel("CGPA:");
    skillsF = new JLabel("Skills:");

```

```

Connection con;
Statement s;
ResultSet rs;
try {
    con = ConnectionManager.getConnection();
    s = con.createStatement();
    rs = s.executeQuery("select * from students where student_id =
'" + student_id + "'");
    rs.next();
    fnameV = new JLabel(rs.getString(2));
    lnameV = new JLabel(rs.getString(3));
    rs = s.executeQuery("select * from studentprofile where
student_id = '" + student_id + "'");
    if (rs.next()) {
        dobV = new JLabel((new
SimpleDateFormat("dd-MMM-YYYY")).format(rs.getDate(2)));
        collegeV = new JLabel (rs.getString(3));
        skillsV = new JLabel(rs.getString(4));
        phoneV = new JLabel(rs.getString(5));
        gpaV = new JLabel(rs.getString(6));
    }
    else {
        dobV = new JLabel("");
        collegeV = new JLabel ("");
        skillsV = new JLabel("");
        phoneV = new JLabel("");
        gpaV = new JLabel("");
    }

    s.close();
    con.close();
}
catch (Exception e1) {
    e1.printStackTrace();
}

unameF.setBounds(50, 30, 100, 30);
unameF.setFont(new Font("", Font.PLAIN, 20));
unameV.setBounds(250, 30, 100, 30);
unameV.setFont(new Font("", Font.PLAIN, 20));

fnameF.setBounds(50, 60, 100, 30);

```

```

fnameF.setFont(new Font("", Font.PLAIN, 20));
fnameV.setBounds(250, 60, 100, 30);
fnameV.setFont(new Font("", Font.PLAIN, 20));

lnameF.setBounds(50, 90, 100, 30);
lnameF.setFont(new Font("", Font.PLAIN, 20));
lnameV.setBounds(250, 90, 100, 30);
lnameV.setFont(new Font("", Font.PLAIN, 20));

dobF.setBounds(50, 120, 130, 30);
dobF.setFont(new Font("", Font.PLAIN, 20));
dobV.setBounds(250, 120, 140, 30);
dobV.setFont(new Font("", Font.PLAIN, 20));

phoneF.setBounds(50, 150, 150, 30);
phoneF.setFont(new Font("", Font.PLAIN, 20));
phoneV.setBounds(250, 150, 140, 30);
phoneV.setFont(new Font("", Font.PLAIN, 20));

collegeF.setBounds(50, 180, 150, 30);
collegeF.setFont(new Font("", Font.PLAIN, 20));
collegeV.setBounds(250, 180, 140, 30);
collegeV.setFont(new Font("", Font.PLAIN, 20));

gpaF.setBounds(50, 210, 150, 30);
gpaF.setFont(new Font("", Font.PLAIN, 20));
gpaV.setBounds(250, 210, 140, 30);
gpaV.setFont(new Font("", Font.PLAIN, 20));

skillsF.setBounds(50, 240, 150, 30);
skillsF.setFont(new Font("", Font.PLAIN, 20));
skillsV.setBounds(250, 240, 200, 30);
skillsV.setFont(new Font("", Font.PLAIN, 20));

frame.add(unameF);
frame.add(unameV);
frame.add(lnameF);
frame.add(lnameV);
frame.add(fnameF);
frame.add(fnameV);
frame.add(dobF);
frame.add(dobV);
frame.add(phoneF);

```

```

        frame.add(phoneV);
        frame.add(collegeF);
        frame.add(collegeV);
        frame.add(gpaF);
        frame.add(gpaV);
        frame.add(skillsF);
        frame.add(skillsV);

    }

    public void profileEdit(JFrame frame) {
        JButton update = new JButton("Update");
        update.setBounds(200, 320, 154, 23);
        update.setBorder(BorderFactory.createLineBorder(Color.BLACK));
        update.setBackground(Color.WHITE);

        JLabel unameF, unameV, fnameF, lnameF, dobF, collegeF, skillsF,
phoneF, gpaF;
        JTextField fnameV = new JTextField("");
        JTextField lnameV = new JTextField("");
        JTextField dobV = new JTextField("");
        JTextField collegeV = new JTextField("");
        JTextField phoneV = new JTextField("");
        JTextField gpaV = new JTextField("");
        JTextField skillsV = new JTextField("");

        unameF = new JLabel("Username:");
        unameV = new JLabel(student_id);
        fnameF = new JLabel("First name:");
        lnameF = new JLabel("Last name:");
        dobF = new JLabel("Date of Birth:");
        phoneF = new JLabel("Phone number:");
        collegeF = new JLabel("College name:");
        gpaF = new JLabel("CGPA:");
        skillsF = new JLabel("Skills:");

        Connection con;
        Statement s;
        ResultSet rs;
        try {
            con = ConnectionManager.getConnection();
            s = con.createStatement();

```

```

        rs = s.executeQuery("select * from students where student_id =
'" + student_id + "'");
        rs.next();
        fnameV.setText(rs.getString(2));
        lnameV.setText(rs.getString(3));
        rs = s.executeQuery("select * from studentprofile where
student_id = '" + student_id + "'");
        if (rs.next()) {
            dobV.setText((new
SimpleDateFormat("dd-MMM-YYYY")).format(rs.getDate(2)));
            collegeV.setText(rs.getString(3));
            skillsV.setText(rs.getString(4));
            phoneV.setText(rs.getString(5));
            gpaV.setText(rs.getString(6));
        }
        else {
            dobV.setText("dd-mmm-yyyy");
            collegeV.setText("");
            skillsV.setText("");
            phoneV.setText("");
            gpaV.setText("");
        }
        s.close();
        con.close();
    }
    catch (Exception e1) {
        e1.printStackTrace();
    }

    unameF.setBounds(50, 30, 100, 30);
    unameF.setFont(new Font("", Font.PLAIN, 20));
    unameV.setBounds(250, 30, 100, 30);
    unameV.setFont(new Font("", Font.PLAIN, 20));

    fnameF.setBounds(50, 60, 100, 30);
    fnameF.setFont(new Font("", Font.PLAIN, 20));
    fnameV.setBounds(250, 60, 100, 30);
    fnameV.setFont(new Font("", Font.PLAIN, 20));

    lnameF.setBounds(50, 90, 100, 30);
    lnameF.setFont(new Font("", Font.PLAIN, 20));
    lnameV.setBounds(250, 90, 100, 30);
    lnameV.setFont(new Font("", Font.PLAIN, 20));

```

```

dobF.setBounds(50, 120, 130, 30);
dobF.setFont(new Font("", Font.PLAIN, 20));
dobV.setBounds(250, 120, 140, 30);
dobV.setFont(new Font("", Font.PLAIN, 20));

phoneF.setBounds(50, 150, 150, 30);
phoneF.setFont(new Font("", Font.PLAIN, 20));
phoneV.setBounds(250, 150, 140, 30);
phoneV.setFont(new Font("", Font.PLAIN, 20));

collegeF.setBounds(50, 180, 150, 30);
collegeF.setFont(new Font("", Font.PLAIN, 20));
collegeV.setBounds(250, 180, 140, 30);
collegeV.setFont(new Font("", Font.PLAIN, 20));

gpaF.setBounds(50, 210, 150, 30);
gpaF.setFont(new Font("", Font.PLAIN, 20));
gpaV.setBounds(250, 210, 140, 30);
gpaV.setFont(new Font("", Font.PLAIN, 20));

skillsF.setBounds(50, 240, 150, 30);
skillsF.setFont(new Font("", Font.PLAIN, 20));
skillsV.setBounds(250, 240, 200, 30);
skillsV.setFont(new Font("", Font.PLAIN, 20));

update.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        try {
            validateProfile(fnameV.getText(), lnameV.getText(),
dobV.getText(), phoneV.getText(), collegeV.getText(), gpaV.getText(),
skillsV.getText());
        }
        catch (Exception e1) {
            e1.printStackTrace();
        }
    }
});

frame.add(unameF);
frame.add(unameV);
frame.add(lnameF);

```

InternHub

```
        frame.add(lnameV);
        frame.add(fnameF);
        frame.add(fnameV);
        frame.add(dobF);
        frame.add(dobV);
        frame.add(phoneF);
        frame.add(phoneV);
        frame.add(collegeF);
        frame.add(collegeV);
        frame.add(gpaF);
        frame.add(gpaV);
        frame.add(skillsF);
        frame.add(skillsV);
        frame.add(update);

    }

    public void internshipsAppliedView(JFrame frame) {
        String iid, position, salary, status, location;
        String header[] = new String[] { "ID", "Role", "Salary",
"Location", "Selected (yes/no)"};

        JTable table = new JTable();
        DefaultTableModel dtm = new DefaultTableModel(0, 0);
        dtm.setColumnIdentifiers(header);
        table.setModel(dtm);

        Connection con;
        Statement s1, s2;
        ResultSet r1, r2;
        try {
            con = ConnectionManager.getConnection();
            s1 = con.createStatement();
            s2 = con.createStatement();
            r1 = s1.executeQuery("select * from internshipsapplied where
student_id = '" + student_id + "'");
            while (r1.next()) {
                iid = r1.getString(2);
                status = r1.getString(3);
                r2 = s2.executeQuery("select salary, location, position
from internships where internship_id = " + iid + " ");
                r2.next();
                salary = r2.getString(1);
```



```

        location = r2.getString(2);
        position = r2.getString(3);
        dtm.addRow(new Object[] {iid, position, salary, location,
status});
    }
    s1.close();
    s2.close();
    con.close();
}
catch (Exception e1) {
    e1.printStackTrace();
}
table.setBounds(100, 70, 250, 100);
table.getColumnModel().getColumn(0).setPreferredWidth(50);
table.getColumnModel().getColumn(1).setPreferredWidth(180);
table.getColumnModel().getColumn(2).setPreferredWidth(70);
table.getColumnModel().getColumn(3).setPreferredWidth(160);
table.getColumnModel().getColumn(4).setPreferredWidth(160);
table.setFont(new Font("", Font.PLAIN, 15));
table.setAutoResizeMode(JTable.AUTO_RESIZE_OFF);
JScrollPane scroll = new JScrollPane(table);
scroll.setBounds(15, 70, 450, 150);
frame.add(scroll);
}

public void internshipsView(JFrame frame) {
    String iid, position, salary, location, company, mid;
    String header[] = new String[] { "ID", "Role", "Salary",
"Company", "Location"};

    JTable table = new JTable();
    DefaultTableModel dtm = new DefaultTableModel(0, 0);
    dtm.setColumnIdentifiers(header);
    table.setModel(dtm);

    Connection con;
    Statement s1, s2;
    ResultSet r1, r2;
    try {
        con = ConnectionManager.getConnection();
        s1 = con.createStatement();
        s2 = con.createStatement();

```

```

        r1 = s1.executeQuery("select manager_id, internship_id,
position, salary, location from internships where internship_id not in
(select internship_id from internshipsapplied where student_id = '" +
student_id + "') order by internship_id");
        while (r1.next()) {
            mid = r1.getString(1);
            iid = r1.getString(2);
            position = r1.getString(3);
            salary = r1.getString(4);
            location = r1.getString(5);
            r2 = s2.executeQuery("select companyname from
managerprofile where manager_id = '" + mid + "'");
            r2.next();
            company = r2.getString(1);
            dtm.addRow(new Object[] {iid, position, salary, company,
location});
        }
        s1.close();
        s2.close();
        con.close();
    }
    catch (Exception e1) {
        e1.printStackTrace();
    }

    table.setBounds(100, 70, 250, 100);
    table.getColumnModel().getColumn(0).setPreferredWidth(50);
    table.getColumnModel().getColumn(1).setPreferredWidth(180);
    table.getColumnModel().getColumn(2).setPreferredWidth(70);
    table.getColumnModel().getColumn(3).setPreferredWidth(170);
    table.getColumnModel().getColumn(4).setPreferredWidth(160);
    table.setAutoResizeMode(JTable.AUTO_RESIZE_OFF);
    table.setFont(new Font("", Font.PLAIN, 15));
    JScrollPane scroll = new JScrollPane(table);
    scroll.setBounds(15, 70, 450, 150);
    frame.add(scroll);

}

    public void validateProfile(String fname, String lname, String dob,
String phone, String college, String gpa, String skills) {
        String message = "";
        if (fname.length() == 0) {

```

```
        message = message + "First name cannot be empty!\n";
    }
    else {
        for(char c : fname.toCharArray()) {
            if (!Character.isAlphabetic(c)) {
                message = message + "First name cannot have digits or
special characters!\n";
                break;
            }
        }
    }

    for(char c : lname.toCharArray()) {
        if (!Character.isAlphabetic(c)) {
            message = message + "Last name cannot have digits or
special characters!\n";
            break;
        }
    }

    if (dob.length() == 0 || dob.equals("dd-mmm-yyyy")) {
        message = message + "Date of Birth cannot be empty!\n";
    }
    else if (dob.length() != 11 ) {
        message += "Enter a valid Date of Birth!\n";
    }

    if (phone.length() == 0) {
        message = message + "Phone number cannot be empty!\n";
    }
    else {
        for(char c : phone.toCharArray()) {
            if (!Character.isDigit(c)) {
                message = message + "Phone number cannot have
alphabets or special characters!\n";
                break;
            }
        }
        if (phone.length() != 10) {
            message += "Phone number must have exactly 10
digits!\n";
        }
    }
}
```

```

    if (college.length() == 0) {
        message = message + "College name cannot be empty!\n";
    }
    else {
        for(char c : college.replaceAll(" ", "").toCharArray()) {
            if (!Character.isAlphabetic(c) ) {
                message = message + "College name cannot have digits
or special characters!\n";
                break;
            }
        }
    }

    if (gpa.length() == 0) {
        message = message + "CGPA cannot be empty!\n";
    }
    else {
        try {
            float f = Float.parseFloat(gpa);
            if (f > 10 || f < 0) {
                message += "CGPA must lie between 0 to 10!\n";
            }
        }
        catch (Exception e) {
            message += "CGPA must be a valid Number!\n";
        }
    }

    if (skills.length() == 0) {
        message = message + "Skills cannot be empty!\n";
    }

    if (message.length() != 0) {
        JOptionPane.showMessageDialog(new JFrame(), message, "error",
JOptionPane.ERROR_MESSAGE);
    }
    else {
        Connection con;
        Statement s;
        ResultSet r;
        try {
            con = ConnectionManager.getConnection();

```

```

        s = con.createStatement();
        s.executeQuery("update students set firstname = '" + fname
+ "', lastname = '" + lname + "' where student_id = '" + student_id +
+ "'");

        s.executeQuery("commit");
        r = s.executeQuery("update studentprofile set dob = '" +
dob + "', collegename = '" + college + "', skills = '" + skills + "',
phonenumner = '" + phone + "', cgpa = '" + gpa + "' where student_id = '" +
student_id + "'");

        if ( ! r.next() ) {
            s.executeQuery("insert into studentprofile values('" +
student_id + "', '" + dob + "', '" + college + "', '" + skills + "', '" +
phone + "', '" + gpa + "')");
        }
        s.executeQuery("commit");
        s.close();
        con.close();
    }
    catch (Exception e1) {
        e1.printStackTrace();
    }
    JOptionPane.showMessageDialog(new JFrame(), "Updated
successfully!");
}

}

public void internshipsAppliedDelete(JFrame frame) {
    JTextField iidT = new JTextField();
    iidT.setBounds(150, 150, 150, 40);
    iidT.setFont(new Font("", Font.PLAIN, 16));

    JLabel prompt = new JLabel("<html><p style=\\\\"text-align:center;>
Enter the internship_id of the internship which you want to remove the
application </p>");
    prompt.setBounds(20, 50, 430, 50);
    prompt.setFont(new Font("", Font.PLAIN, 20));

    JButton delete = new JButton("Delete");
    delete.setFont(new Font("Delete", Font.PLAIN, 15));
    delete.setBounds(170, 220, 100, 30);
    delete.setBorder(BorderFactory.createLineBorder(Color.BLACK));
    delete.setBackground(Color.WHITE);
    delete.addActionListener(new ActionListener() {

```

```

public void actionPerformed(ActionEvent e) {
    String message = "";
    if (iidT.getText().length() == 0) {
        message = message + "internship_id cannot be
empty!\n";
    }
    else {
        try {
            int iid = Integer.parseInt(iidT.getText());
            Connection con;
            Statement s;
            ResultSet r1, r2;
            try {
                con = DriverManager.getConnection();
                s = con.createStatement();
                r1 = s.executeQuery("select * from internships
where internship_id = " + iid);
                if (! r1.next()) {
                    message = message + "internship_id does
not exist!\n";
                }
                else {
                    r2 = s.executeQuery("select * from
internshipsapplied where internship_id = " + iid + " and student_id = '" +
student_id + "'");
                    if (!r2.next() ) {
                        message = message + "You have not
applied to this internship yet!\n";
                    }
                }
                s.close();
                con.close();
            }
            catch (Exception e1) {
                e1.printStackTrace();
            }
        }
        catch (Exception e1) {
            message = message + "Enter a valid
internship_id!\n";
        }
    }
}

```

```

        if ( message.length() == 0 ) {
            int iid = Integer.parseInt(iidT.getText());
            Connection con;
            Statement s;
            try {
                con = ConnectionManager.getConnection();
                s = con.createStatement();
                s.executeQuery("delete from internshipsapplied
where internship_id = " + iid + " and student_id = '" + student_id + "'
");

                s.executeQuery("commit");
                s.close();
                con.close();
            }
            catch (Exception e1) {
                e1.printStackTrace();
            }
            iidT.setText("");
            JOptionPane.showMessageDialog(new JFrame(), "Deleted
successfully!");
        }
        else {
            JOptionPane.showMessageDialog(new JFrame(), message,
"error", JOptionPane.ERROR_MESSAGE);
        }

    }

});

iidT.addKeyListener(new KeyListener() {
    public void keyPressed(KeyEvent e) {
        if(e.getKeyCode() == KeyEvent.VK_ENTER) {
            delete.doClick();
        }
    }
    public void keyReleased(KeyEvent e) {}
    public void keyTyped(KeyEvent e) {}
});

frame.add(iidT);
frame.add(prompt);
frame.add(delete);
}

```

```

public void internshipsApply(JFrame frame) {
    JTextField iidT = new JTextField();
    iidT.setBounds(150, 150, 150, 40);
    iidT.setFont(new Font("", Font.PLAIN, 16));

    JLabel prompt = new JLabel("<html><p style=\\\\"text-align:center;>
Enter the internship_id of the internship which you want to apply for
</p>");
    prompt.setBounds(20, 50, 430, 50);
    prompt.setFont(new Font("", Font.PLAIN, 20));

    JButton apply = new JButton("Apply");
    apply.setFont(new Font("Apply", Font.PLAIN, 15));
    apply.setBounds(170, 220, 100, 30);
    apply.setBorder(BorderFactory.createLineBorder(Color.BLACK));
    apply.setBackground(Color.WHITE);
    apply.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            String message = "";
            if (iidT.getText().length() == 0) {
                message = message + "internship_id cannot be
empty!\\n";
            }
            else {
                try {
                    int iid = Integer.parseInt(iidT.getText());
                    Connection con;
                    Statement s;
                    ResultSet r1, r2;
                    try {
                        con = ConnectionManager.getConnection();
                        s = con.createStatement();
                        r1 = s.executeQuery("select * from internships
where internship_id = " + iid);
                        if (! r1.next()) {
                            message = message + "internship_id does
not exist!\\n";
                        }
                    }
                    else {
                        r2 = s.executeQuery("select * from
internshipsapplied where internship_id = " + iid + " and student_id = '" +
student_id + "'");

```



```

        if (r2.next() ) {
            message = message + "You have already
applied to this internship!\n";
        }
    }
    s.close();
    con.close();
}
catch (Exception e1) {
    e1.printStackTrace();
}
}
catch (Exception e1) {
    message = message + "Enter a valid
internship_id!\n";
}
}

if ( message.length() == 0) {
    int iid = Integer.parseInt(iidT.getText());
    Connection con;
    Statement s;
    try {
        con = ConnectionManager.getConnection();
        s = con.createStatement();
        s.executeQuery("insert into internshipsapplied
values ('" + student_id + "', " + iid + ", 'no')");
        s.executeQuery("commit");
        s.close();
        con.close();
    }
    catch (Exception e1) {
        e1.printStackTrace();
    }
    iidT.setText("");
    JOptionPane.showMessageDialog(new JFrame(), "Applied
successfully!");
}
else {
    JOptionPane.showMessageDialog(new JFrame(), message,
"error", JOptionPane.ERROR_MESSAGE);
}
}

```

```

    }

    });

    iidT.addKeyListener(new KeyListener() {
        public void keyPressed(KeyEvent e) {
            if(e.getKeyCode() == KeyEvent.VK_ENTER){
                apply.doClick();
            }
        }
        public void keyReleased(KeyEvent e) {}
        public void keyTyped(KeyEvent e) {}
    });

    frame.add(iidT);
    frame.add(prompt);
    frame.add(apply);
}
}

```

User interface for project managers

```

package UI;

import javax.swing.*;
import javax.swing.table.DefaultTableModel;

import main.ConnectionManager;

import java.awt.*;
import java.awt.event.*;
import java.sql.*;

public class Manager {
    public JFrame frame;
    public String manager_id;
    public JMenuBar menuBar;
    public JMenu profileMenu, internshipsMenu, internshipRequestsMenu;
    public JMenuItem viewProfile, editProfile, viewInternships,
postInternships, editInternships, deleteInternships,
viewInternshipRequests, approveInternshipRequests;
    public JButton back;
    public JLabel welcome;

```

1602-19-737-120

N.Taruni

```

public Manager(JFrame frame, String username) {
    this.frame = frame;
    this.manager_id = username;

    String name = " name";
    Connection con;
    Statement s;
    ResultSet rs;
    try {
        con = DriverManager.getConnection();
        s = con.createStatement();
        rs = s.executeQuery("select firstname from projectmanagers
where manager_id = '" + username + "'");
        rs.next();
        name = rs.getString(1);
        s.close();
        con.close();
    }
    catch (Exception e1) {
        e1.printStackTrace();
    }

    frame.setTitle("Project Manager: " + name);
    frame.getContentPane().setBackground(new Color(174, 230, 101));

    menuBar = new JMenuBar();
    frame.add(menuBar);

    profileMenu = new JMenu("Profile");

    profileMenu.setBorder(BorderFactory.createLineBorder(Color.BLACK));
    menuBar.add(profileMenu);

    internshipsMenu = new JMenu("Internships Posted by me");

    internshipsMenu.setBorder(BorderFactory.createLineBorder(Color.BLACK));
    menuBar.add(internshipsMenu);

    internshipRequestsMenu = new JMenu("Internship requests
available");

```

```

internshipRequestsMenu.setBorder(BorderFactory.createLineBorder(Color.BLACK));

menuBar.add(internshipRequestsMenu);

viewProfile = new JMenuItem("View profile");
viewProfile.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        welcome.setText("");
        frame.getContentPane().removeAll();
        frame.repaint();
        frame.add(back);
        profileView(frame);
    }
});

editProfile = new JMenuItem("Edit profile");
editProfile.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        welcome.setText("");
        frame.getContentPane().removeAll();
        frame.repaint();
        frame.add(back);
        profileEdit(frame);
    }
});

profileMenu.add(viewProfile);
profileMenu.add(editProfile);

viewInternships = new JMenuItem("View internships posted");
viewInternships.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        welcome.setText("");
        frame.getContentPane().removeAll();
        frame.repaint();
        frame.add(back);
        internshipsPostedView(frame);
    }
});

postInternships = new JMenuItem("Post new internships");
postInternships.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        welcome.setText("");
        frame.getContentPane().removeAll();
    }
});

```

```

        frame.repaint();
        frame.add(back);
        internshipsPostNew(frame);
    }
});
editInternships = new JMenuItem("Edit internships posted");
editInternships.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        welcome.setText("");
        frame.getContentPane().removeAll();
        frame.repaint();
        frame.add(back);
        internshipsPostedEdit(frame);
    }
});
deleteInternships = new JMenuItem("Delete internships posted");
deleteInternships.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        welcome.setText("");
        frame.getContentPane().removeAll();
        frame.repaint();
        frame.add(back);
        internshipsPostedDelete(frame);
    }
});
internshipsMenu.add(viewInternships);
internshipsMenu.add(editInternships);
internshipsMenu.add(postInternships);
internshipsMenu.add(deleteInternships);

viewInternshipRequests = new JMenuItem("View internships
requests");
viewInternshipRequests.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        welcome.setText("");
        frame.getContentPane().removeAll();
        frame.repaint();
        frame.add(back);
        internshipRequestsView(frame);
    }
});
approveInternshipRequests = new JMenuItem("Approve internship
requests");

```

```

        approveInternshipRequests.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                welcome.setText("");
                frame.getContentPane().removeAll();
                frame.repaint();
                frame.add(back);
                internshipRequestsApprove(frame);
            }
        });
        internshipRequestsMenu.add(viewInternshipRequests);
        internshipRequestsMenu.add(approveInternshipRequests);

        back = new JButton("Go back to Home Page");
        back.setBorder(BorderFactory.createLineBorder(Color.BLACK));
        back.setBackground(Color.WHITE);
        back.setFont(new Font("Go back to Home Page", Font.ITALIC, 11));
        back.setBounds(20, 320, 154, 23);
        frame.add(back);
        back.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                frame.getContentPane().removeAll();
                frame.setJMenuBar(null);
                new HomePage(frame);
            }
        });

        welcome = new JLabel("Welcome " + name + "!");
        welcome.setBounds(60, 120, 400, 40);
        Font promptFont = welcome.getFont();
        int stringWidth =
welcome.getFontMetrics(promptFont).stringWidth("Welcome <name>!");
        double widthRatio = (double)400 / (double)stringWidth;
        int newFontSize = (int)(promptFont.getSize() * widthRatio);
        welcome.setFont(new Font("Welcome", Font.ITALIC,
Math.min(newFontSize, 40)));
        frame.add(welcome);

        frame.setJMenuBar(menuBar);
        frame.setSize(500, 440);
        frame.setLayout(null);
        frame.setVisible(true);
    }

```

InternHub

```
public void profileView(JFrame frame) {
    JLabel unameF, unameV, lnameF, lnameV = null, fnameF, fnameV =
null, companyF, companyV = null, designationF, designationV = null,
phoneF, phoneV = null;
    unameF = new JLabel("Username:");
    unameV = new JLabel(manager_id);
    fnameF = new JLabel("First name:");
    lnameF = new JLabel("Last name:");
    companyF = new JLabel("Company name");
    designationF = new JLabel("Designation:");
    phoneF = new JLabel("Phone number:");

    Connection con;
    Statement s;
    ResultSet rs;
    try {
        con = DriverManager.getConnection();
        s = con.createStatement();
        rs = s.executeQuery("select * from projectmanagers where
manager_id = '" + manager_id + "'");
        rs.next();
        fnameV = new JLabel(rs.getString(2));
        lnameV = new JLabel(rs.getString(3));
        rs = s.executeQuery("select * from managerprofile where
manager_id = '" + manager_id + "'");
        if (rs.next()) {
            companyV = new JLabel(rs.getString(2));
            designationV = new JLabel(rs.getString(3));
            phoneV = new JLabel(rs.getString(4));
        }
        else {
            companyV = new JLabel("");
            designationV = new JLabel("");
            phoneV = new JLabel("");
        }
        s.close();
        con.close();
    }
    catch (Exception e1) {
        e1.printStackTrace();
    }

    unameF.setBounds(50, 30, 100, 30);
```

```
        unameF.setFont(new Font("", Font.PLAIN, 20));
        unameV.setBounds(250, 30, 100, 30);
        unameV.setFont(new Font("", Font.PLAIN, 20));

        fnameF.setBounds(50, 70, 100, 30);
        fnameF.setFont(new Font("", Font.PLAIN, 20));
        fnameV.setBounds(250, 70, 100, 30);
        fnameV.setFont(new Font("", Font.PLAIN, 20));

        lnameF.setBounds(50, 110, 100, 30);
        lnameF.setFont(new Font("", Font.PLAIN, 20));
        lnameV.setBounds(250, 110, 100, 30);
        lnameV.setFont(new Font("", Font.PLAIN, 20));

        companyF.setBounds(50, 150, 150, 30);
        companyF.setFont(new Font("", Font.PLAIN, 20));
        companyV.setBounds(250, 150, 140, 30);
        companyV.setFont(new Font("", Font.PLAIN, 20));

        phoneF.setBounds(50, 190, 150, 30);
        phoneF.setFont(new Font("", Font.PLAIN, 20));
        phoneV.setBounds(250, 190, 140, 30);
        phoneV.setFont(new Font("", Font.PLAIN, 20));

        designationF.setBounds(50, 230, 150, 30);
        designationF.setFont(new Font("", Font.PLAIN, 20));
        designationV.setBounds(250, 230, 180, 30);
        designationV.setFont(new Font("", Font.PLAIN, 20));

        frame.add(unameF);
        frame.add(unameV);
        frame.add(lnameF);
        frame.add(lnameV);
        frame.add(fnameF);
        frame.add(fnameV);
        frame.add(companyF);
        frame.add(companyV);
        frame.add(phoneF);
        frame.add(phoneV);
        frame.add(designationF);
        frame.add(designationV);
    }
```


InternHub

```
public void internshipsPostedView(JFrame frame) {
    String iid, position, salary, location, skills;
    String header[] = new String[] { "ID", "Role", "Salary",
    "Location", "Skills Required"};

    JTable table = new JTable();
    DefaultTableModel dtm = new DefaultTableModel(0, 0);
    dtm.setColumnIdentifiers(header);
    table.setModel(dtm);

    Connection con;
    Statement s;
    ResultSet r;
    try {
        con = ConnectionManager.getConnection();
        s = con.createStatement();
        r = s.executeQuery("select * from internships where manager_id
= '" + manager_id + "'");
        while (r.next()) {
            iid = r.getString(2);
            position = r.getString(3);
            salary = r.getString(4);
            location = r.getString(5);
            skills = r.getString(6);
            dtm.addRow(new Object[] {iid, position, salary, location,
skills});
        }
        s.close();
        con.close();
    }
    catch (Exception e1) {
        e1.printStackTrace();
    }
    table.setBounds(100, 70, 250, 100);
    table.getColumnModel().getColumn(0).setPreferredWidth(40);
    table.getColumnModel().getColumn(1).setPreferredWidth(180);
    table.getColumnModel().getColumn(2).setPreferredWidth(60);
    table.getColumnModel().getColumn(3).setPreferredWidth(90);
    table.getColumnModel().getColumn(4).setPreferredWidth(250);
    table.setFont(new Font("", Font.PLAIN, 15));
    table.setAutoResizeMode(JTable.AUTO_RESIZE_OFF);
    JScrollPane scroll = new JScrollPane(table);
    scroll.setBounds(15, 70, 450, 150);
}
```

```

        frame.add(scroll);
    }

    public void internshipRequestsView(JFrame frame) {
        String iid, position, student_id, skills, status;
        String header[] = new String[] { "ID", "Role", "studentid",
        "Skills", "Selected(yes/no)"};

        JTable table = new JTable();
        DefaultTableModel dtm = new DefaultTableModel(0, 0);
        dtm.setColumnIdentifiers(header);
        table.setModel(dtm);

        Connection con;
        Statement s1, s2, s3;
        ResultSet r1, r2, r3;
        try {
            con = ConnectionManager.getConnection();
            s1 = con.createStatement();
            s2 = con.createStatement();
            s3 = con.createStatement();
            r1 = s1.executeQuery("select * from internshipsapplied where
        internship_id in (select internship_id from internships where manager_id =
        '" + manager_id + "') and selectedyesno = 'no'");
            while (r1.next()) {
                student_id = r1.getString(1);
                iid = r1.getString(2);
                status = r1.getString(3);
                r2 = s2.executeQuery("select position from internships
        where internship_id = " + iid);
                r2.next();
                position = r2.getString(1);
                r3 = s3.executeQuery("select skills from studentprofile
        where student_id = '" + student_id + "'");
                r3.next();
                skills = r3.getString(1);
                dtm.addRow(new Object[] {iid, position, student_id,
        skills, status});
            }
            s1.close();
            s2.close();
            con.close();
        }
    }

```

```

        catch (Exception e1) {
            e1.printStackTrace();
        }
        table.setBounds(100, 70, 250, 100);
        table.getColumnModel().getColumn(0).setPreferredWidth(40);
        table.getColumnModel().getColumn(1).setPreferredWidth(150);
        table.getColumnModel().getColumn(2).setPreferredWidth(100);
        table.getColumnModel().getColumn(3).setPreferredWidth(250);
        table.getColumnModel().getColumn(4).setPreferredWidth(100);

        table.setAutoResizeMode(JTable.AUTO_RESIZE_OFF);
        table.setFont(new Font("", Font.PLAIN, 15));
        JScrollPane scroll = new JScrollPane(table);
        scroll.setBounds(15, 70, 450, 150);
        frame.add(scroll);
    }

    public void internshipRequestsApprove(JFrame frame) {
        JTextField iidT = new JTextField("internship_id");
        iidT.setBounds(80, 150, 150, 40);
        iidT.setFont(new Font("student_id", Font.PLAIN, 16));

        JTextField sidT = new JTextField("student_id");
        sidT.setBounds(260, 150, 150, 40);
        sidT.setFont(new Font("", Font.PLAIN, 16));

        JLabel prompt = new JLabel("<html><p style=\\\\"text-align:center;>
Enter the internship_id and student_id which you want to approve </p>");
        prompt.setBounds(20, 50, 430, 50);
        prompt.setFont(new Font("", Font.PLAIN, 20));

        JButton approve = new JButton("Approve");
        approve.setBounds(160, 250, 154, 23);
        approve.setBorder(BorderFactory.createLineBorder(Color.BLACK));
        approve.setBackground(Color.WHITE);
        approve.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                boolean iidexists = false, sidexists = false, isValid =
true;

                Connection con;
                Statement s;
                ResultSet rs;

```

```

String message = "", iid = iidT.getText(), sid =
sidT.getText();

if (iid.length() == 0 || iid == null) {
    message = message + "internship_id cannot be
empty!\n";
}
for (char c : iid.toCharArray()) {
    if (!Character.isDigit(c)) {
        message = message + "internship_id cannot have
alphabets or special characters!\n";
        isValid = false;
        break;
    }
}
if (isValid) {
    try {
        con = ConnectionManager.getConnection();
        s = con.createStatement();
        rs = s.executeQuery("select * from internships
where internship_id = " + iid);
        if (!rs.next()) {
            message = message + "internship_id does not
exist!\n";
        }
        else {
            rs = s.executeQuery("select * from internships
where internship_id = " + iid + " and manager_id = '" + manager_id + "'");
            if (!rs.next()) {
                message = message + "This internship_id
has not been posted by you, you cannot approveit!\n";
            }
            else {
                iidexists = true;
            }
        }
        s.close();
        con.close();
    }
    catch (Exception e1) {
        e1.printStackTrace();
    }
}

```

```

    }

    if (sid.length() == 0 || sid.equals("student_id") || sid
== null) {
        message = message + "student_id cannot be empty!\n";
    }
    else {
        try {
            con = ConnectionManager.getConnection();
            s = con.createStatement();
            rs = s.executeQuery("select * from students where
student_id = '" + sid + "'");
            if (!rs.next()) {
                message = message + "student_id does not
exist!\n";
            }
            else {
                rs = s.executeQuery("select * from
internshipsapplied where student_id = '" + sid + "'");
                if (!rs.next()) {
                    message = message + "This student has not
applied for any internship!\n";
                }
                else {
                    sidexists = true;
                }
            }
            s.close();
            con.close();
        }
        catch (Exception e1) {
            e1.printStackTrace();
        }
    }

    if (sidexists && iidexists) {
        try {
            con = ConnectionManager.getConnection();
            s = con.createStatement();
            rs = s.executeQuery("delete from
internshipsapplied where student_id = '" + sid + "' and internship_id = "
+ iid);

            if (!rs.next()) {

```

```

        message = message + "This student has not
applied to this internship!\n";
    }
    else {
        s.executeQuery("commit");
    }
    s.close();
    con.close();
}
catch (Exception e1) {
    e1.printStackTrace();
}
}

if (message.length() != 0) {
    JOptionPane.showMessageDialog(new JFrame(), message,
"error", JOptionPane.ERROR_MESSAGE);
}
else {
    JOptionPane.showMessageDialog(new JFrame(), "Approved
successfully!");
}
}

});

frame.add(prompt);
frame.add(iidT);
frame.add(sidT);
frame.add(approve);

}

public void profileEdit(JFrame frame) {
    JButton update = new JButton("Update");
    update.setBounds(200, 320, 154, 23);
    update.setBorder(BorderFactory.createLineBorder(Color.BLACK));
    update.setBackground(Color.WHITE);

    JLabel unameF, unameV, lnameF, fnameF, companyF, designationF,
phoneF;
    unameF = new JLabel("Username:");
    unameV = new JLabel(manager_id);
    fnameF = new JLabel("First name:");

```

InternHub

```
lnameF = new JLabel("Last name:");
companyF = new JLabel("Company name");
designationF = new JLabel("Designation:");
phoneF = new JLabel("Phone number:");

JTextField fnameV = new JTextField("");
JTextField lnameV = new JTextField("");
JTextField companyV = new JTextField("");
JTextField designationV = new JTextField("");
JTextField phoneV = new JTextField("");

Connection con;
Statement s;
ResultSet rs;
try {
    con = DriverManager.getConnection();
    s = con.createStatement();
    rs = s.executeQuery("select * from projectmanagers where
manager_id = '" + manager_id + "'");
    rs.next();
    fnameV.setText(rs.getString(2));
    lnameV.setText(rs.getString(3));
    rs = s.executeQuery("select * from managerprofile where
manager_id = '" + manager_id + "'");
    if (rs.next()) {
        companyV.setText(rs.getString(2));
        designationV.setText(rs.getString(3));
        phoneV.setText(rs.getString(4));
    }
    else {
        companyV.setText("");
        designationV.setText("");
        phoneV.setText("");
    }
    s.close();
    con.close();
}
catch (Exception e1) {
    e1.printStackTrace();
}

unameF.setBounds(50, 30, 100, 30);
unameF.setFont(new Font("", Font.PLAIN, 20));
```

```
unameV.setBounds(250, 30, 100, 30);
unameV.setFont(new Font("", Font.PLAIN, 20));

fnameF.setBounds(50, 70, 100, 30);
fnameF.setFont(new Font("", Font.PLAIN, 20));
fnameV.setBounds(250, 70, 100, 30);
fnameV.setFont(new Font("", Font.PLAIN, 20));

lnameF.setBounds(50, 110, 100, 30);
lnameF.setFont(new Font("", Font.PLAIN, 20));
lnameV.setBounds(250, 110, 100, 30);
lnameV.setFont(new Font("", Font.PLAIN, 20));

companyF.setBounds(50, 150, 150, 30);
companyF.setFont(new Font("", Font.PLAIN, 20));
companyV.setBounds(250, 150, 140, 30);
companyV.setFont(new Font("", Font.PLAIN, 20));

phoneF.setBounds(50, 190, 150, 30);
phoneF.setFont(new Font("", Font.PLAIN, 20));
phoneV.setBounds(250, 190, 140, 30);
phoneV.setFont(new Font("", Font.PLAIN, 20));

designationF.setBounds(50, 230, 150, 30);
designationF.setFont(new Font("", Font.PLAIN, 20));
designationV.setBounds(250, 230, 180, 30);
designationV.setFont(new Font("", Font.PLAIN, 20));

update.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        try {
            validateProfile(fnameV.getText(), lnameV.getText(),
companyV.getText(), phoneV.getText(), designationV.getText());
        }
        catch (Exception e1) {
            e1.printStackTrace();
        }
    }
});

frame.add(unameF);
frame.add(unameV);
frame.add(lnameF);
```


InternHub

```
        frame.add(lnameV);
        frame.add(fnameF);
        frame.add(fnameV);
        frame.add(companyF);
        frame.add(companyV);
        frame.add(phoneF);
        frame.add(phoneV);
        frame.add(designationF);
        frame.add(designationV);
        frame.add(update);
    }

    public void validateProfile(String fname, String lname, String
company, String phone, String designation) {
        String message = "";
        if (fname.length() == 0) {
            message = message + "First name cannot be empty!\n";
        }
        else {
            for(char c : fname.toCharArray()) {
                if (!Character.isAlphabetic(c)) {
                    message = message + "First name cannot have digits or
special characters!\n";
                    break;
                }
            }
        }

        for(char c : lname.toCharArray()) {
            if (!Character.isAlphabetic(c)) {
                message = message + "Last name cannot have digits or
special characters!\n";
                break;
            }
        }

        if (phone.length() == 0) {
            message = message + "Phone number cannot be empty!\n";
        }
        else {
            for(char c : phone.toCharArray()) {
                if (!Character.isDigit(c)) {
```

```

        message = message + "Phone number cannot have
alphabets or special characters!\n";
        break;
    }
}
if (phone.length() != 10) {
    message += "Phone number must have exactly 10
digits!\n";
}
}

if (company.length() == 0) {
    message = message + "Company name cannot be empty!\n";
}
else {
    for(char c : company.replaceAll(" ", "").toCharArray()) {
        if (!Character.isAlphabetic(c) ) {
            message = message + "Company name cannot have digits
or special characters!\n";
            break;
        }
    }
}

if (designation.length() == 0) {
    message = message + "College name cannot be empty!\n";
}
else {
    for(char c : designation.replaceAll(" ", "").toCharArray()) {
        if (!Character.isAlphabetic(c) ) {
            message = message + "Designation name cannot have
digits or special characters!\n";
            break;
        }
    }
}

if (message.length() != 0) {
    JOptionPane.showMessageDialog(new JFrame(), message, "error",
JOptionPane.ERROR_MESSAGE);
}
else {
    Connection con;

```

```

Statement s;
ResultSet r;
try {
    con = DriverManager.getConnection();
    s = con.createStatement();
    s.executeQuery("update projectmanagers set firstname = '"
+ fname + "', lastname = '" + lname + "' where manager_id = '" +
manager_id + "'");
    s.executeQuery("commit");
    r = s.executeQuery("update managerprofile set companyname
= '" + company + "', designation = '" + designation + "', phonenumber = "
+ phone + " where manager_id = '" + manager_id + "'");
    if (!r.next()) {
        s.executeQuery("insert into managerprofile values('" +
manager_id + "', '" + company + "', '" + designation + "', " + phone +
")");
    }
    s.executeQuery("commit");
    s.close();
    con.close();
}
catch (Exception e1) {
    e1.printStackTrace();
}
JOptionPane.showMessageDialog(new JFrame(), "Updated
successfully!");
}
}

public void internshipsPostNew(JFrame frame) {
    JTextField iidT = new JTextField();
    iidT.setBounds(150, 150, 150, 40);
    iidT.setFont(new Font("", Font.PLAIN, 16));

    JLabel prompt = new JLabel("<html><p style=\\\\"text-align:center;>
Enter the internship_id of the internship which you want to post </p>");
    prompt.setBounds(20, 50, 430, 50);
    prompt.setFont(new Font("", Font.PLAIN, 20));

    JButton post = new JButton("Post");
    post.setBounds(150, 220, 154, 23);
    post.setBorder(BorderFactory.createLineBorder(Color.BLACK));
    post.setBackground(Color.WHITE);

```

InternHub

```
post.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        String message = "";
        if (iidT.getText().length() == 0) {
            message = message + "internship_id cannot be
empty!\n";
        }
        else {
            try {
                int iid = Integer.parseInt(iidT.getText());
                Connection con;
                Statement s;
                ResultSet r;
                try {
                    con = ConnectionManager.getConnection();
                    s = con.createStatement();
                    r = s.executeQuery("select * from internships
where internship_id = " + iid);
                    if (r.next()) {
                        message = message + "internship_id already
exists! Please choose another internship_id\n";
                    }
                    s.close();
                    con.close();
                }
                catch (Exception e1) {
                    e1.printStackTrace();
                }
            }
            catch (Exception e1) {
                message = message + "Enter a valid
internship_id!\n";
            }
        }

        if ( message.length() == 0) {
            welcome.setText("");
            frame.getContentPane().removeAll();
            frame.repaint();
            frame.add(back);
            internshipsPostNewEditPage(frame, iidT.getText(),
"post");
        }
    }
});
```

```

        else {
            JOptionPane.showMessageDialog(new JFrame(), message,
"error", JOptionPane.ERROR_MESSAGE);
        }
    }
});

iidT.addKeyListener(new KeyListener() {
    public void keyPressed(KeyEvent e) {
        if(e.getKeyCode() == KeyEvent.VK_ENTER){
            post.doClick();
        }
    }
    public void keyReleased(KeyEvent e) {}
    public void keyTyped(KeyEvent e) {}
});

frame.add(prompt);
frame.add(post);
frame.add(iidT);

}

public void internshipsPostNewEditPage(JFrame frame, String iid,
String type ) {
    JLabel iidF, iidV, positionF, salaryF, locationF, skillsF;
    iidF = new JLabel("Internship_id:");
    iidV = new JLabel(iid);
    positionF = new JLabel("Role:");
    salaryF = new JLabel("Salary:");
    locationF = new JLabel("Location:");
    skillsF = new JLabel("Skills Required:");

    JTextField positionV = new JTextField("");
    JTextField locationV = new JTextField("");
    JTextField salaryV = new JTextField("");
    JTextField skillsV = new JTextField("");

    if (type.equals("edit")) {
        Connection con;
        Statement s;
        ResultSet r;
        try {

```

```

        con = ConnectionManager.getConnection();
        s = con.createStatement();
        r = s.executeQuery("select * from internships where
internship_id = " + iid);
        r.next();
        positionV.setText(r.getString(3));
        salaryV.setText(r.getString(4));
        locationV.setText(r.getString(5));
        skillsV.setText(r.getString(6));
        s.executeQuery("commit");
        s.close();
        con.close();
    }
    catch (Exception e1) {
        e1.printStackTrace();
    }
}

JButton post = new JButton("Post");
if (type.equals("edit")) {
    post.setText("Update");
}

post.setBounds(150, 260, 154, 23);
post.setBorder(BorderFactory.createLineBorder(Color.BLACK));
post.setBackground(Color.WHITE);
post.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        internshipsValidate(iid, positionV.getText(),
locationV.getText(), salaryV.getText(), skillsV.getText(), type);
    }
});

iidF.setBounds(50, 30, 200, 30);
iidF.setFont(new Font("", Font.PLAIN, 20));
iidV.setBounds(250, 30, 100, 30);
iidV.setFont(new Font("", Font.PLAIN, 20));

positionF.setBounds(50, 70, 100, 30);
positionF.setFont(new Font("", Font.PLAIN, 20));
positionV.setBounds(250, 70, 150, 30);
positionV.setFont(new Font("", Font.PLAIN, 20));

salaryF.setBounds(50, 110, 100, 30);

```

```

        salaryF.setFont(new Font("", Font.PLAIN, 20));
        salaryV.setBounds(250, 110, 150, 30);
        salaryV.setFont(new Font("", Font.PLAIN, 20));

        locationF.setBounds(50, 150, 100, 30);
        locationF.setFont(new Font("", Font.PLAIN, 20));
        locationV.setBounds(250, 150, 150, 30);
        locationV.setFont(new Font("", Font.PLAIN, 20));

        skillsF.setBounds(50, 190, 200, 30);
        skillsF.setFont(new Font("", Font.PLAIN, 20));
        skillsV.setBounds(250, 190, 150, 30);
        skillsV.setFont(new Font("", Font.PLAIN, 20));

        frame.add(post);
        frame.add(locationF);
        frame.add(locationV);
        frame.add(skillsF);
        frame.add(skillsV);
        frame.add(positionF);
        frame.add(positionV);
        frame.add(salaryF);
        frame.add(salaryV);
        frame.add(iidF);
        frame.add(iidV);
    }

    public void internshipsValidate(String iid, String position, String
location, String salary, String skills, String type) {
        String message = "";

        if (position == null || position.length() == 0) {
            message = message + "Position cannot be empty!\n";
        }
        else {
            for(char c : position.replaceAll(" ", "").toCharArray()) {
                if (!Character.isAlphabetic(c)) {
                    message = message + "Position cannot have digits or
special characters!\n";
                    break;
                }
            }
        }
    }
}

```

```

        if (location == null || location.length() == 0) {
            message = message + "Location cannot be empty!\n";
        }
        else {
            for(char c : location.replaceAll(" ", "").toCharArray()) {
                if (!Character.isAlphabetic(c)) {
                    message = message + "Location cannot have digits or
special characters!\n";
                    break;
                }
            }
        }

        if (skills == null || skills.length() == 0) {
            message = message + "Skills required cannot be empty!\n";
        }
        else {
            for(char c : skills.replaceAll(" ", "").toCharArray()) {
                if (!Character.isAlphabetic(c)) {
                    message = message + "Skills required cannot have
digits or special characters!\n";
                    break;
                }
            }
        }

        if (salary == null || salary.length() == 0) {
            message = message + "Salary number cannot be empty!\n";
        }
        else {
            try {
                Integer.parseInt(salary);
            }
            catch (NumberFormatException e) {
                message += "Salary cannot have alphabets or special
characters!\n";
            }
        }

        if (message.length() != 0) {
            JOptionPane.showMessageDialog(new JFrame(), message, "error",
JOptionPane.ERROR_MESSAGE);

```



```

    }

    else {
        Connection con;
        Statement s;
        try {
            con = ConnectionManager.getConnection();
            s = con.createStatement();
            if (type.equals("post")) {
                s.executeQuery("insert into internships values('" +
manager_id + "', " + iid + ", '" + position + "', " + salary + ", '" +
location + "', '" + skills + "')");
            }
            else {
                s.executeQuery("update internships set position = '" +
position + "', salary = " + salary + ", location = '" + location + "',
skills_required = '" + skills + "' where internship_id = " + iid + "and
manager_id = '" + manager_id + "'");
            }
            s.executeQuery("commit");
            s.close();
            con.close();
        }
        catch (Exception e1) {
            e1.printStackTrace();
        }
        if (type.equals("post")) {
            frame.getContentPane().removeAll();
            frame.repaint();
            frame.add(back);
            internshipsPostNew(frame);
            JOptionPane.showMessageDialog(new JFrame(), "Posted
successfully!");
        }
        else {
            JOptionPane.showMessageDialog(new JFrame(), "Updated
successfully!");
        }
    }

}

}

```

InternHub

```
public void internshipsPostedEdit(JFrame frame) {
    JTextField iidT = new JTextField();
    iidT.setBounds(150, 150, 150, 40);
    iidT.setFont(new Font("", Font.PLAIN, 16));

    JLabel prompt = new JLabel("<html><p style=\\\\"text-align:center;>
Enter the internship_id of the internship which you want to edit </p>");
    prompt.setBounds(20, 50, 430, 50);
    prompt.setFont(new Font("", Font.PLAIN, 20));

    JButton edit = new JButton("Update");
    edit.setBounds(150, 220, 154, 23);
    edit.setBorder(BorderFactory.createLineBorder(Color.BLACK));
    edit.setBackground(Color.WHITE);
    edit.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            String message = "";
            if (iidT.getText().length() == 0) {
                message = message + "internship_id cannot be
empty!\\n";
            }
            else {
                try {
                    int iid = Integer.parseInt(iidT.getText());
                    Connection con;
                    Statement s;
                    ResultSet r;
                    try {
                        con = ConnectionManager.getConnection();
                        s = con.createStatement();
                        r = s.executeQuery("select * from internships
where internship_id = " + iid);
                        if (!r.next()) {
                            message = message + "internship_id does
not exist!\\n";
                        }
                        else {
                            r = s.executeQuery("select * from
internships where internship_id = " + iid + " and manager_id = '" +
manager_id + "'");
                            if (!r.next() ) {
                                message = message + "You have not
posted this internship, you cannot edit it!\\n";

```

```

        }
    }
    s.close();
    con.close();
}
catch (Exception e1) {
    e1.printStackTrace();
}
}
catch (Exception e1) {
    message = message + "Enter a valid
internship_id!\n";
}
}

if ( message.length() == 0) {
    welcome.setText("");
    frame.getContentPane().removeAll();
    frame.repaint();
    frame.add(back);
    internshipsPostNewEditPage(frame, iidT.getText(),
"edit");
}
else {
    JOptionPane.showMessageDialog(new JFrame(), message,
"error", JOptionPane.ERROR_MESSAGE);
}
}
});

iidT.addKeyListener(new KeyListener() {
    public void keyPressed(KeyEvent e) {
        if(e.getKeyCode() == KeyEvent.VK_ENTER){
            edit.doClick();
        }
    }
    public void keyReleased(KeyEvent e) {}
    public void keyTyped(KeyEvent e) {}
});

frame.add(prompt);
frame.add(edit);
frame.add(iidT);

```

```

    }

    public void internshipsPostedDelete(JFrame frame) {
        JTextField iidT = new JTextField();
        iidT.setBounds(150, 150, 150, 40);
        iidT.setFont(new Font("", Font.PLAIN, 16));

        JLabel prompt = new JLabel("<html><p style=\\\\"text-align:center;>
Enter the internship_id of the internship which you want to delete </p>");
        prompt.setBounds(20, 50, 430, 50);
        prompt.setFont(new Font("", Font.PLAIN, 20));

        JButton delete = new JButton("Delete");
        delete.setFont(new Font("Delete", Font.PLAIN, 15));
        delete.setBounds(170, 220, 100, 30);
        delete.setBorder(BorderFactory.createLineBorder(Color.BLACK));
        delete.setBackground(Color.WHITE);
        delete.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                String message = "";
                if (iidT.getText().length() == 0) {
                    message = message + "internship_id cannot be
empty!\n";
                }
                else {
                    try {
                        int iid = Integer.parseInt(iidT.getText());
                        Connection con;
                        Statement s;
                        ResultSet r1, r2;
                        try {
                            con = ConnectionManager.getConnection();
                            s = con.createStatement();
                            r1 = s.executeQuery("select * from internships
where internship_id = " + iid);
                            if (! r1.next()) {
                                message = message + "internship_id does
not exist!\n";
                            }
                        }
                        else {
                            r2 = s.executeQuery("select * from
internships where internship_id = " + iid + " and manager_id = '" +
manager_id + "'");

```

```

        if (!r2.next() ) {
            message = message + "You have not
posted this internship, you cannot delete it!\n";
        }
    }
    s.close();
    con.close();
}
catch (Exception e1) {
    e1.printStackTrace();
}
}
catch (Exception e1) {
    message = message + "Enter a valid
internship_id!\n";
}
}

if ( message.length() == 0) {
    int iid = Integer.parseInt(iidT.getText());
    Connection con;
    Statement s;
    try {
        con = ConnectionManager.getConnection();
        s = con.createStatement();
        s.executeQuery("delete from internships where
internship_id = " + iid + " and manager_id = '" + manager_id + "' ");
        s.executeQuery("commit");
        s.close();
        con.close();
    }
    catch (Exception e1) {
        e1.printStackTrace();
    }
    iidT.setText("");
    JOptionPane.showMessageDialog(new JFrame(), "Deleted
successfully!");
}
else {
    JOptionPane.showMessageDialog(new JFrame(), message,
"error", JOptionPane.ERROR_MESSAGE);
}
}

```

InternHub

```
        }
    });

    iidT.addKeyListener(new KeyListener() {
        public void keyPressed(KeyEvent e) {
            if(e.getKeyCode() == KeyEvent.VK_ENTER) {
                delete.doClick();
            }
        }
        public void keyReleased(KeyEvent e) {}
        public void keyTyped(KeyEvent e) {}
    });

    frame.add(iidT);
    frame.add(prompt);
    frame.add(delete);
}
}
```

Main Program

```
package main;

import javax.swing.*;
import UI.*;

public class Caller {
    public static void main(String[] args) {

        JFrame frame = new JFrame();
        new HomePage(frame);
    }
}
```

GitHub Links and Folder Structure


Link: <https://github.com/taruni-always/internHub>

Folder Structure:


1602-19-737-120


N.Taruni


InternHub

	taruni-always refined the code	0d85fd4 7 hours ago	🕒 51 commits
dbTables	changed folder structure	27 days ago	
src	refined the code	7 hours ago	
.gitignore	added into gitignore	26 days ago	
README.md	created README.md	last month	

master	internHub / dbTables /	Go to file	Add file	...
	taruni-always changed folder structure	3b83ad2 27 days ago	🕒 History	
..				
DDL.txt	changed folder structure	27 days ago		
DML.txt	changed folder structure	27 days ago		
InternHub ER.png	changed folder structure	27 days ago		
InternHub_737_120.pdf	changed folder structure	27 days ago		

master	internHub / src /	Go to file	Add file	...
	taruni-always small changes	742299d 3 days ago	🕒 History	
..				
UI	small changes	3 days ago		
main	called the final homepage	5 days ago		

master	internHub / src / UI /	Go to file	Add file	...
	taruni-always small changes	742299d 3 days ago	🕒 History	
..				
HomePage.java	small changes	3 days ago		
LoginWithUsername.java	fixed pages overlapping on pressing back and made student viewprofile...	19 days ago		
Manager.java	added code for approve internships	5 days ago		
NewManagerProfile.java	fixed the phone number validation bug	5 days ago		
NewStudentProfile.java	fixed the phone number validation bug	5 days ago		
NewUserSignup.java	refined some lines	10 days ago		
Student.java	fixed the phone number validation bug and edit profile for users who ...	5 days ago		
UserLoginSignup.java	formatted tables	10 days ago		

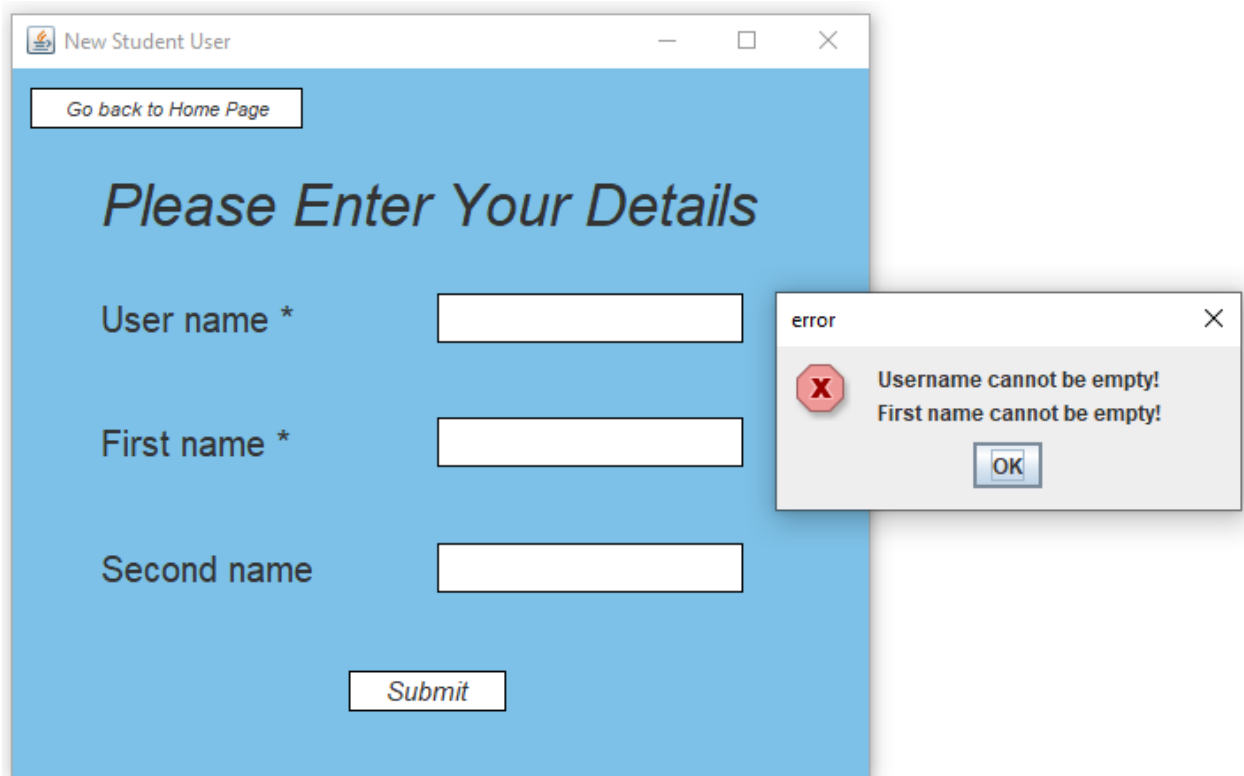
master	internHub / src / main /	Go to file	Add file	...
	taruni-always called the final homepage	8fef401 5 days ago	🕒 History	
..				
Caller.java	called the final homepage	5 days ago		
ConnectionManager.java	refined some lines	10 days ago		

1602-19-737-120
N.Taruni

TESTING

New student user sign-up

The student username is unique and it cannot be duplicate or empty. The first name cannot be empty or contain any digits or special characters. The last name may be left empty but it cannot contain digits or special characters. If any of the conditions fail, the user is sent a warning message.



The screenshot shows a web browser window titled "New Student User". Inside the window, there is a blue header bar with a "Go back to Home Page" button. Below the header, the text "Please Enter Your Details" is displayed. The form contains three input fields: "User name *", "First name *", and "Second name". Each field is currently empty. At the bottom of the form is a "Submit" button. An error message dialog box is overlaid on the right side of the form. The dialog box has a title bar that says "error" and a close button (X). The message inside the dialog box reads: "Username cannot be empty! First name cannot be empty!". There is a red 'X' icon next to the message and an "OK" button at the bottom of the dialog box.

The image shows a web browser window titled "New Student User". At the top left, there is a link "Go back to Home Page". The main heading is "Please Enter Your Details". Below this, there are three input fields: "User name *" with the value "sam123", "First name *" with the value "#123", and "Second name" with the value "123\$\$". A "Submit" button is at the bottom. An error dialog box is overlaid on the form, titled "error", with a red 'X' icon. The message in the dialog reads: "The username already exists. Please choose a different username! First name cannot have digits or special characters! Last name cannot have digits or special characters!". There is an "OK" button at the bottom of the dialog.

New Student User

[Go back to Home Page](#)


Please Enter Your Details

User name *

First name *

Second name

error

 The username already exists. Please choose a different username!
First name cannot have digits or special characters!
Last name cannot have digits or special characters!

When all the fields are correctly entered, we're prompted to enter our profile details:



The screenshot shows a web browser window titled "New Student Profile". Inside the window, there is a green background. At the top left, there is a button labeled "Go back to Home Page". Below this, a message reads "Welcome Sam, Please Enter Your Profile Details". The form contains five input fields with labels and asterisks indicating they are required:

- Date of Birth (Ex:05-dec-2001)*: 20-mar-1995
- Phone Number *: 1122334455
- College Name *: CBIT
- CGPA *: 9.2
- Skills *: Big data, Java

At the bottom center, there is a "Submit" button.

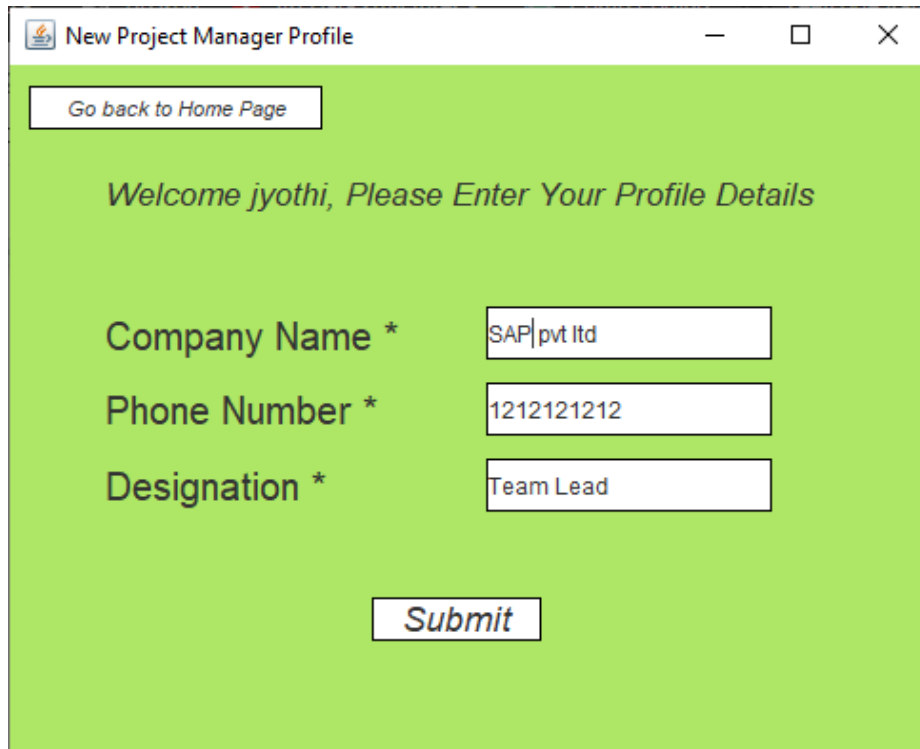
New project manager user sign-up

The student username is unique and it cannot be duplicate or empty. The first name cannot be empty or contain any digits or special characters. The last name may be left empty but it cannot contain digits or special characters. If any of the conditions fail, the user is sent a warning message.

The screenshot shows a web browser window titled "New Project Manager User". The page has a light blue background and a header with a "Go back to Home Page" link. The main heading is "Please Enter Your Details". There are three input fields: "User name *", "First name *", and "Second name". A "Submit" button is at the bottom. An error dialog box is open, displaying a red 'X' icon and the message: "Username cannot be empty! First name cannot be empty!". The dialog has an "OK" button.

The screenshot shows the same web browser window with the "New Project Manager User" form. The input fields now contain "jyo123" for "User name *", "jyo\$\$" for "First name *", and an empty field for "Second name". The "Submit" button is still present. A different error dialog box is open, displaying a red 'X' icon and the message: "The username already exists. Please choose a different username! First name cannot have digits or special characters!". The dialog has an "OK" button.

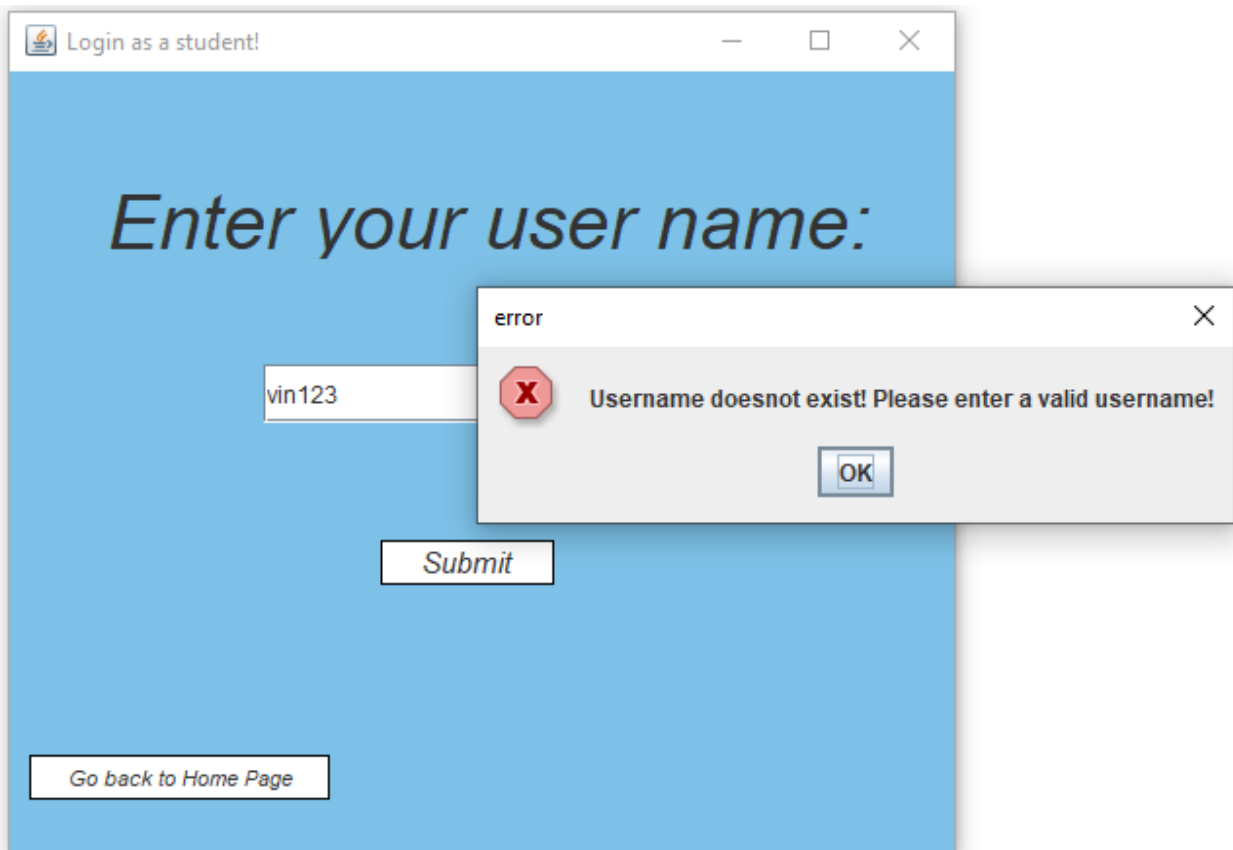
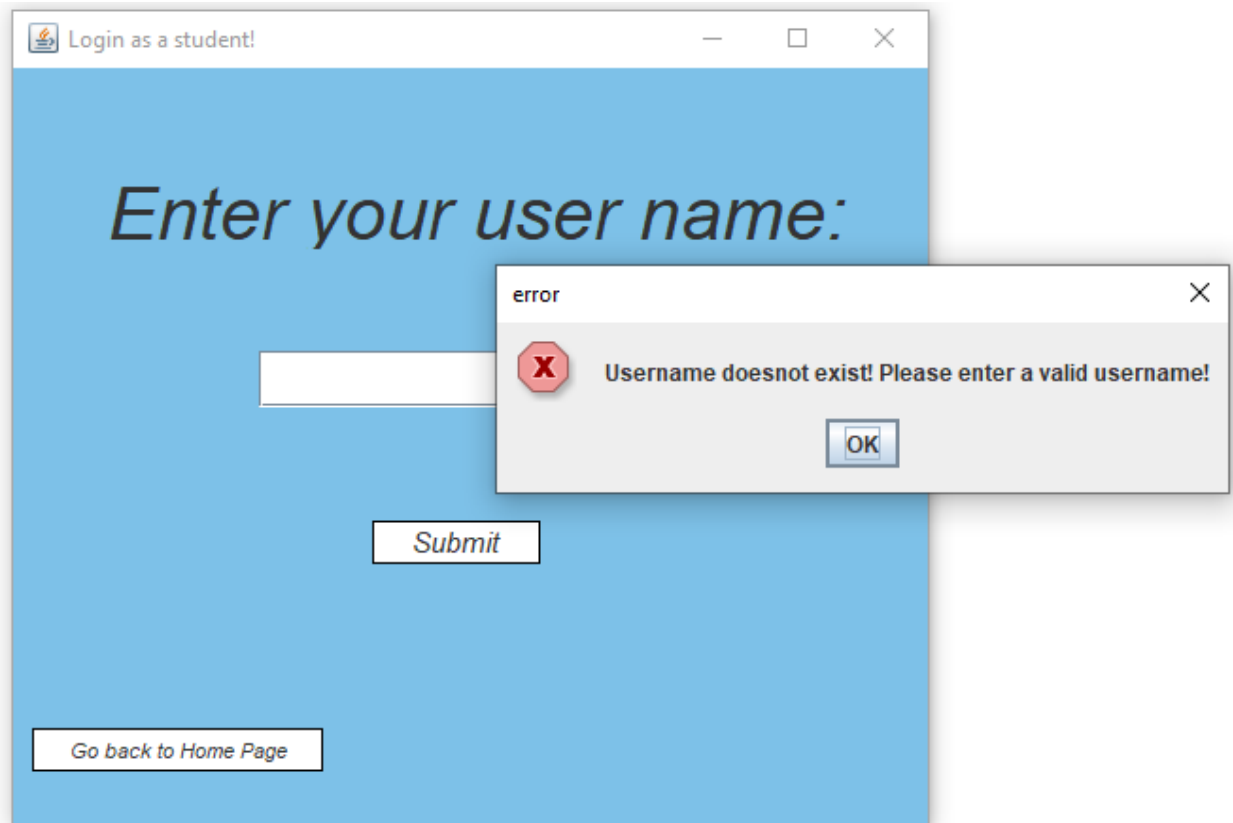
When all the fields are correctly entered, we're prompted to enter our profile details:



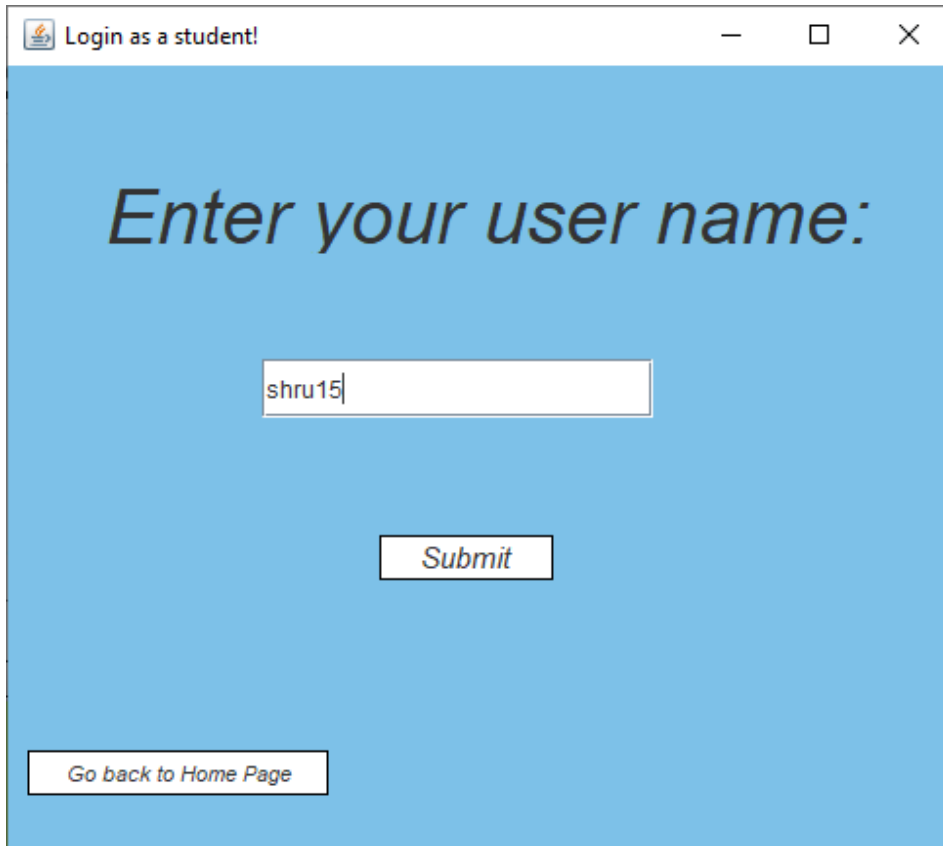
The screenshot shows a web browser window titled "New Project Manager Profile". Inside the window, there is a green background. At the top left, there is a button labeled "Go back to Home Page". Below this, a message reads "Welcome jyothi, Please Enter Your Profile Details". There are three input fields with labels and asterisks indicating they are required: "Company Name *" with the value "SAP|pvt ltd", "Phone Number *" with the value "1212121212", and "Designation *" with the value "Team Lead". At the bottom center, there is a "Submit" button.

Log-in as existing student user

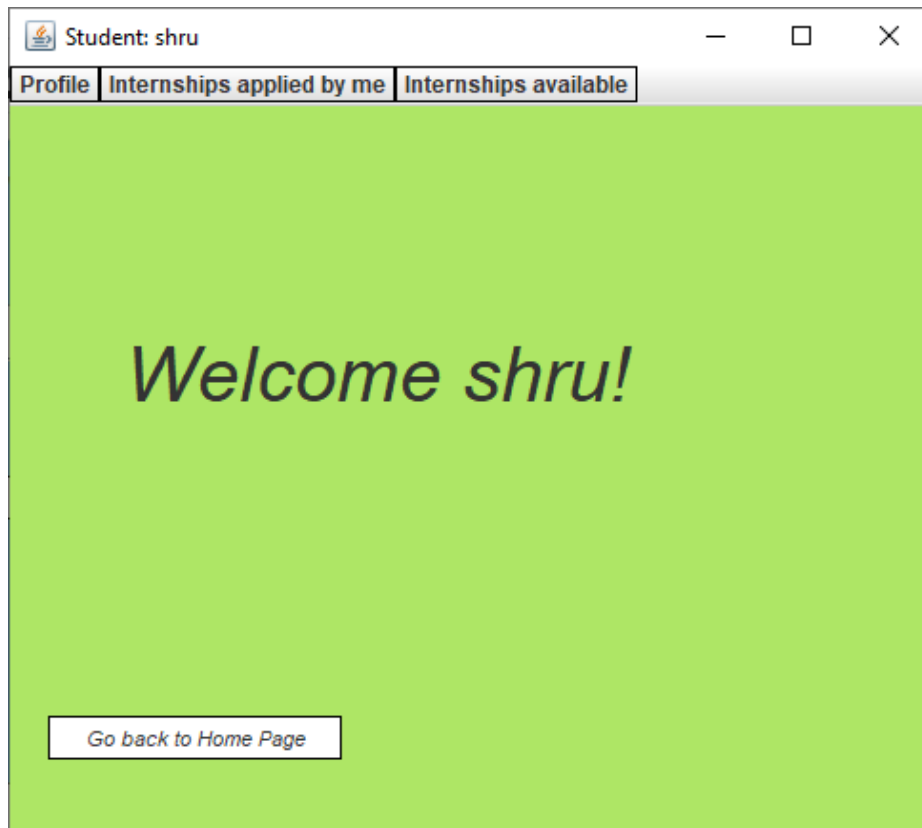
The username entered must be a valid one. It cannot be left empty. When the field is left empty or an invalid username is entered, a warning is shown.



When the username is correctly entered, this page appears:



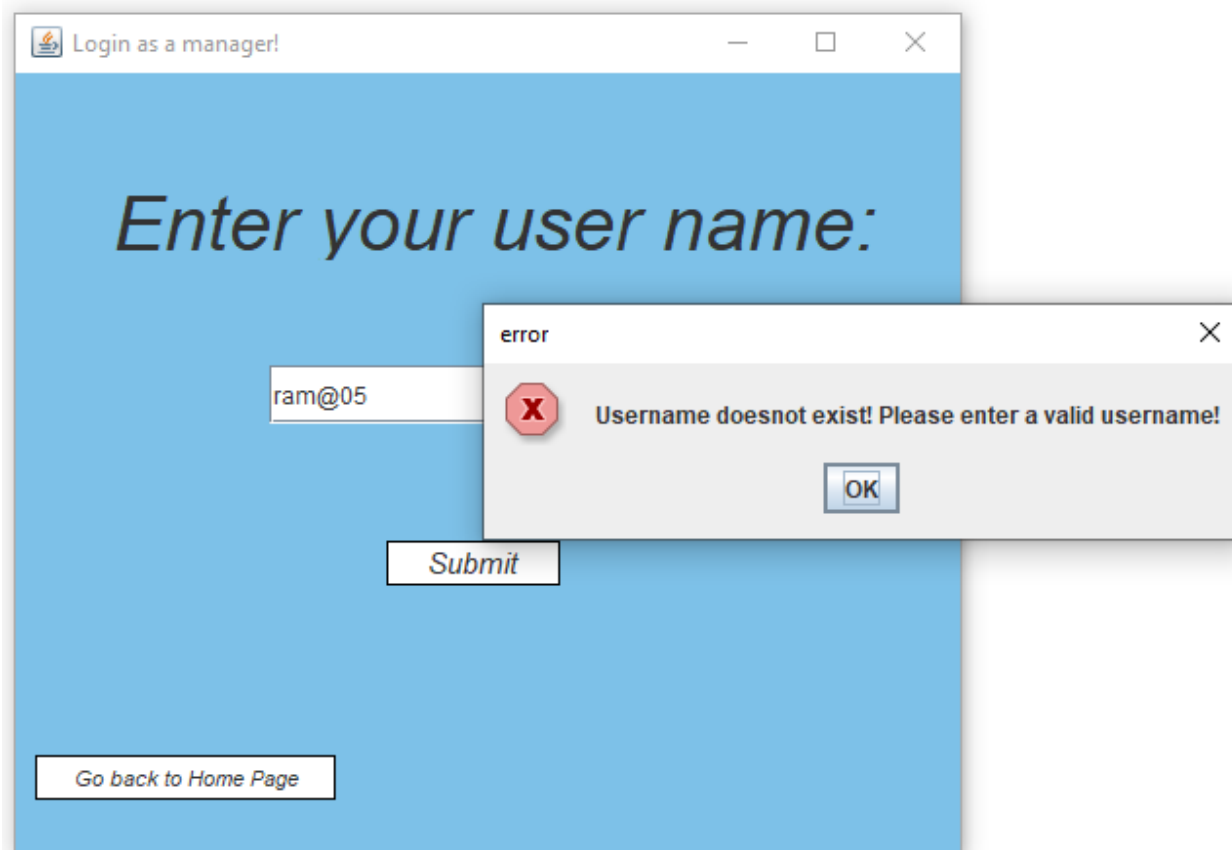
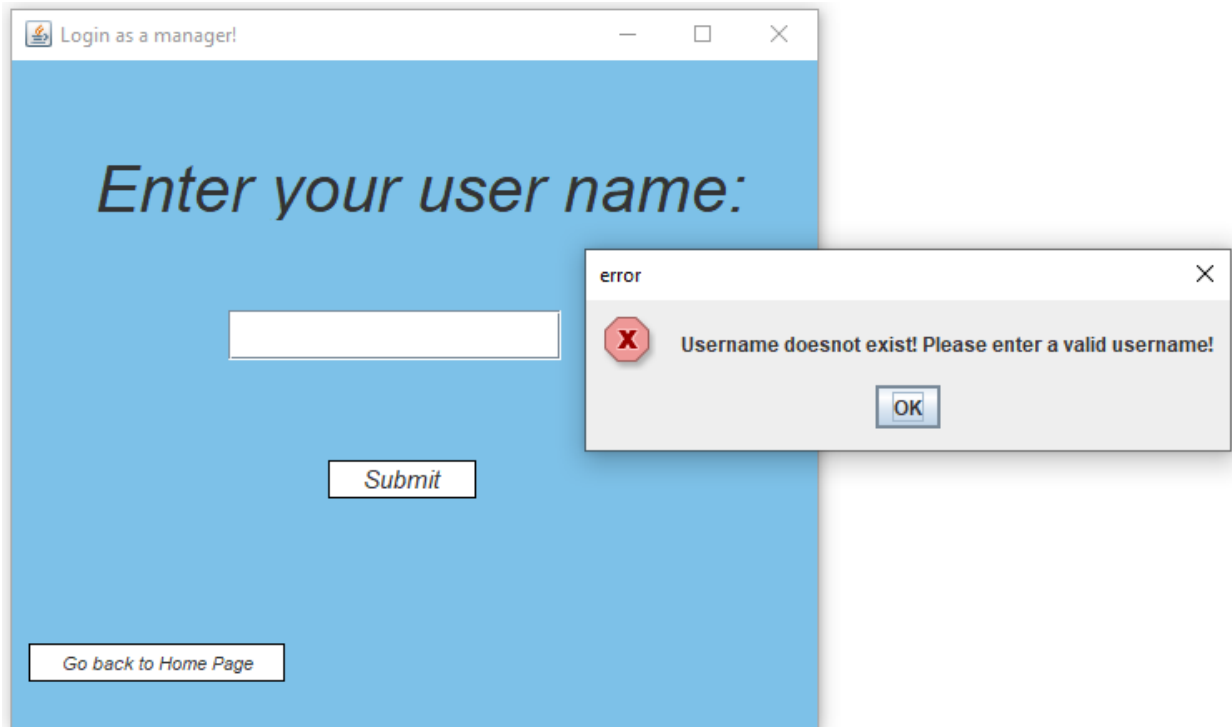
The screenshot shows a web browser window with the title "Login as a student!". The main content area has a light blue background. At the top, the text *Enter your user name:* is displayed in a large, italicized font. Below this, there is a text input field containing the username "shru15". Underneath the input field is a button labeled *Submit*. At the bottom left of the page, there is a button labeled *Go back to Home Page*.



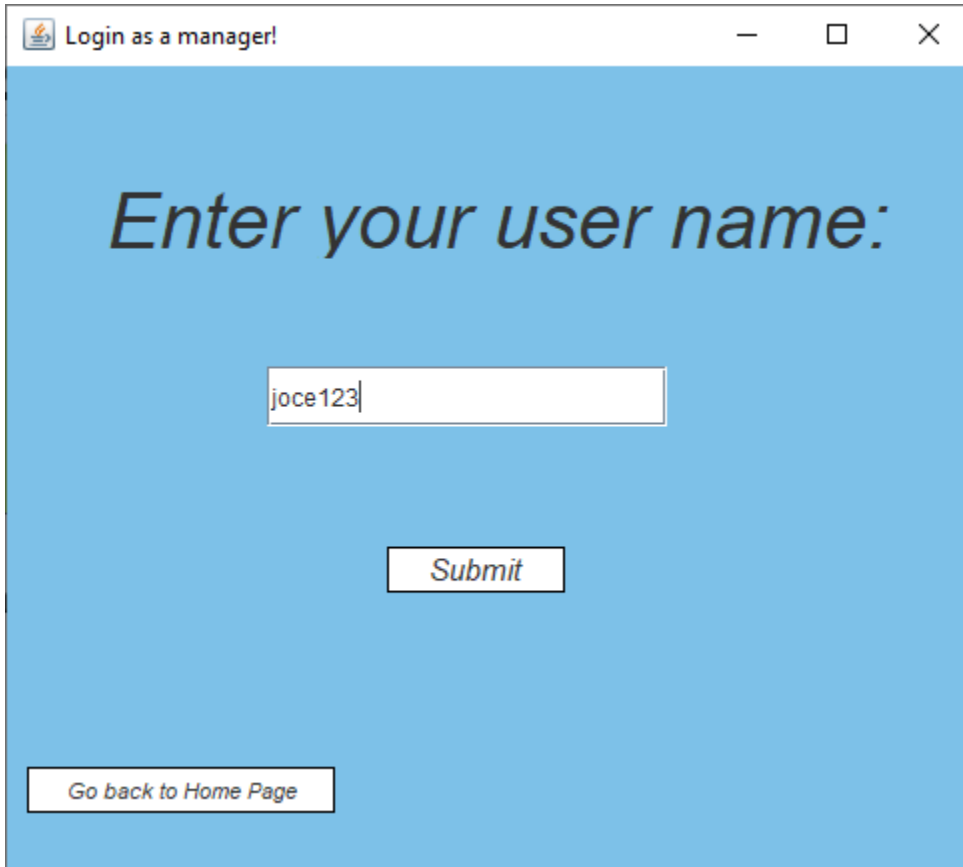
Log-in as existing project manager user

The username entered must be a valid one. It cannot be left empty.

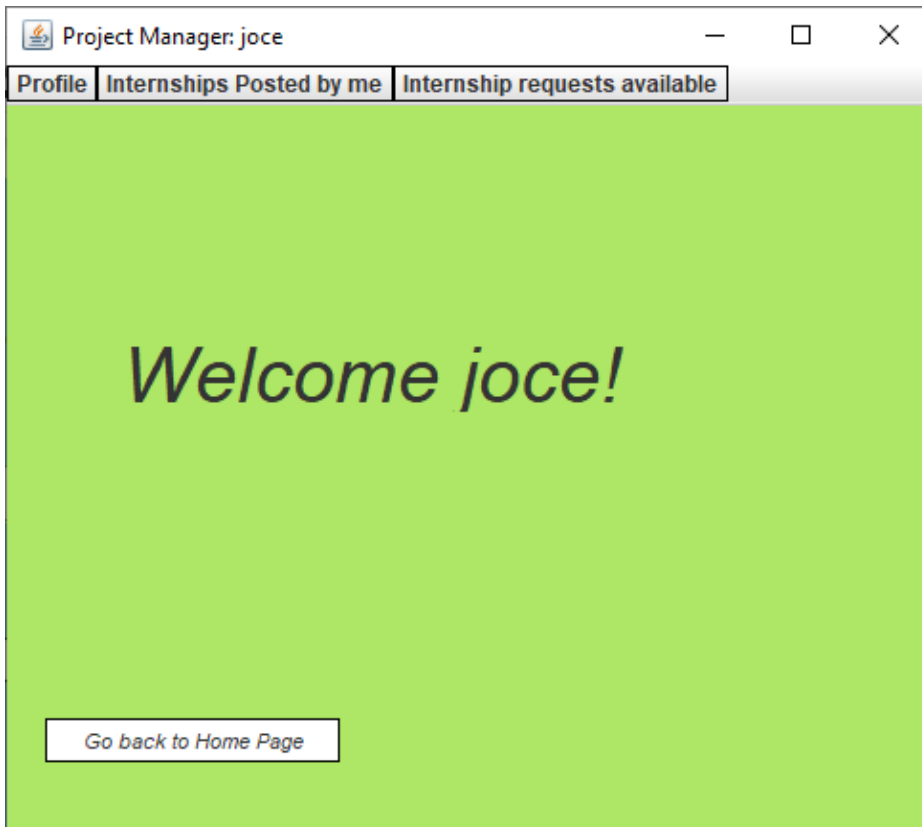
When the field is left empty or an invalid username is entered, a warning is shown:



When the username is correctly entered, this page appears:

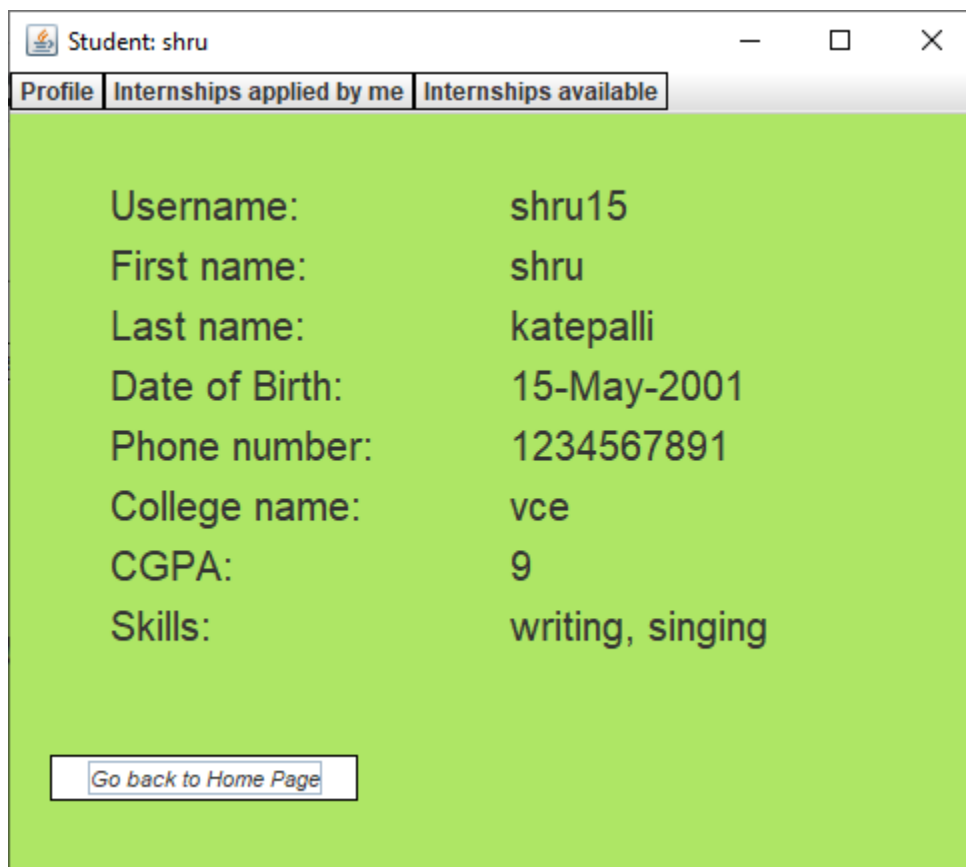
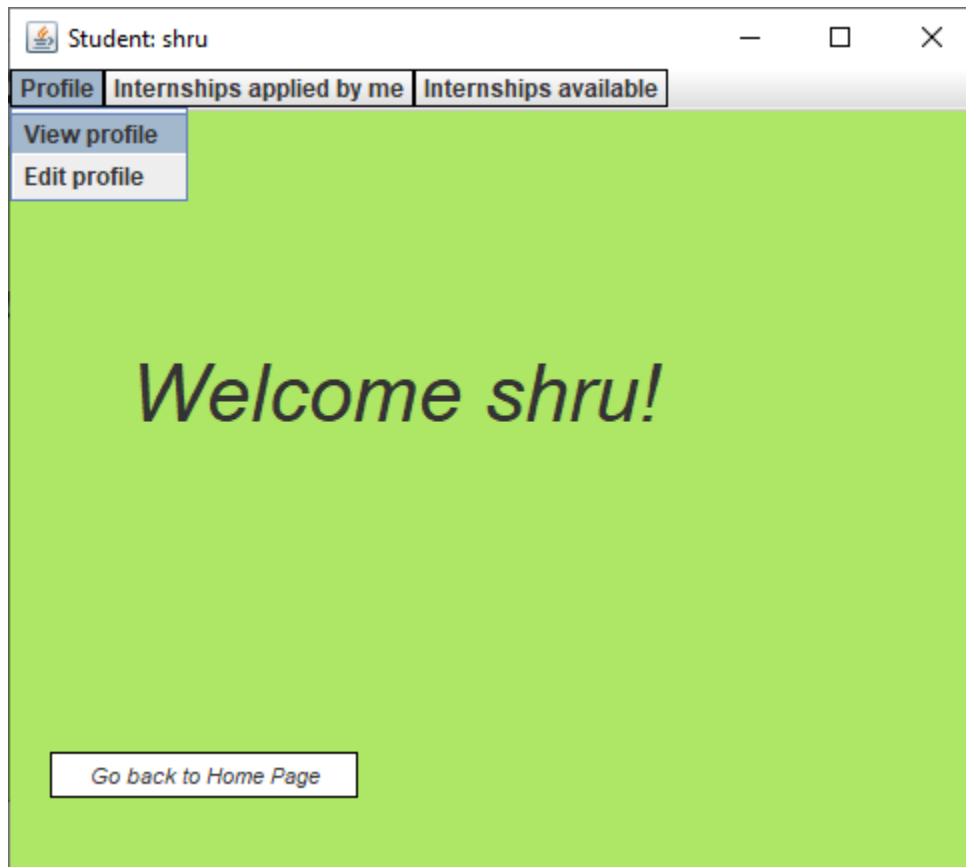


A screenshot of a web browser window titled "Login as a manager!". The window has a light blue background. At the top, there is a title bar with a small icon and the text "Login as a manager!". Below the title bar, the text "Enter your user name:" is displayed in a large, italicized font. Underneath this text is a text input field containing the text "jocel23". Below the input field is a button labeled "Submit". At the bottom left of the window is a button labeled "Go back to Home Page".



A screenshot of a web browser window titled "Project Manager: jocel". The window has a light green background. At the top, there is a title bar with a small icon and the text "Project Manager: jocel". Below the title bar, there is a navigation bar with three tabs: "Profile", "Internships Posted by me", and "Internship requests available". The "Profile" tab is currently selected. Below the navigation bar, the text "Welcome jocel!" is displayed in a large, italicized font. At the bottom left of the window is a button labeled "Go back to Home Page".

View student profile



Edit student profile

Student: shru		
Profile	Internships applied by me	Internships available
View profile		
Edit profile		
Username:	shru15	
First name:	shru	
Last name:	katepalli	
Date of Birth:	15-May-2001	
Phone number:	1234567891	
College name:	vce	
CGPA:	9	
Skills:	writing, singing	
Go back to Home Page		

No fields except the last name can be left empty. The phone number field only accepts a ten-digit phone number. The CGPA field cannot have alphabets or special characters. Not following the constraints sends a warning message.

Student: shru

Profile Internships applied by me Internships available

Username: shru15

First name:

Last name:

Date of Birth:

Phone number:

College name:

CGPA:

Skills:

error

First name cannot be empty!
Date of Birth cannot be empty!
Phone number cannot be empty!
College name cannot be empty!
CGPA cannot be empty!
Skills cannot be empty!

OK

Go back to Home Page Update

Student: shru

Profile Internships applied by me Internships available

Username: shru15

First name: shru@

Last name:

Date of Birth: testing

Phone number: abc

College name: haha

CGPA: testing

Skills: communication, writing

error

First name cannot have digits or special characters!
Enter a valid Date of Birth!
Phone number cannot have alphabets or special characters!
Phone number must have exactly 10 digits!
CGPA must be a valid Number!

OK

Go back to Home Page Update

Student: shru

Profile Internships applied by me Internships available

Username: shru15

First name: shru

Last name:

Date of Birth: 15-may-2001

Phone number: 1234

College name: VCE

CGPA: 9

Skills: communication, writing

Go back to Home Page Update

error

Phone number must have exactly 10 digits!

OK

Student: shru

Profile Internships applied by me Internships available

Username: shru15

First name: shru

Last name: katepalli

Date of Birth: 15-May-2001

Phone number: 1234567891

College name: vce

CGPA: 9

Skills: writing, singing

Go back to Home Page Update

When all the constraints are followed. The profile is updated successfully in the database.

The screenshot shows a web application window titled "Student: shru". It has three tabs: "Profile", "Internships applied by me", and "Internships available". The "Profile" tab is active. The form displays the following fields and values:

Field	Value
Username:	shru15
First name:	shru
Last name:	
Date of Birth:	
Phone number:	
College name:	
CGPA:	9
Skills:	writing, singing

A modal message box is displayed in the center, titled "Message", with an information icon and the text "Updated successfully!". It has an "OK" button.

At the bottom, there are two buttons: "Go back to Home Page" and "Update".

View internships applied by a student

The screenshot shows the same web application window, but the "Internships applied by me" tab is active. A dropdown menu is open, showing two options: "View internships applied" and "Delete internships applied". The form displays the following fields and values:

Field	Value
Username:	shru15
First name:	shru
Last name:	katepalli
Date of Birth:	15-May-2001
Phone number:	1234567891
College name:	vce
CGPA:	9
Skills:	writing, singing

At the bottom, there is a button: "Go back to Home Page".

Student: shru

Profile Internships applied by me Internships available

ID	Role	Salary	Location
201	sample	2000	blore
109	content writer	7000	hyderabad

[Go back to Home Page](#)

Delete internships applied by a student

Student: shru

Profile Internships applied by me Internships available

View internships applied
Delete internships applied

Username: shru15

First name: shru

Last name: katepalli

Date of Birth: 15-May-2001

Phone number: 1234567891

College name: vce

CGPA: 9

Skills: writing, singing

[Go back to Home Page](#) [Update](#)

Student: shru

Profile Internships applied by me Internships available

Enter the internship_id of the internship which you want to remove the application

Delete

[Go back to Home Page](#)

error

✖ internship_id cannot be empty!

OK

Student: shru

Profile Internships applied by me Internships available

Enter the internship_id of the internship which you want to remove the application

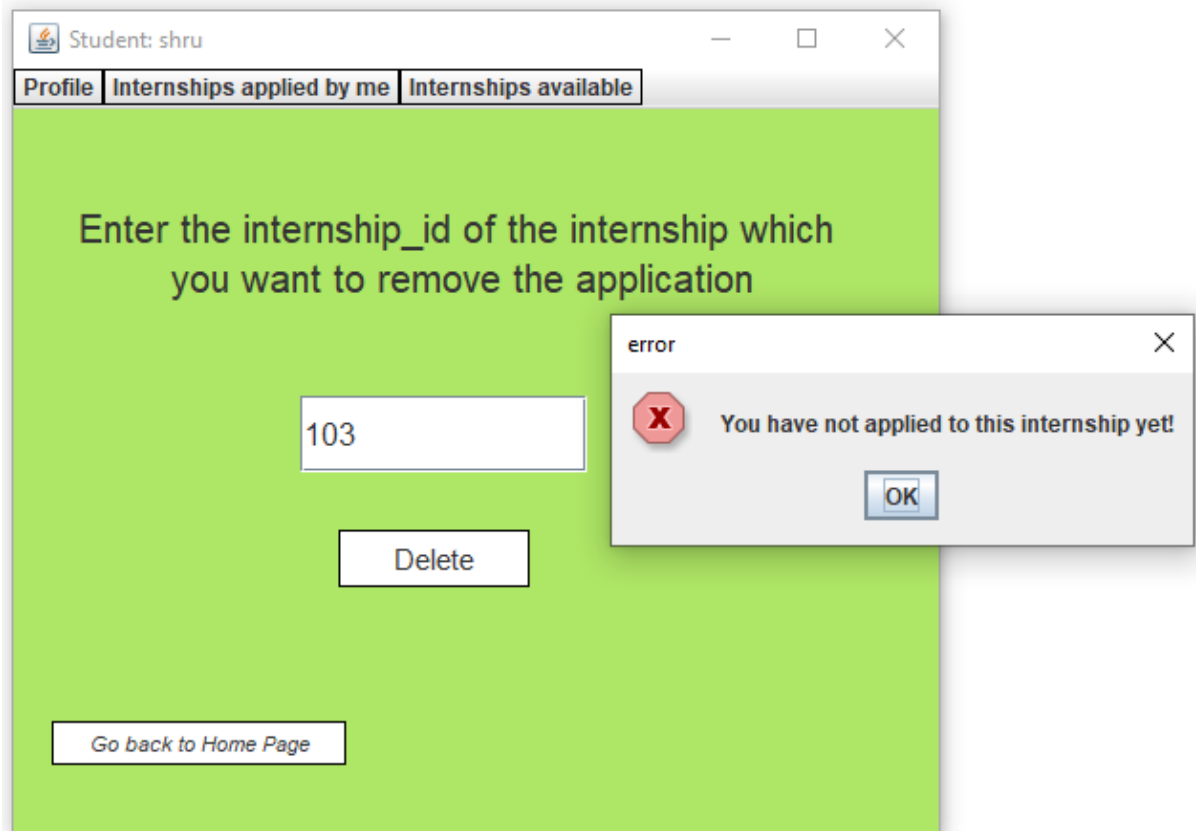
Delete

[Go back to Home Page](#)

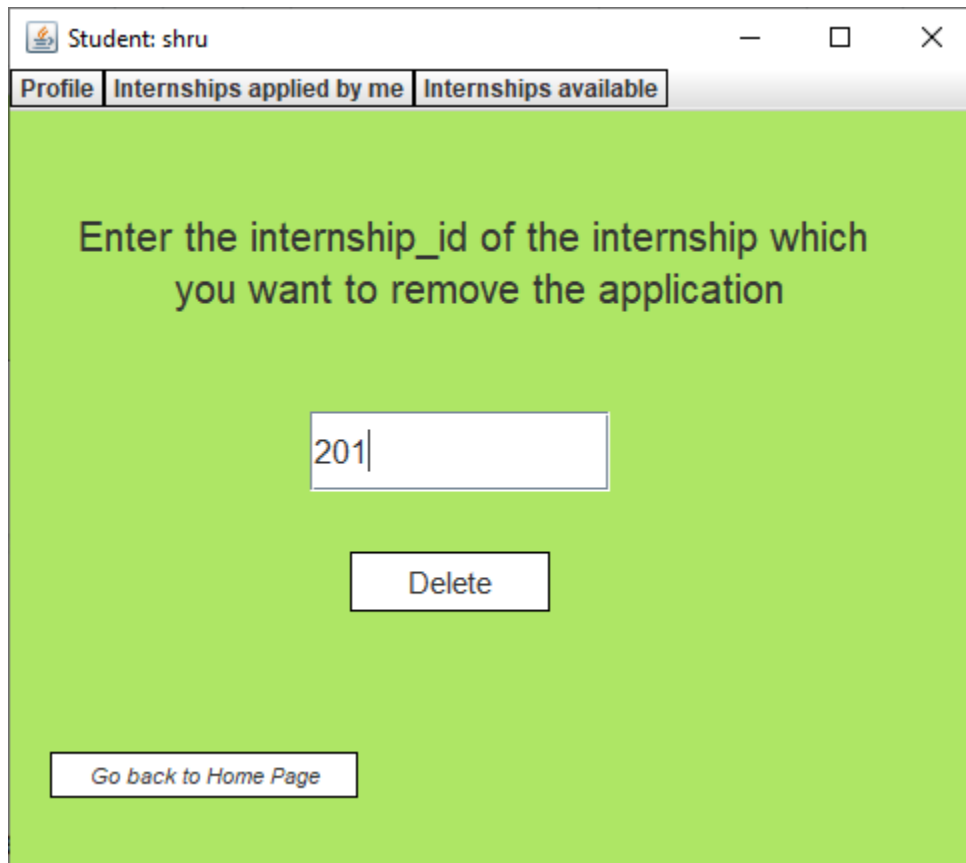
error

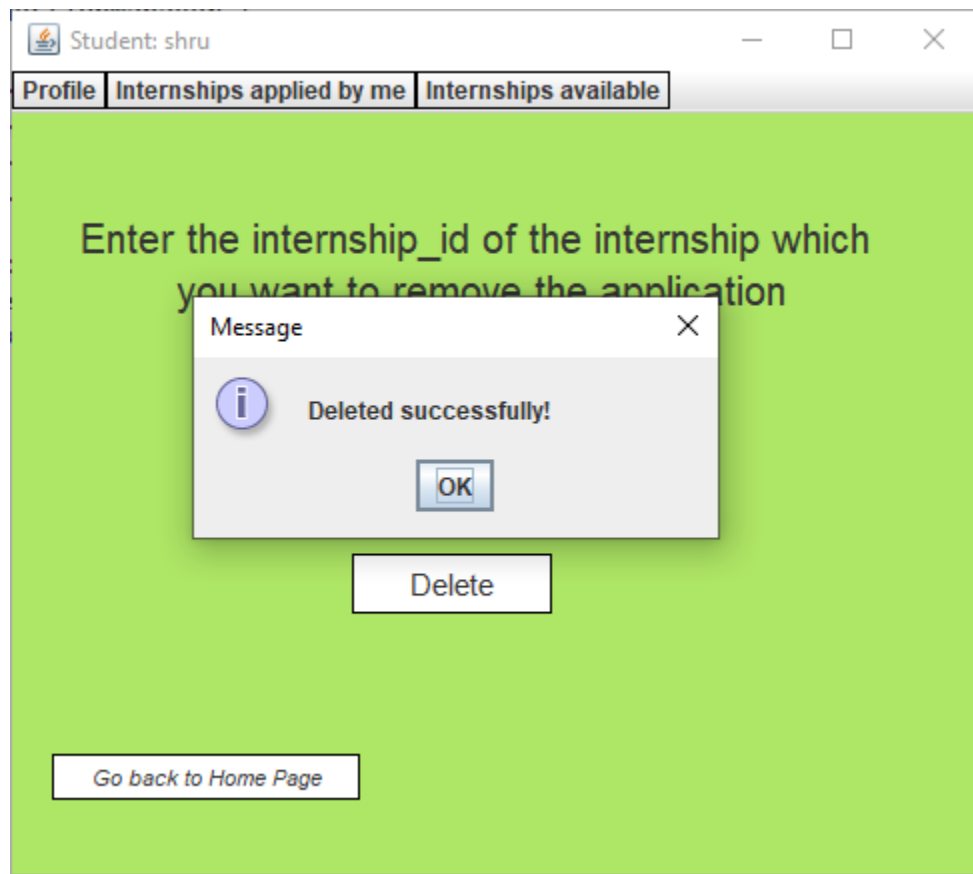
✖ Enter a valid internship_id!

OK




When a valid internship_id is entered, that the student has applied for, it gets deleted successfully from the database.





View internships available for a student to apply


 Student: shru

Profile
 Internships applied by me
 Internships available

View available internships
 Apply for new internships

ID	Role	Salary	Location
201	sample	2000	blore
109	content writer	7000	hyderabad

Go back to Home Page

 Student: shru

Profile
 Internships applied by me
 Internships available

ID	Role	Salary	Company
101	assistant director	8000	warner bros
102	data analyst	12000	fission labs
103	salesperson	4000	oracle
104	receptionist	6000	bookworld
105	assistant	4000	scarlet magazine
106	software testing	9000	google
200	testing	1800	oracle

Go back to Home Page

Apply for internships available by a student

Student: shru

Profile Internships applied by me Internships available

View available internships
Apply for new internships

Username: shru15

First name: shru

Last name: katepalli

Date of Birth: 15-May-2001

Phone number: 1234567891

College name: vce

CGPA: 9

Skills: writing, singing

[Go back to Home Page](#)


Student: shru

Profile Internships applied by me Internships available

Enter the internship_id of the internship which you want to apply for

[Go back to Home Page](#)

error

 internship_id cannot be empty!

Student: shru

Profile Internships applied by me Internships available

Enter the internship_id of the internship which you want to apply for

abc

Apply

Go back to Home Page

error

Enter a valid internship_id!

OK

Student: shru

Profile Internships applied by me Internships available

Enter the internship_id of the internship which you want to apply for

109

Apply

Go back to Home Page

error

You have already applied to this internship!

OK

When a valid internship_id is entered that hasn't been applied by the student yet, it gets successfully inserted into the database.

Student: shru

Profile Internships applied by me Internships available

Enter the internship_id of the internship which you want to apply for

101

Apply

Go back to Home Page

Student: shru

Profile Internships applied by me Internships available

Enter the internship_id of the internship which you want to apply for

Message

Applied successfully!

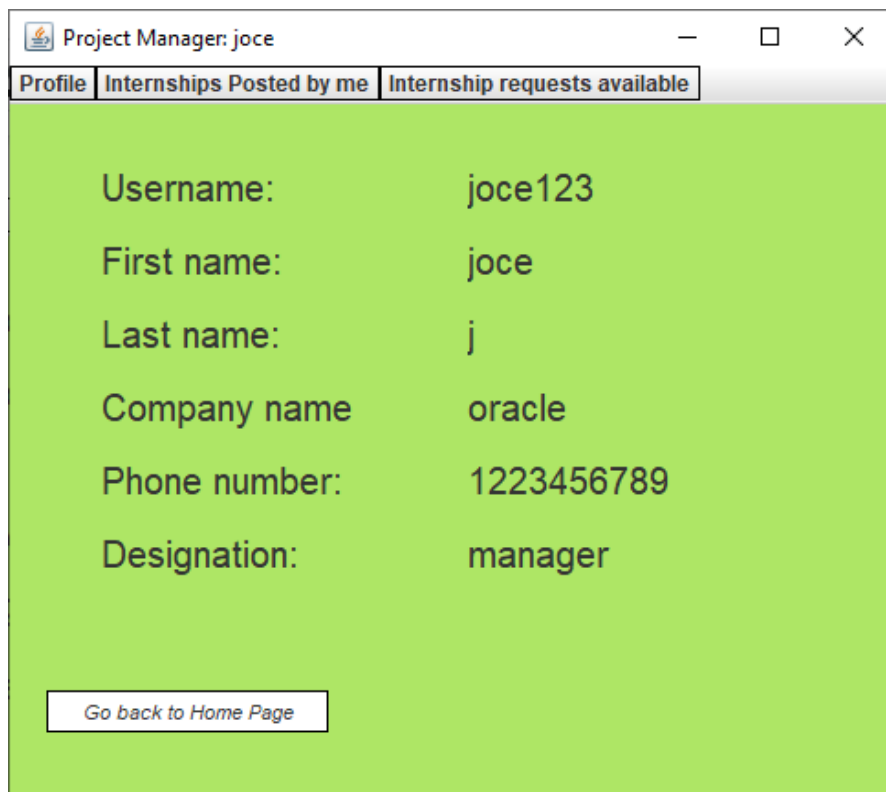
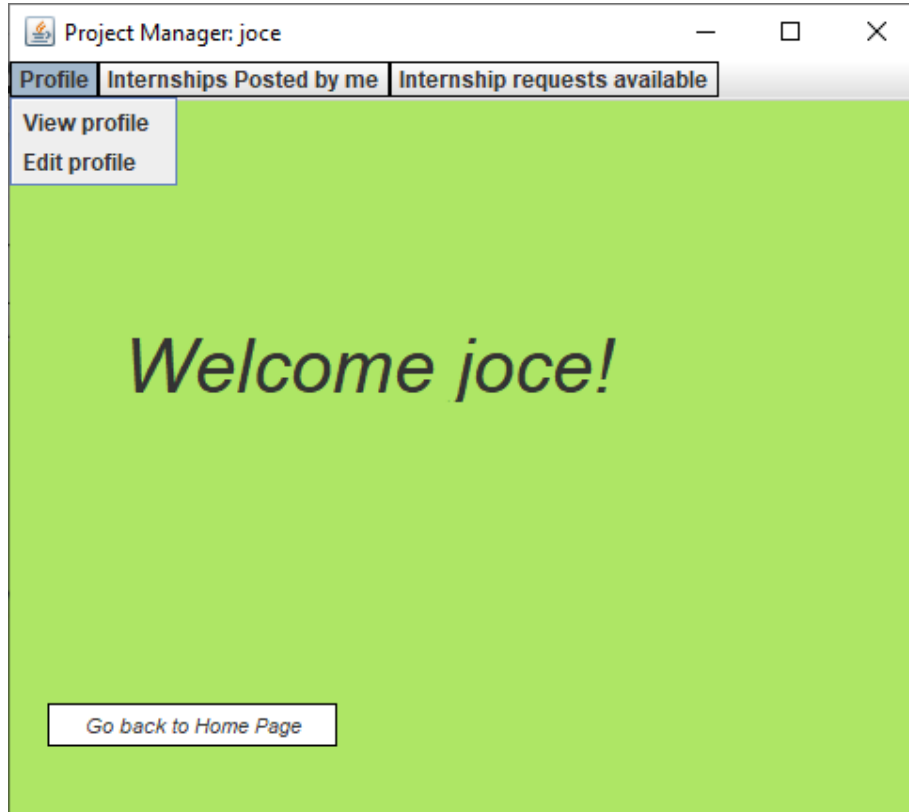
OK

Apply

Go back to Home Page

View project manager profile

On clicking the“view profile” menu item, the user profile is shown.



Edit project manager profile

Project Manager: joce

Profile

Internships Posted by me

Internship requests available

View profile

Edit profile

Username:

joce123

First name:

joce

Last name:

j

Company name

oracle

Phone number:

1223456789

Designation:

manager

Go back to Home Page

Project Manager: joce

Profile Internships Posted by me Internship requests available

Username: joce123

First name:

Last name:

Company name:

Phone number:

Designation:

[Go back to Home Page](#) [Update](#)

Project Manager: joce

Profile Internships Posted by me Internship requests available

Username: joce123

First name:

Last name:

Company name:

Phone number:

Designation:

[Go back to Home Page](#) [Update](#)

Message

Updated successfully!

OK

View Internships posted by a project manager:

Project Manager: joce

Profile | Internships Posted by me | Internship requests available

View internships posted
Edit internships posted
Post new internships
Delete internships posted

First name: joce123
Last name: joce
Last name: j
Company name: oracle
Phone number: 1223456789
Designation: manager

[Go back to Home Page](#)

Project Manager: joce

Profile | Internships Posted by me | Internship requests available

ID	Role	Salary	Location	
103	salesperson	4000	gujarat	communica
200	testing	1800	hyd	software
201	sample	2000	blore	testing

[Go back to Home Page](#)

Edit internships posted by a project manager

Project Manager: joce

Profile Internships Posted by me Internship requests available

View internships posted
Edit internships posted
Post new internships
Delete internships posted

ID		Salary	Location	
103	salesperson	4000	gujarat	communica
200	testing	1800	hyd	software
201	sample	2000	blore	testing

Go back to Home Page

Project Manager: joce

Profile Internships Posted by me Internship requests available

Enter the internship_id of the internship which you want to edit

Update

Go back to Home Page

error

✖ internship_id cannot be empty!

OK

Project Manager: joce

Profile | Internships Posted by me | Internship requests available

Enter the internship_id of the internship which you want to edit

abc

Update

Go back to Home Page

error

Enter a valid internship_id!

OK

Project Manager: joce

Profile | Internships Posted by me | Internship requests available

Enter the internship_id of the internship which you want to edit

105


Update

Go back to Home Page

error

You have not posted this internship, you cannot edit it!

OK


 Project Manager: joce

Profile Internships Posted by me Internship requests available

Enter the internship_id of the internship which you want to edit

Update

Go back to Home Page

 Project Manager: joce

Profile Internships Posted by me Internship requests available

Internship_id:	200
Role:	<input type="text" value="testing"/>
Salary:	<input type="text" value="1800"/>
Location:	<input type="text" value="hyd"/>
Skills Required:	<input type="text" value="software"/>

Update

Go back to Home Page

No fields can be left empty. The salary field cannot contain alphabets or special characters.

Project Manager: joce

Profile Internships Posted by me Internship requests available

Internship_id: 200

Role:

Salary:

Location:

Skills Required:

Update

Go back to Home Page

error

Position cannot be empty!
Location cannot be empty!
Skills required cannot be empty!
Salary number cannot be empty!

OK

Project Manager: joce

Profile Internships Posted by me Internship requests available

Internship_id: 200

Role: Team Lead

Salary: abc

Location: hyd

Skills Required: software

Update

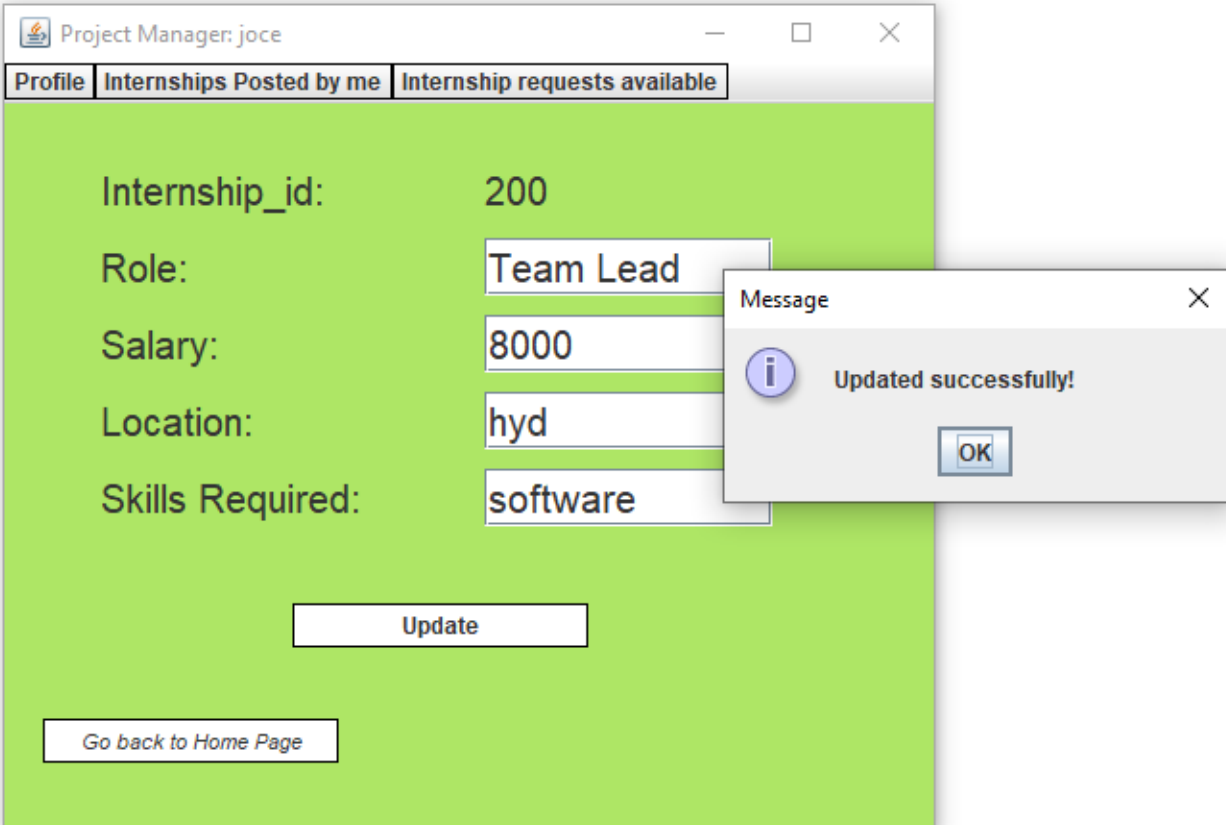
Go back to Home Page

error

Salary cannot have alphabets or special characters!

OK

When all the constraints are followed, it gets updated successfully in the database.



The screenshot displays a web application window titled "Project Manager: joce". It features a navigation bar with three tabs: "Profile", "Internships Posted by me", and "Internship requests available". The main content area has a light green background and contains the following form fields:

- Internship_id: 200
- Role: Team Lead
- Salary: 8000
- Location: hyd
- Skills Required: software

Below the form fields is an "Update" button. At the bottom left, there is a button labeled "Go back to Home Page". A modal message box is overlaid on the right side of the form, displaying the text "Updated successfully!" with an information icon and an "OK" button.

Delete internships posted by project manager

Project Manager: joce

Profile

Internships Posted by me

Internship requests available

View internships posted

Edit internships posted

Post new internships

Delete internships posted

200

testing

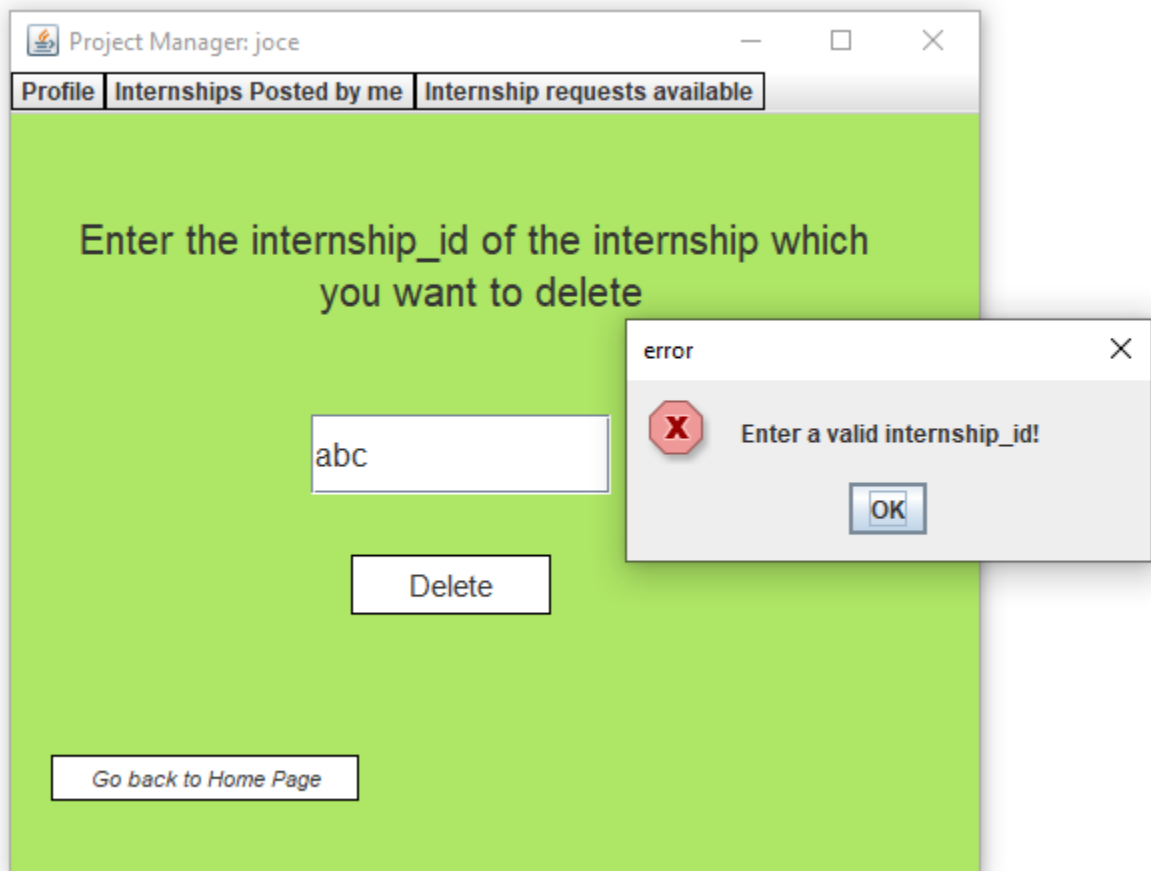
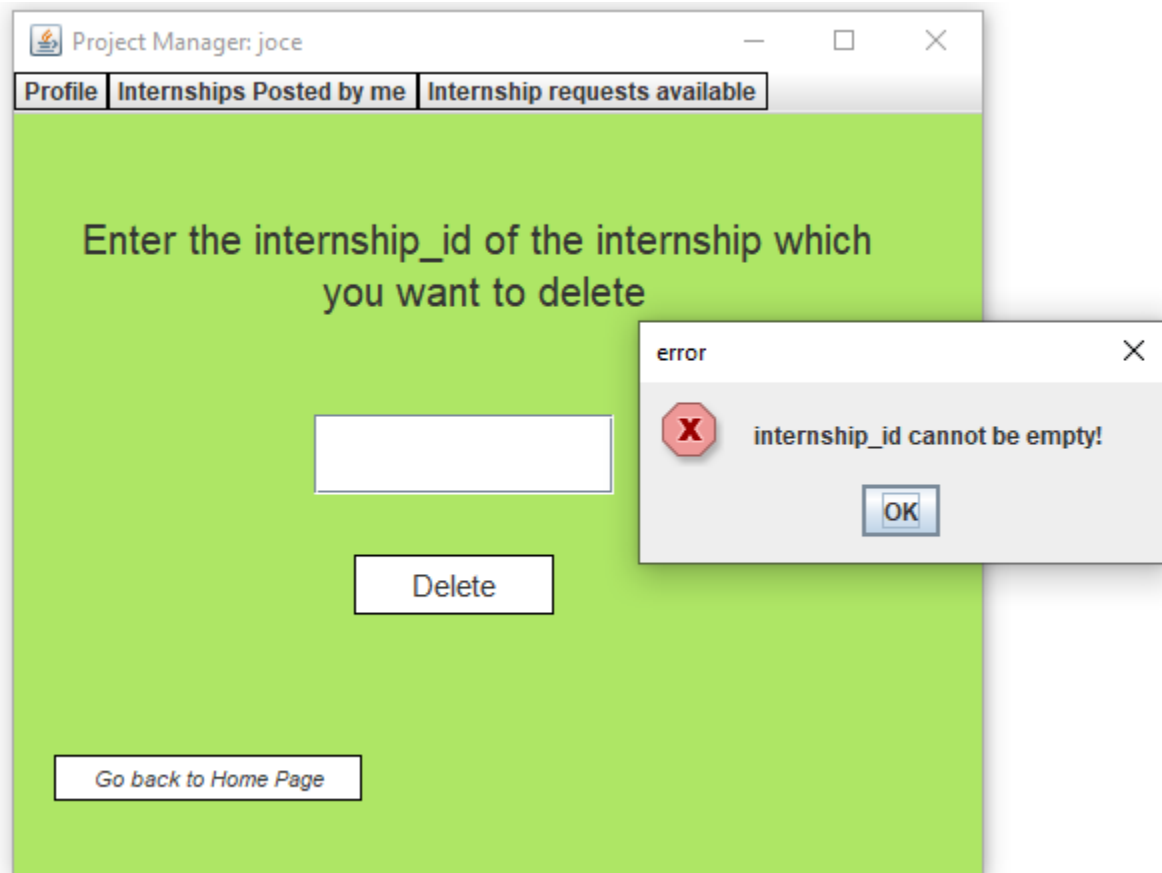
Salary: 1600

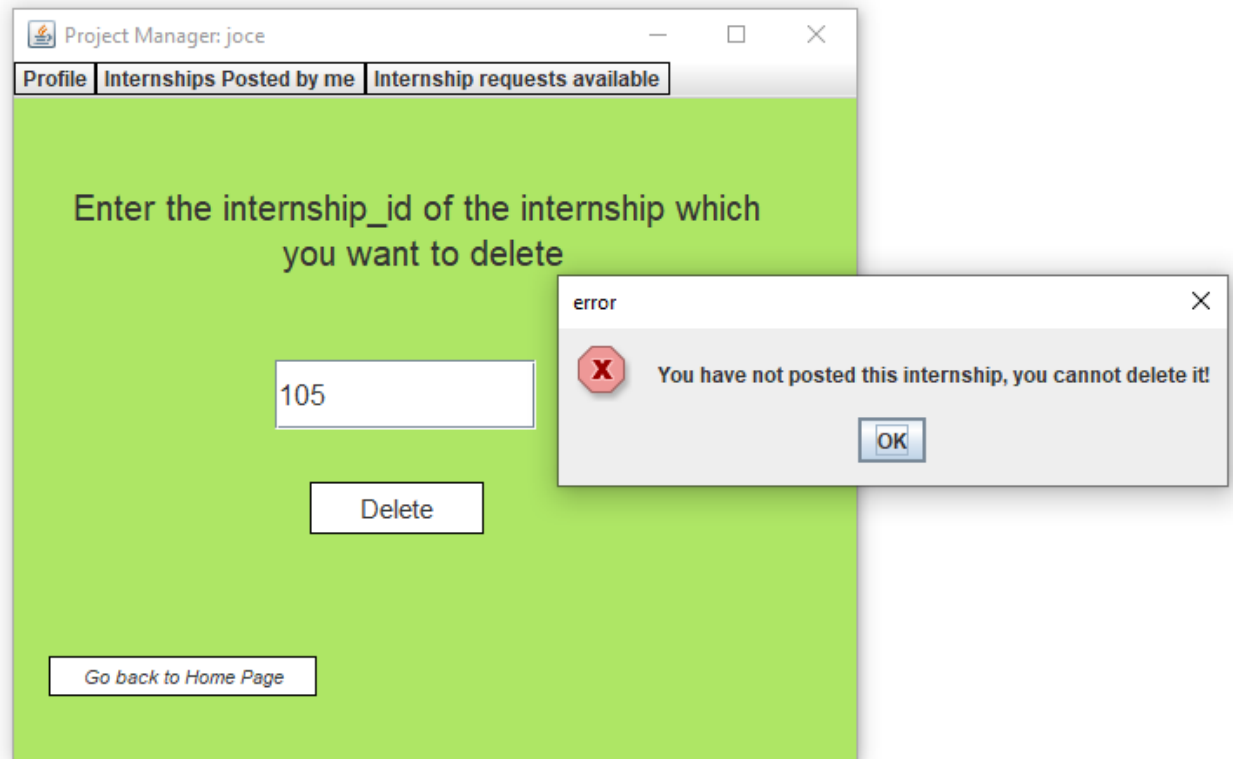
Location: hyd

Skills Required: software

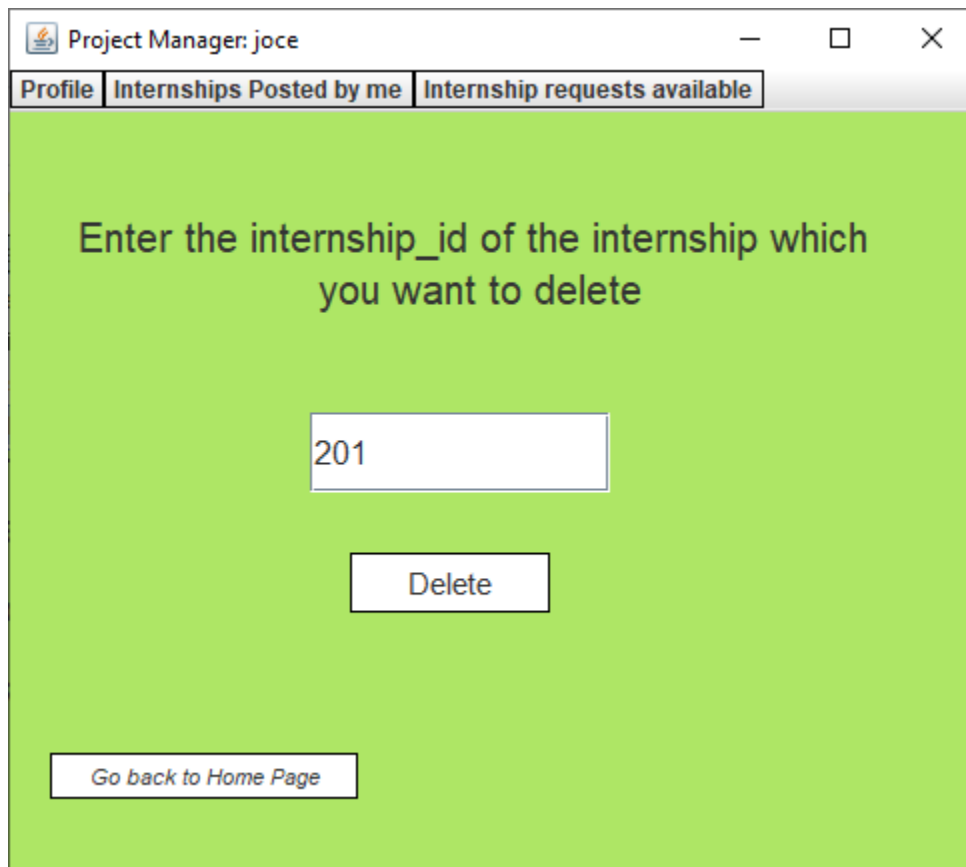
Update

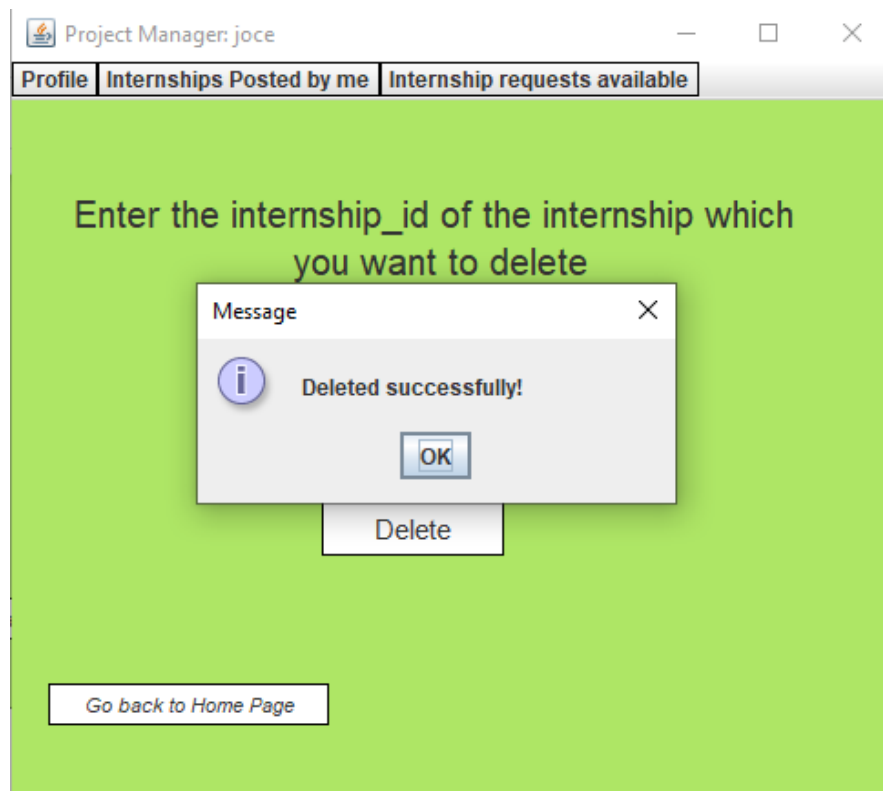
Go back to Home Page



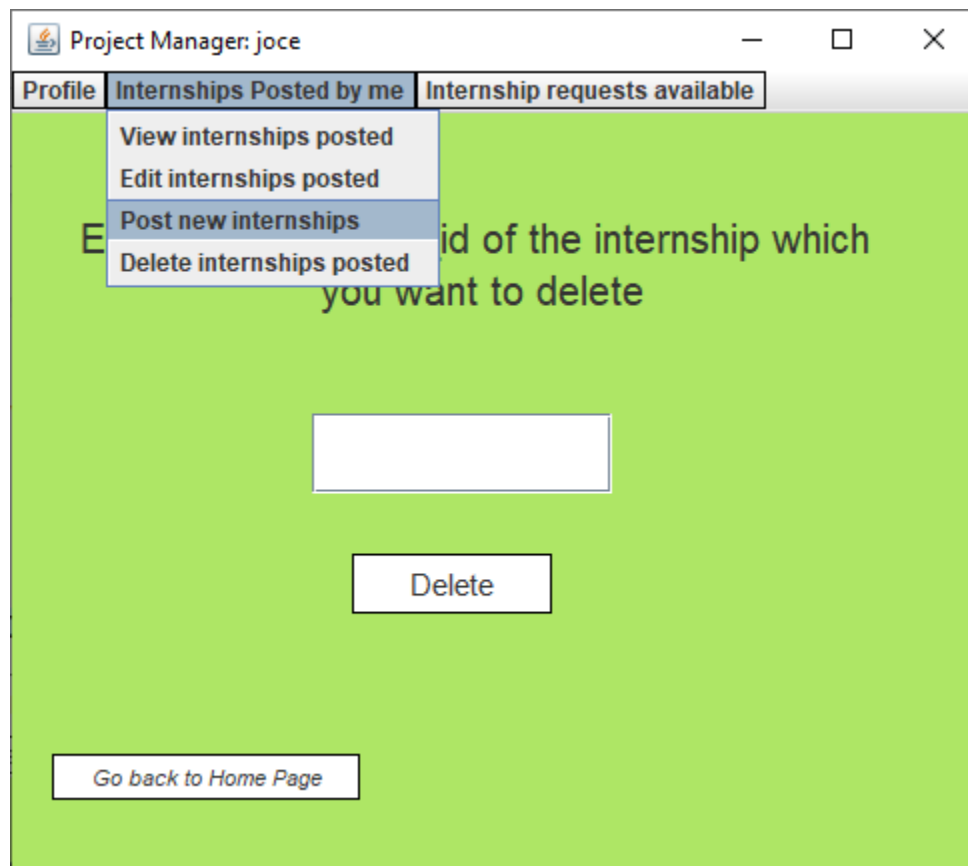


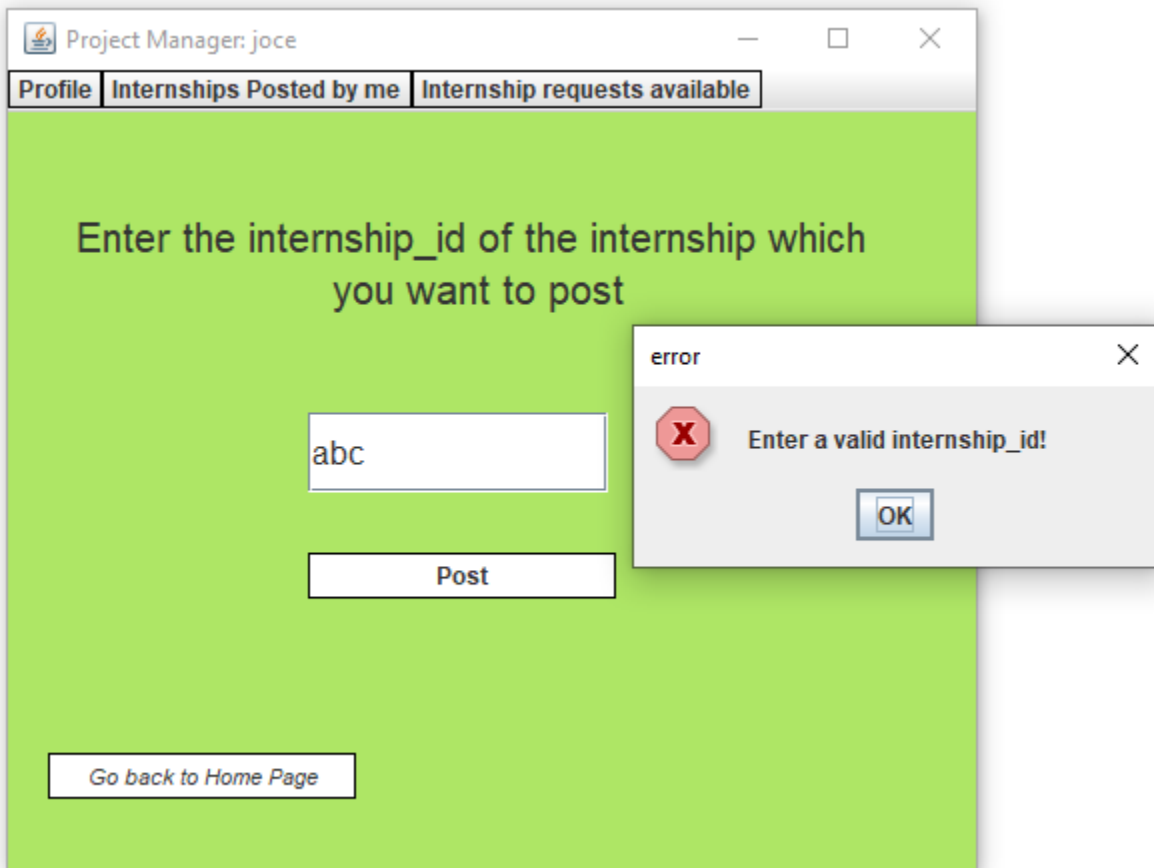
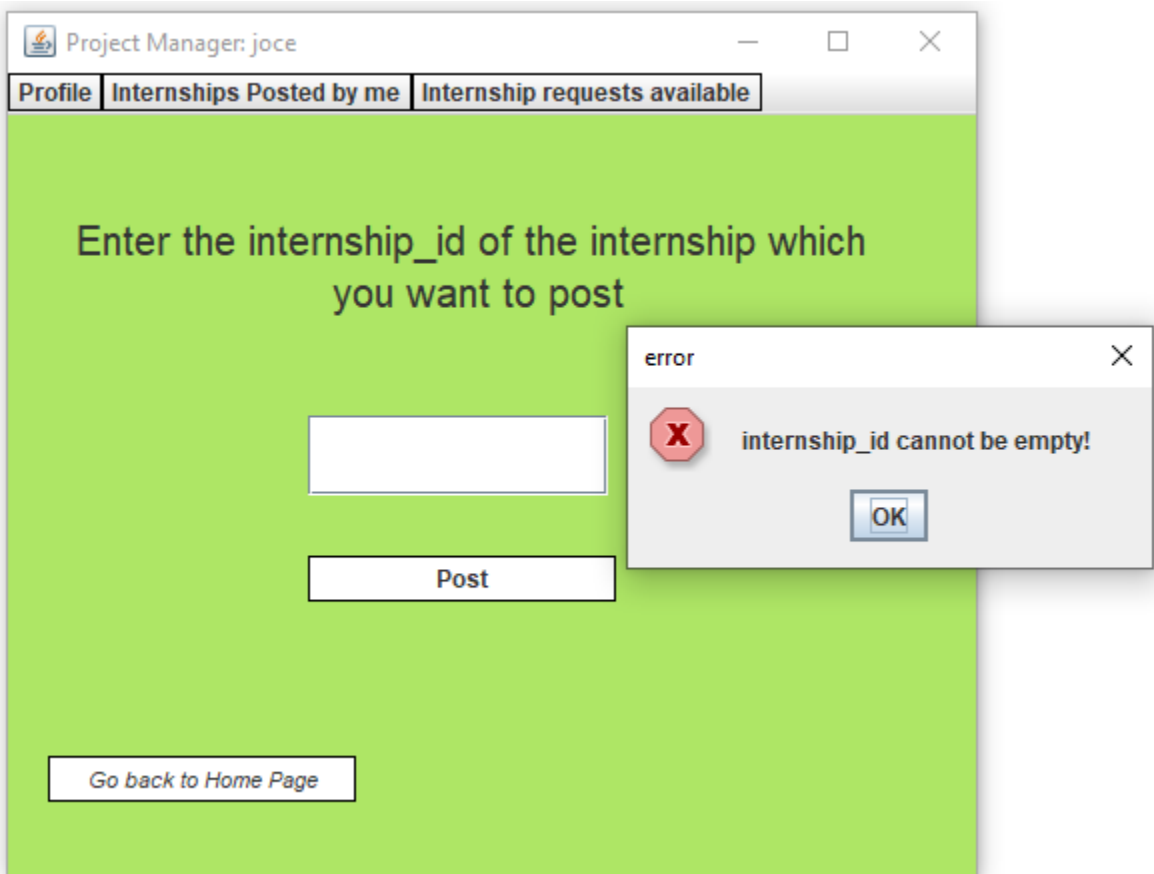
When a valid internship_id is entered, that the project manager has posted, it gets deleted successfully from the database.

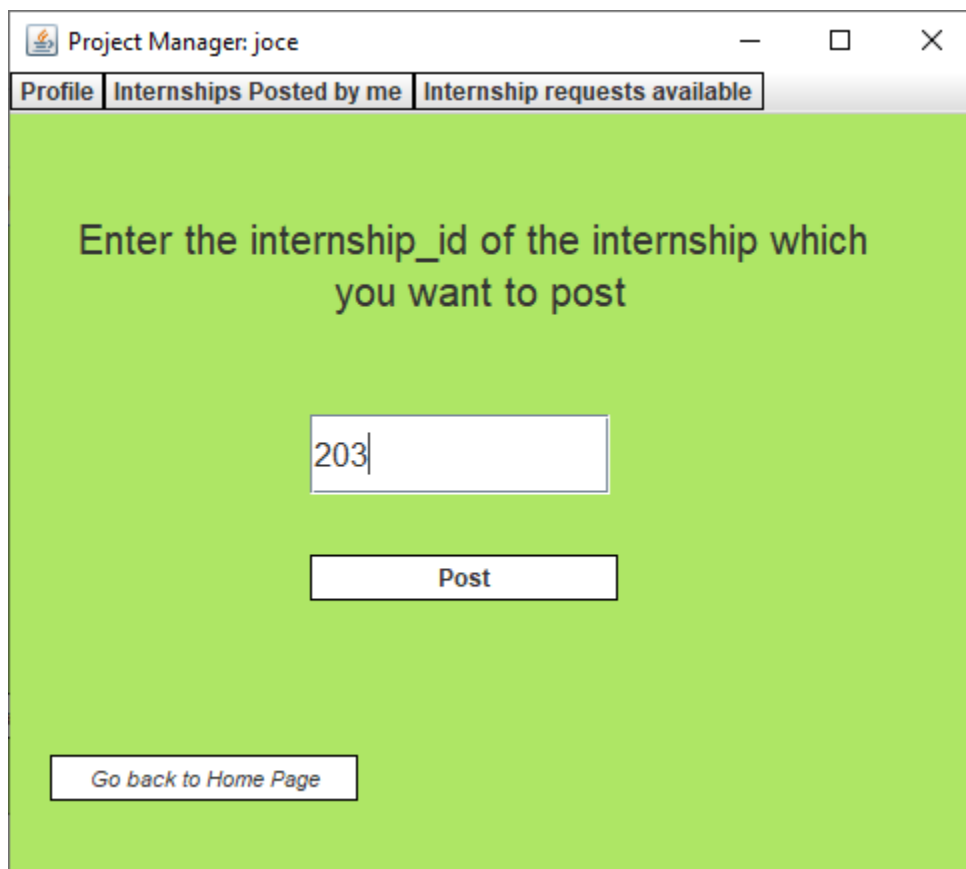
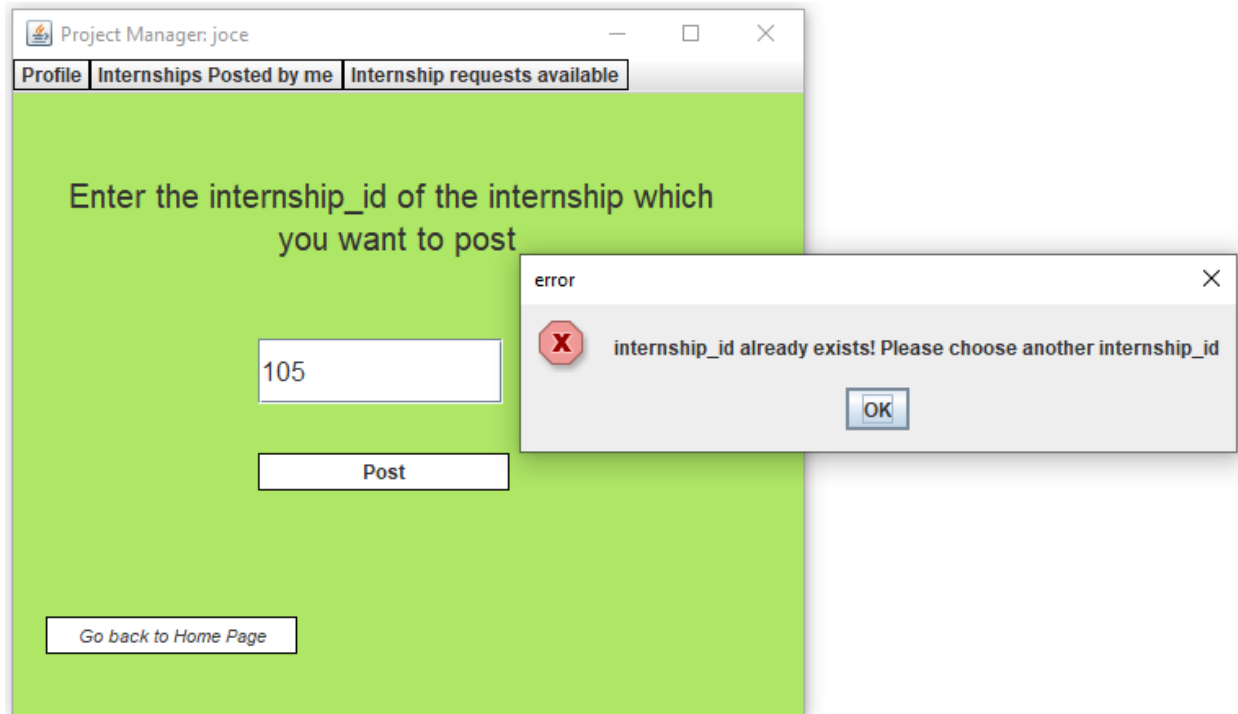




Post new internships by a project manager







When we enter a valid internship_id and follow all the constraints, it gets inserted into the database successfully.

Project Manager: joce

Profile Internships Posted by me Internship requests available

Internship_id: 203

Role: software trainee

Salary: 15000

Location: work from home

Skills Required: DSA

Post

[Go back to Home Page](#)

Project Manager: joce

Profile Internships Posted by me Internship requests available

Enter the internship_id of the internship which you want to post

Message

Posted successfully!

OK

[Go back to Home Page](#)

View internships requests for a project manager

Project Manager: joce

Profile

Internships Posted by me

Internship requests available

View internships requests

Approve internship requests

Username:

joce123

First name:

joce

Last name:

j

Company name

oracle

Phone number:

1223456789

Designation:

manager

Go back to Home Page

Project Manager: joce

Profile

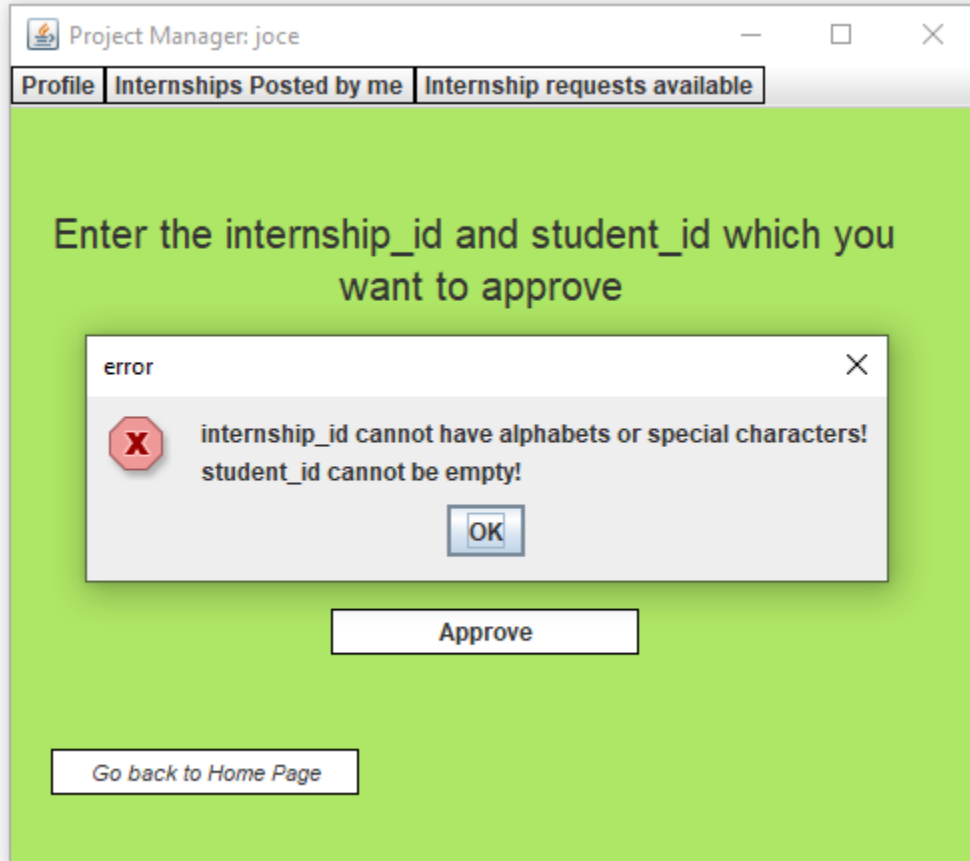
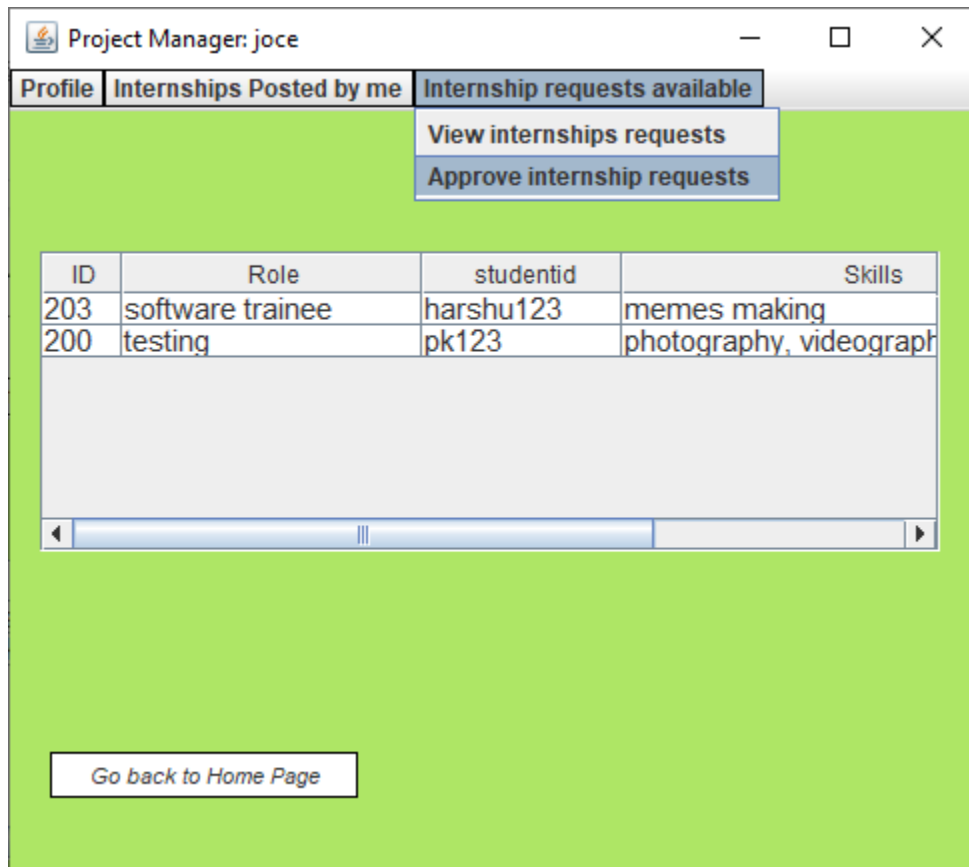
Internships Posted by me

Internship requests available

ID	Role	studentid	Skills
203	software trainee	harshu123	memes making
200	testing	pk123	photography, videograph

Go back to Home Page

Approve internship requests for a project manager




Project Manager: joce

Profile | Internships Posted by me | **Internship requests available**

Enter the internship_id and student_id which you want to approve

error


 internship_id does not exist!
student_id does not exist!

Project Manager: joce

Profile | Internships Posted by me | **Internship requests available**

Enter the internship_id and student_id which you want to approve

error

 This internship_id has not been posted by you, you cannot approve it!

Project Manager: joce

Profile Internships Posted by me Internship requests available

Enter the internship_id and student_id which you want to approve

103 harshu123

Approve

Go back to Home Page

error

This student has not applied to this internship!

OK

Project Manager: joce

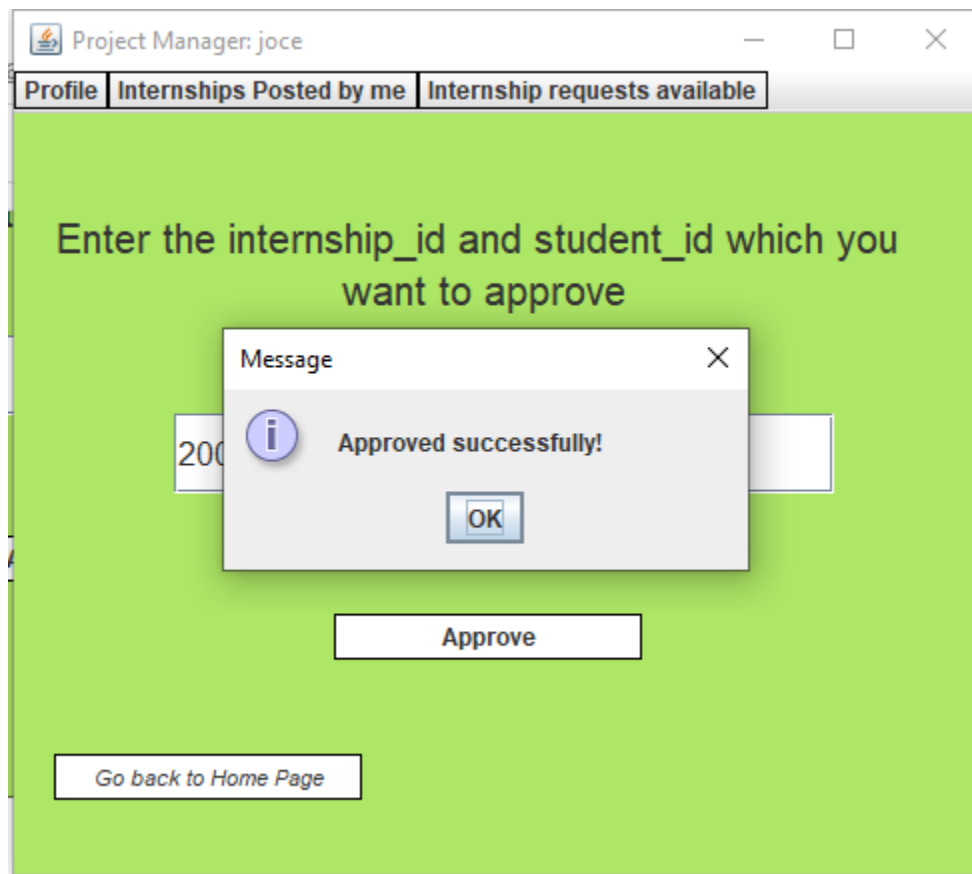
Profile Internships Posted by me Internship requests available

Enter the internship_id and student_id which you want to approve

200 pk123

Approve

Go back to Home Page



RESULTS

I have successfully completed the mini-project “*InternHub*” .

DISCUSSION AND FUTURE WORK

This project contains the basic interaction of applying for internships by students and hiring of interns by project managers. It has a very basic user interface. Future scope would be to make the UI more appealing by using graphics. One more feature would be to allow student-users to upload their resumes and official documents required so that the project managers can take a look while they hire them. We can also think of including a feedback system to allow the users to leave their valuable feedback after using this app. Making this feedback to be publicly viewable, would attract many more users to use this app.

REFERENCES

- <https://docs.oracle.com/javase/7/docs/api/>
- <https://www.javatpoint.com/java-swing>
- <https://stackoverflow.com/>