



NEW HORIZON
COLLEGE OF ENGINEERING

Autonomous College Permanently Affiliated to VTU, Approved by AICTE & UGC
Accredited by NAAC with 'A' Grade.

RECRUITMENT AND TRAINING DATABASE APPLICATION

A MINI PROJECT

REPORT

Submitted by

RAMYASHREE S

*In partial fulfillment for the award of
the degree of*

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING



**NEW HORIZON
COLLEGE OF ENGINEERING**

Autonomous College Permanently Affiliated to VTU, Approved by AICTE & UGC
Accredited by NAAC with 'A' Grade.

Certificate

This is to certify that the mini project work titled

RECRUITMENT AND TRAINING DATABASE APPLICATION

*Submitted in partial fulfillment of the degree of Bachelor of Engineering
in Computer Science and Engineering*

Submitted by

RAMYASHEE

1NH18CS156

DURING

EVEN SEMESTER 2019-2020

For

19CSE48

Signature of Reviewer

Signature of HOD

SEMESTER END EXAMINATION

Name of the Examiner

Signature with date

1. _____

2. _____

Recruitment and Training Database Application

ORIGINALITY REPORT

6%

SIMILARITY INDEX

%

INTERNET SOURCES

6%

PUBLICATIONS

%

STUDENT PAPERS

PRIMARY SOURCES

1

Jeff Friesen. "Chapter 5 Writers and Readers", Springer Science and Business Media LLC, 2015

Publication

1%

2

Vaskaran Sarcar. "Interactive Object-Oriented Programming in Java", Springer Science and Business Media LLC, 2020

Publication

1%

3

S G Ganesh, Tushar Sharma. "Oracle Certified Professional Java SE 7 Programmer Exams 1Z0-804 and 1Z0-805", Springer Science and Business Media LLC, 2013

Publication

1%

4

Premkumar. "Inheritance", Beginning JavaFX™, 2010

Publication

<1%

5

Daniel Solis, Cal Schrottenboer. "Illustrated C# 7", Springer Science and Business Media LLC, 2018

Publication

<1%

ABSTRACT

Employability is highly important in the current scenario of higher education .Given the increase in fee and students debt ,graduates are in hunt for a good job at the end of their course. Work placement and work-related learning is a great way to enhance employability. Students prefer colleges based on the placement capabilities. So it is necessary for any institution to have a department to deal with placement activities.

For a placement department it is necessary to deal with lots of data. In the existing system Training Agencies and Students records are added manually in the books or in Excel sheets. Student details have to be stored in separate records. There is a need to maintain all noted papers, causing large amount of space. As it is manually done, chances of missing, difficult to handle the details of student. Placement officers register the information of Agencies and Students. If any new Agency wants to register it should be done manually. This process is tedious and time consuming, lack of security of data, took more man power, and consumes large volume of paper and space. This process is so difficult when number of Agencies and Student increases. Though using Excel makes job easier to maintain data,it is tedious to prepare the sheet and take entries. It is not very pleasing to operate the data in Excel sheets.

So we need a Desktop application, that is user friendly ,effective and easier to use .It must provide easy methods to operate with data. It must have the count of registrations. The data received must be stored in files and databases .

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be impossible without the mention of the people who made it possible, whose constant guidance and encouragement crowned our efforts with success.

I have great pleasure in expressing gratitude to **Dr.Mohan Manghnani**, Chairman of New Horizon Educational Institutions for providing necessary infrastructure and creating good environment

I take this opportunity to express my profound gratitude to **Dr.Manjunatha**, Principal NHCE, for his constant support and encouragement.

I am grateful to **Dr.Prashanth C.S.R**, Dean Academics, for his unfailing encouragement and suggestions, given to me in the course of my project work.

I would also like to thank **Dr. B. Rajalakshmi**, Professor and Head, Department of Computer Science and Engineering, for her constant support.

I express my gratitude to **Ms.Lakshmi**, assistant professor, my project guide, for constantly monitoring the development of the project and setting up precise deadlines. Her valuable suggestions were the motivating factors in completing the work.

RAMYASHREE S

1NH18CS156

CONTENTS

	Page no.
ABSTRACT	I
ACKNOWLEDGEMENT	II
LIST OF FIGURES	V
1. INTRODUCTION	
1.1. PROBLEM DEFINITION	1
1.2. OBJECTIVES	2
1.3. METHODOLOGY TO BE FOLLOWED	2
1.4. EXPECTED OUTCOMES	2
1.5. HARDWARE AND SOFTWARE REQUIREMENTS	3
2. OBJECT ORIENTED CONCEPTS	
2.1. CLASS	4
2.2 OBJECT	4
2.3. INHERITANCE	4
2.4. POLYMORPHISM	8
2.5 ABSTRACT CLASS	9
2.6 MULTITHREADING	11
2.7 I/O FUNCTIONS	13
2.8 JAVA PACKAGES	14
2.9 EXCEPTION HANDLING	15
3. DESIGN	
3.1. DESIGN GOALS	16

3.1.1	MySQL Implementation	16
3.1.2	JAVA DATABASE CONNECTIVITY	17
3.1.3	JAVA VECTOR CLASS	19
3.1.4	Java AWT	19
3.1.5	Java Swing	20
3.1.6	Project Outline	20
3.2.	ALGORITHM/ PSEUDOCODE	21
4. IMPLEMENATION		
4.1.	MODULE 1 FUNCTIONALITY	22
4.2	MODULE 2 FUNCTIONALITY	25
4.3	MODULE 3 FUNCTIONALITY	26
4.4	MODULE 4 FUNCTIONALITY	27
4.5	MODULE 5 FUNCTIONALITY	28
5. RESULTS		
5.1	WELCOME PAGE	29
5.2	LOGIN PAGE	29
5.3	MENU PAGE	31
5.4	STUDENT MENU	32
5.5	TRAINEE MENU	38
5.6	RECRUITER MENU	40
6. CONCLUSION		44
7. REFERENCES		44

LIST OF FIGURES

<u>Fig.No.</u>	<u>Figure Description</u>	<u>Page no.</u>
1.	Single Inheritance	5
2.	Multilevel Inheritance	5
3.	Hierarchical Inheritance	6
4.	Multiple Inheritance	6
5.	Hybrid Inheritance	7
6.	Welcome page	29
7.	Login page	29
8.	Wrong Username or Password	30
9.	Exiting Login page	30
10.	Menu Panel	31
11.	Number of Registrations	31
12.	Student Menu	32
13.	Student Registration	32
14.	Registered Students	33
15.	Student Edit Panel	33
16.	Student Delete	34
17.	Student Deleted	34
18.	Student to be deleted not found	34
19.	Student Search	35
20.	Individual search successful	35
21.	Individual search unsuccessful	35
22.	CGPA Search Input	36
23.	CGPA Search Output	36
24.	Backlogwise search	37
25.	Branchwise Search	37
26.	Trainee Menu	38
27.	Trainee Registration Panel	38
28.	Trainee Delete	39
29.	Trainee Delete Unsuccessful	39

30.	Recruiter Menu	40
31.	Recruiter Registration	41
32.	Text file on Company Information	41
33.	Drives	42
34.	Placed Student Input	42
35.	Text file on Placed Students	43
36.	Placed Students	43

CHAPTER 1

INTRODUCTION

1.1 PROBLEM DEFINITION

Employability is highly important in the current scenario of higher education .Given the increase in fee and students debt ,graduates are in hunt for a good job at the end of their course. Work placement and work-related learning is a great way to enhance employability. Students prefer colleges based on the placement capabilities. So it is necessary for any institution to have a department to deal with placement activities. For a placement department it is necessary to deal with lots of data.

In the existing system Training Agencies and Students records are added manually in the books or in Excel sheets. Student details have to be stored in separate records. There is a need to maintain all noted papers, causing large amount of space. As it is manually done, chances of missing, difficult to handle the details of student. Placement officers register the information of Agencies and Students. If any new Agency wants to register it should be done manually. This process is tedious and time consuming, lack of security of data, took more man power, and consumes large volume of paper and space. This process is so difficult when number of Agencies and Student increases. Though using Excel makes job easier to maintain data,it is tedious to prepare the sheet and take entries. It is not very pleasing to operate the data in Excel sheets.

So we need a Desktop application, that is user friendly ,effective and easier to use .It must provide easy methods to operate with data. It must have the count of registrations. The data received must be stored in files and databases .

1.2 OBJECTIVES

- Secured application of recruiting cell.
- Organising and managing data of the applicants ,training heads and companies.
- Easy to segregate the pool of applicants in various criteria .
- Tracking of training status of selected applicants.
- Effective and efficient data management.
- Shortlisting of placed students
- Maintain records of students separately after placement in files.

1.3 METHODOLOGY TO BE FOLLOWED

- Develop a desktop application that is attractive and user friendly by using GUI support of java.
- Develop panels to construct the project by using Java Swing and AWT.
- To develop other panels,we use Object oriented concepts of Java.We create a class that sets the basic attributes of a panel like size,bacakground image,icon,font and size, that is common to all other panels.The members of this class is inherited by other classes to create panel and the methods are overridden.
- A JDBC must be established to save the data of the students,trainees and recruiters registered using the desktop application.Detailed records must be written onto files.
- The registered data must be secured.The operation of data must be well handled .The data in database is displayed by using Java Collection Vector.
- The project must display the number of registration at any point of time during running of the application.This is possible by multithreading.

1.4 EXPECTED OUTCOMES

- Secured login to the application.
- Registration forms for student,trainee and company.
- Number of registrations of each category displayed at any point of time during running of the application.
- Operations like search (individual,cgpa,backlogswise and brandwise) and delete amongst

students possible.

- Deletion and Displaying of registered students, trainees and companies possible.
 - The information regarding company drives is maintained in a file.
- The placed students is maintained separately along with placement details in a file and database.

Hardware Requirements

- System : Intel-3
- Hard Disk : 250 GB
- Ram : 4 GB
- Speed : 1.1GHz

Software Requirements

- Operating System : Windows 10
- Technology : Jdk1.8.0
- Tools : NetBeans 8.2
- Database : MySQL.

CHAPTER 2

OBJECT ORIENTED PROGRAMING

2.1 CLASS

Class can be defined as a blueprint or prototype or a set of instructions that builds a specific type of object. It is a basic concept of OOP which determines how an object must behave and what it must contain. Class defines the variables and the methods or functions that are common to all objects of a certain kind. The principles for creating a class is as follows:

- **Single Responsibility Principle** : Class should have one reason only to change.
- **Open Closed Responsibility** : Class should be able to extend any classes without modifying it.
- **Liskov Substitution Responsibility** : Child classes must be substitutable for their parent class.
- **Dependency Inversion Principle**: Dependency not on concretions but on abstraction.
- **Interface Segregation Principle** : The interfaces should be client specific.

2.2 OBJECT

Object can be called as an instance or a specimen of a class. In programming point of view, it can be considered as data structure, a variable or a function. It is designed based on class and has a memory allocated. It is a self-contained component consisting of methods and members to make a certain type specific.

2.3 INHERITANCE

This is a fundamental feature of an Object-Oriented programming. It is the process of creating a new Class, called the Derived/Super/Parent Class, from the existing class, called the Base/Sub/Child Class. It is a very elegant way to reuse and modify the data and functionality that has already been defined in the Child Class, also you can add new data and functionality to the Parent Class. Since the Child Class inherits all properties of the Parent Class, the subclass has a larger set of properties than the super class. However, the Child Class may override some or all the properties of the Base Class.

Advantages:

- ❖ The reusability of code, rather than developing new objects from scratch, new code can be based on the work of other developers, adding the new properties only that are needed.
- ❖ Reuse of existing classes saves time and effort. When we create a class to derive from another class, the derived class implicitly gains all the data and functionality of the parent class, except for its constructors and destructors.

TYPES OF INHERITANCES :

There can be three types of inheritance in java on basis of class :

- Single
- Multilevel
- Hierarchical

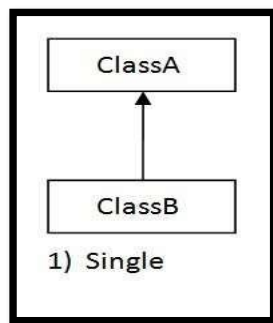


Fig no.1 Single Inheritance

SINGLE INHERITANCE :

Single subclass inherit the features of one superclass. In above image, subclass B inherits features of superclass A.

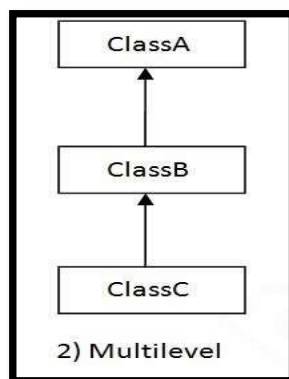


Fig no.2 Multilevel Inheritance

MULTILEVEL INHERITANCE :

A derived class will be inheriting a super class and as well as the derived class also act as the base class to other class. In above image, the class A serves as a base class for the derived class B, which in turn serves as a base class for the derived class C. In Java, a class cannot directly access the grandparent's variables and methods.

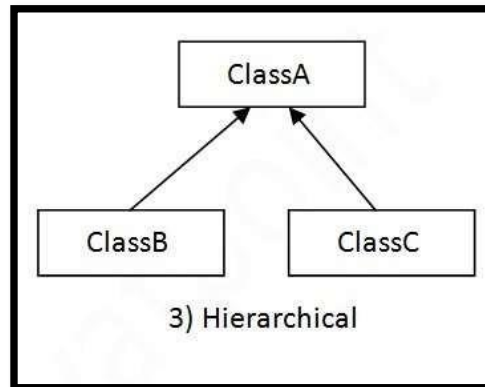


Fig no.3 Multilevel Inheritance

HIERARCHICAL INHERITANCE:

Single class serves as a base class for more than one sub class. In above image, the class A serves as a base class for the derived class B and C.

We can achieve multiple and hybrid Inheritance, on the basis of Interface.

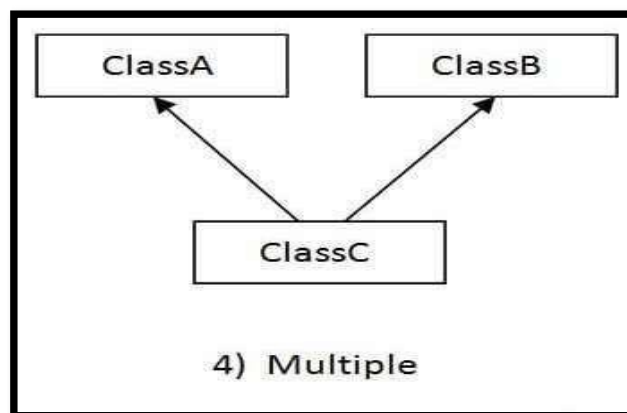


Fig no.4 Multiple Inheritance

MULTIPLE INHERITANCE:

Single class can have more than one superclass and inherit features from all parent classes. Java does not support multiple inheritance with classes. In image above, Class C is derived from interface A and B.

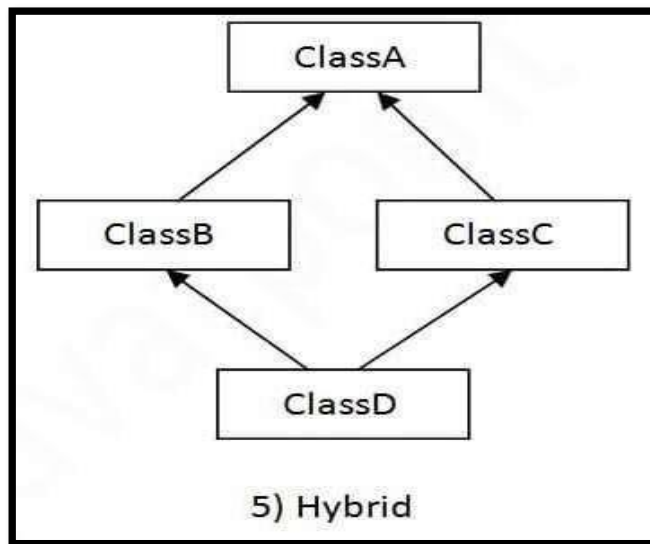


Fig no.5 Hybrid Inheritance

HYBRID INHERITANCE

This is a mix of two or more of the above types of inheritance. Java doesn't support multiple inheritance with classes, therefore hybrid inheritance is also not possible with classes. We can achieve hybrid inheritance only through Interfaces in Java.

ADDITIONAL FEATURES OF INHERITANCE:

- **Default superclass:** In the absence of any other explicit superclass, every class is implicitly a subclass of Object class(Single inheritance).
- **Superclass can only be one:** A superclass can have any number of derived classes. But a subclass can have only one superclass(exception:interfaces). This is because Java does not support multiple inheritance with classes, but possible through interfaces.
- **Inheriting Constructors:** A derived class inherits all the members (fields, methods, and nested classes) from its superclass. Constructors are not inherited by subclasses, but the constructor of the superclass can be invoked from the subclass by using "super" keyword.

Usage of super keyword:

- I. Use of super with variables: When base and derived class have same data members, there is a possibility of ambiguity for the JVM. To refer to base class data member in derived class, we use “super” keyword.
- II. Use of super with methods: When base and derived class have same method, there is a possibility of ambiguity for the JVM. To refer to base overridden method in derived class, we use “super” keyword.
- III. Use of super with constructor: “super” keyword can also be used to access the parent class parameterised or non parameterised constructor.

Private member inheritance: A subclass does not inherit the private members of its base class. If the superclass has public or protected methods for accessing its private fields, these can also be used by the subclass/base class.

2.4 POLYMORPHISM

Polymorphism is a OOPS concept by which we can perform a single action in different ways. Polymorphism is derived from 2 Greek words, namely, poly(many) and morphs(forms). There are two types of polymorphism in Java: compile-time runtime polymorphism(overriding) and polymorphism(overloading).

❑ **COMPILE-TIME /STATIC POLYMORPHISM:** Method overloading is one of the static polymorphism. It allows us to have more than one method having the same name. The advantage of method overloading is it increases the readability of the program. There are two ways to overload the method in java

- I. By changing number of arguments
- II. By changing the data type

In java, this is not possible by changing the return type of the method only.

❑ **RUNTIME POLYMORPHISM/DYNAMIC METHOD DISPATCH**

Method Overriding is dynamic polymorphism. In this the subclass declares a method which is already declared in parent class. The method in base class is overridden. To call the overridden method, we use “super” keyword

2.5 ABSTRACT CLASS

Data abstraction is one of the key principles of OOP concept, which can be defined as the process of hiding certain details and showing only essential information to the user. We can achieve abstraction by abstract classes or by using interfaces.

❖ Abstract class

- We use “**abstract**” keyword which is a non-access modifier used for classes and methods.

- **Abstract class:** It is a restricted class that does not allow to create objects. Creation of objects leads to accessing of it. Therefore, it must be inherited from other class.
- **Abstract method:** It can be used only in an abstract class. Otherwise we can also tell that the class with at least one abstract method, must be abstract. Abstract methods do not have body or the definition is not provided in the method. Instead the bodies provided in the derived class.
- An abstract class can have both abstract methods and regular methods.
- We use abstract classes to achieve security. It allows to hide certain details and only show the important object that must be viewed to the user.

❖ Interface

- We can define interface as a complete abstract i.e. It has methods with empty bodies.
- Interfaces can have only abstract methods but no regular methods allowed unless they are default in accessibility.
- We cannot create objects, therefore, to implement these abstract methods we must implement this interface in another class. The keyword used is “implements”.
- It cannot have a constructor.
- Implementation of interfaces in turn override all of its methods.
- By default, its methods are abstract and public.
- By default, its attributes are public, static and final.
- We use interfaces to achieve security. It allows to hide certain details and only show the important object that must be viewed to the user.
- Interfaces also helps in multiple inheritance that Java does not support through regular classes.

2.6 MULTITHREADING

- It is a Java feature that allows concurrency in execution of two or more parts of a program. This helps in the maximum utilization of CPU.
- A thread is a smallest unit of dispatchable code that defines a separate path of execution. It is a light weight subprocess.
- Therefore, multithreaded programming is a process of executing multiple threads simultaneously.
- Life cycle of thread has five stages and it is controlled by Java Virtual Machine. The stages are New, Runnable, Running, Non-Runnable and Terminated.

❖ Creation of thread:

a) Extending a thread class:

We can inherit a class from java.lang.Thread class. This class overrides the run() method. As soon as we call start() method, the execution of the thread starts. Start() method invokes the run() method in which the functioning of the thread begins.

b) Implementing the Runnable interface:

We can create and implement class of java.lang.Runnable interface. This overrides run() method and instantiate a Thread object, after call start () method on this object.

- Benefits of using Runnable interface over Thread class: As Java does not support multiple inheritance, extending of Thread class does not allow to inherit any other base class. Therefore, by implementing the Runnable interface, the new class can still extend the other base classes.
- Benefits of using Thread class over Runnable class: Thread class has inbuilt methods like yield(), interrupt(),etc. that are not available in Runnable interface. Therefore, we can achieve basic functionality of a Thread by extending Thread class.
- The most common constructor is Thread(Runnable thread-object, String thread name).

- The Thread reference returns the name of the Thread, its priority and its Thread group name. the priority level changes from 1(MIN_PRIORITY) to 10(MAX_PRIORITY). By default, the priority is 5 (NORM_PRIORITY).
- The most often used methods of Thread class are get Name(), get Priority(), is Alive(), join(), run(), sleep(), start(), etc.
- **Synchronization:** To solve the concurrent issues we need to synchronize the action of multithreads and make sure that only one thread can access the shared at the given point of time. This involves the concept called monitors. Each object in Java is associated with a monitor which decides whether a thread can lock or unlock. At a given time on a monitor only one thread may hold a block. We use the keyword “synchronized” before the shared resource to bring concurrency.
- **Interthread Communication :** The concept is used to develop a program where two or more threads exchange information. This is achieved by three methods in synchronized context mentioned as follows:
 - a. **public void wait():** It makes current thread to wait until another thread invokes the notify()
 - b. **public void notify():** It wakes up only one Thread on this object’s monitor.
 - c. **public void notifyAll():** It wakes up all the threads that called wait() on this object.
 - Sometimes it is important to develop an application in keeping mind about deadlock situations.
 - The major operations are suspend(), stop(),resume(), wait() and notify().

2.7 I/O FUNCTIONS

- Java input and output functions helps to process the input and produce the output. The java.io package has all the classes required for input and output operations.
- It uses the concept of Stream to enhance the speed of I/O operations.
- **Stream classes:** The stream based I/O is built upon four abstract classes, namely InputStream, OutputStream, Reader and Writer.
- **Byte Stream classes:** Byte Streams provide a way to handle input and output of bytes. We have two abstract classes, InputStream and OutputStream.

Few Inbuilt Classes are:

- a) **InputStream:** It is an abstract class that define model of streaming byte input. It implements Closeable interface and most of his method will throw IOException on error conditions.
- b) **OutputStream:** It is an abstract class that define model of streaming byte output. It implements both the Closeable and Flushable interfaces and most of its methods throw an IOException in the case of errors.
- c) **FileInputStream:** This class creates an Input Stream that we can use to read bytes from a file. Its methods can throw FileNotFoundException or IOException.
- d) **FileOutputStream:** This class creates an Output Stream that we can use to write bytes from a file. Its methods can throw FileNotFoundException or IOException.
- e) **ByteArrayInputStream:** This class contains all the buffers, containing bytes to be read from the Input Stream. No IO exception is thrown.
- f) **ByteArrayOutputStream:** This class contains all the buffers, containing bytes to be read from the Output Stream. No IO exception in thrown.
- g) **FileWriter :** Class useful to create a file writing characters into it.
- h) **FileReader :** Class useful to read data in the form of characters from a 'text' file.
- i) **BufferedReader:** Class that helps to read text from a character-input stream, buffering characters so as to provide for the efficient reading of characters, arrays, and lines.
- j) **BufferedWriter:** Class that helps to write text from a character-input stream, buffering characters so as to provide for the efficient writing of characters, arrays, and lines.

2.8 Java Packages

- It is a container that encapsulate the group of related classes, sub packages and interface.
- It helps in preventing the naming conflicts between the class names in different packages. It makes searching and usage of classes, interfaces, enumerations and annotations easier.
- It provides controlled access. protected and default have package level access control. A protected member can be accessed by classes in the same package or in its sub classes. A default member is accessible by classes present in the same package only.
- Java packages provide data encapsulation/hiding.

They can be divided into two categories:

a) Built-in packages : Packages from Java API that has prewritten classes, which are free to use and are included in Java development environment. This library contains components for managing input, database programming, etc. We can either import a single class or a whole package that contain all the classes that belong to the specify package. To use a class or a package from the library we use the keyword “import”.

b) User defined packages : These packages are created by the user. To create a package we use a keyword “package”.

- Advantages of Java package
 - I. It categorises the classes and interfaces so that they can be easily maintained.
 - II. It provides access protection.
 - III. It removes naming collision.
- Some of the important packages of Java API are
 - I. **lang:** It is automatically imported and contains language supports.
 - II. **io:** It contains classes for input and output operations.
 - III. **util :** It contains utility classes to implement data structure.
 - IV. **applet :** It contains classes that create applets.
 - V. **awt:** It contains classes that implements components of Graphical User Interface.
 - VI. **net:** It contains classes that support networking operations.

2.9 Exception Handling

- Exception can be defined as an event that interrupts the normal flow of execution of a program. We have compile time and run time exceptions.
- In order to have smooth flow in execution we must need to do robust programming which takes care of exceptional situations. Code written to achieve the same is referred to as exception handler.
- Java exception hierarchy is inherited by two subclasses: Exception and Error, whose root class is java.lang.Throwable class.
- And exception might be checked exception, unchecked exception or error.
- To achieve Java exception handling we use five keywords.
 - I. try: The keyword “try” is used to specify block where the code might have an exception. We cannot use try block alone, instead it must be followed by catch or finally.
 - II. catch: The keyword “catch” is used to specify the block of code which handles the exception caught in the preceding try block. It can be followed by finally block.
 - III. finally: The keyword “finally” is used to specify the block of code which must be executed whether an exception is handled or not.
 - IV. throw: The keyword “throw” is specified to throw an exception.
 - V. throws: The keyword “throws” is specified to declare exceptions that might occur in the method. It is specified with method signature. It does not throw any exception.

CHAPTER 3

DESIGN

3.1 DESIGN GOALS

Recruitment and Training Desktop Application requires database along with the File I/O Operations. We use Java Swing and AWT to create the frame-based GUI Application. Using the concepts of Object-Oriented Programming with Java, we can build this application in an effective and optimized way.

3.1.1 MySQL Implementation

To maintain the records of students, trainees and recruiters in an organized way, we need to create a database named 'miniproject' by executing SQL command: 'CREATE DATABASE miniproject;'. Later to create tables under this database, execute the command: 'USE miniproject;'. To create the further tables used in the project execute the below queries.

```
1) create table STUDENT  
  
  ( ST_ID VARCHAR(40),  
  
    USN VARCHAR(100) NOT NULL,  
  
    NAME VARCHAR(100) NOT NULL,  
  
    BRANCH VARCHAR(100) NOT NULL,  
  
    EMAILID VARCHAR(100) NOT NULL,  
  
    TOTAL_CGPA FLOAT NOT NULL,  
  
    BACKLOGS VARCHAR(10) NOT NULL);
```

2) create table TRAINEE(

TR_ID VARCHAR(40),

ORGANISER VARCHAR(40),

INSTITUTION VARCHAR(100) NOT NULL,

COURSE VARCHAR(100) NOT NULL,

PERIOD VARCHAR(100) NOT NULL,

FEE VARCHAR(100) NOT NULL,

MODE CHAR(10) NOT NULL);

3) create table REC(

REC_ID VARCHAR(40) ,

COMPANY VARCHAR(100) NOT NULL,

DRIVE_DATE VARCHAR(100) NOT NULL);

4) create table SELECTED(

ST_ID VARCHAR(40),

SELECTED_DATE VARCHAR(100) NOT NULL,

COMPANY_NAME VARCHAR(100) NOT NULL);

3.1.2 JAVA DATABASE CONNECTIVITY

JDBC is a Java API that allows one to create connection to the database and also execute queries within the database. It is a part of Java Standard Edition (JavaSE) and it uses drivers called JDBC drivers to connect to the mentioned database. The `java.sql` package contains classes and interfaces related to JDBC API. Along with Driver class, the interfaces used in this project are:

- a. **Driver interface**
- b. **Connection interface**
- c. **Statement interface**
- d. **PreparedStatement interface**
- e. **ResultSet interface**
- f. **ResultSetMetaData interface**

Steps to connect Java and MySQL:

1. Registering the Driver class

The **forName()** method is used to dynamically load the Driver class.

```
Class.forName("com.mysql.jdbc.Driver");
```

2. Establishing the connection

To establish the connection with the database we use **getConnection()** of DriverManager class.

```
Connection con=DriverManager.getConnection (URL, username, password);
```

3. Creating a Statement object

The object of statement created is used for executing the queries within the mentioned database. The JDBC offers Statement, callableStatement and PreparedStatement which enables, to send commands and receive data. We use Statement interface for general purpose access to our database, which does not require any parameters. Whereas, we use

PreparedStatement interface to give arguments or parameters to accept at runtime.

Statement stmt=con.createStatement(); which throws SQLException.

4. Executing SQL Query

To execute an SQL Statement, we have three methods: Boolean execute (String SQL), int executeUpdate (String SQL), ResultSet executeQuery (String SQL).

We most often use Java object called ResultSet which contains the results of the executed SQL Query. It is a table that maintains rows and columns, that contain the column values from the database table that the query asks for. Its object maintains a cursor which points to its current row of data. ResultSetMetaData is a class that is used to find information about the ResultSet returned from a call of executeQuery method.

5. Closing the Connection object

The **close()** method closes the connection and the ResultSet will be closed automatically.

3.1.3 Java Vector class

Java Vector class comes under util package, extends ArrayList and implements the List interfaces. It is a Java collection which maintains insertion order like ArrayList and is synchronized. The DefaultTableModel class stores the data in the JTable in a Vector object. The advantage is that during coding we don't have to add methods like add, insert or delete rows and columns as they already change the data held in the Vector. Thus, making to implement table model quick and easy.

3.1.4 Java AWT

Java abstract window toolkit is an API used to develop graphical user interface or window- based applications in Java. The components of AWT are platform independent. Since the components are using the resources of OS, they are referred to be heavyweight. The popular classes of java.awt. package is TextField, Label, TextArea. RadioButton, CheckBox, etc.

3.1.5 Java Swing

Java Swing comes under Java Foundation Classes and is built on the top of AWT API. It is entirely written in Java and provides platform independency and light weight components. The package is javax.swing. The project we used JFrame (top level container) and JPanel (light weight container). The other JComponents used are JButton, JTextField, JLabel, JRadioButton, JComboBox, JMenu, etc. The methods of Component class used are add, setSize, setLayout, etc. defined in its respective syntax.

3.1.6 Project Outline

Project is developed on the requirements needed for student, trainee or recruiter to satisfy the registration and further processes required by each member of any category. The operator of this desktop application must have a user name and password for security issues. He/ She must be able to register student, trainee and recruiter separately in a unique identification number. And also delete a registered number if required only with the help of the id number in order to solve the issues regarding privacy. He/ She should easily sort the students in various criteria to help in recruitment process. This acts as a help desk for the students to find out the available courses and the trainees dealing them, along with the details about the courses. He/ She must be able to manage the company drives and maintain the record of placed students. He/ She must be able to retrieve the number of registrations of each category up to that point of time.

3.2. ALGORITHM

1. Create a format which forms a frame and components which is common to all the other frames created in our application.

- I. Create a class first that inherits JFrame and implements ActionListener interface.
- II. Build a constructor with a parameter frame title name of type String. Add components such as background image, icon image, frame type, color and font, which is common to all frames, those are displayed in the application.
- III. Add methods which define font and color in different combinations. This helps to set the attributes of the JComponents set in the project.
- IV. Create a Abstract method to perform Event Handling, defined in its subclasses.

2. Login Page

- I. Inherit the class First.
- II. Create fields for user name and password. And also verify them.
- III. If verified, switch to menu page. Else throw an error message.

3. Menu Page

- i. Inherit the class First.
- ii. Create fields to access the categories: Students, Trainees, and Recruiters.
- iii. Give an easy way to count the number of registrations of each category on the menu page.
- iv. Similarly, by inheriting the class first, create Student, Trainee and Recruiter page that performs registrations and further related operations.
- v. Also create MySQL database and files to maintain and manage the data received during the desktop application RunTime.
- vi. List the registered entries using JTable by Vector collection.

CHAPTER 4

IMPLEMENTATION

Totally two packages are present in the project: miniproject and MyThread. The package miniproject can be classified as accessing of recruitment and training desktop application, Student related, Trainee related and recruitment related.

4.1 MODULE 1 FUNCTIONALITY:

Accessing Module

- **class Demo:**

It has the main function, which creates an object of class Welcome.

- **class First:**

This class extends /inherits JFrame and implements ActionListener. Thus, the concept of multiple inheritance is observed. It has a constructor with title name of the frame of type String. In this constructor we set the title of JFrame by calling the constructor of the immediate superclass JFrame using 'super' keyword and create objects of Font, Color, Image and Image icon. We call methods like setSize(int width, int height), setResizable(Boolean), etc. 'this' keyword is used to avoid confusions regarding the declared String 'title' and the parameter 'title'. Three methods are defined to set the attributes of the JComponent in different combinations. The class First is of the type public abstract because it has one abstract method regarding event handler, i.e., `ActionEvent` and the methods of First class is called by the class in MyThread package. This shows partial abstraction and the access modifier needed while working with more than one user defined package. The method *`void actionPerformed(ActionEvent e)`*; is abstract, therefore all sub classes which are not abstract must define this method i.e., override the method of the superclass. This is RunTime polymorphism.

- **class Welcome.**

This class inherits class First and builds a similar constructor as its parent class. It creates two objects of JLabel and one object of JButton. When this button is clicked it calls the method actionPerformed with an object of(ActionEvent). In this method we define, if a button is clicked it must go to login page (creating an object class Login). In this we have multilevel inheritance, method overriding and event handling.

- **class Login**

This class inherits class First and builds a similar constructor as its parent class. It creates three objects of JLabel to display "LOGIN", "USERNAME" and "PASSWORD". It creates two objects JText Fields to take the input of username and password. It creates two objects of JButton if the(ActionEvent) is of "OK", it verifies. If verified right, it creates an object of class Menu. If not, it sends an error message using JOptionPane. If the(ActionEvent) is of "EXIT", displays a JOptionPane with a message if one would like to exit when the project is in progress.

- **class Menu**

This class inherits class First and builds a similar constructor as its parent class. It creates an object of JMenuBar, five objects of JMenu displays "Student", "Trainee", "Recruiter", "St_Details" and "Selected", and ten objects of JMenuItem displays "Register", "List", "Delete", "St_Display", "St_Edit", "St_Register", "REGISTER", "Drives", "Insert", "View".

- **class Display**

This class helps in displaying the SQL Tables in the JTable using Vectors. We follow the steps of JDBC as mentioned earlier in SQL Implementation. The exceptions which might be occurred are ClassNotFoundException and SQL Exception. After executing the SQL query, we are able to get the output using ResultSet rs and ResultSetMetaData rsmt. Since the specific size of number of rows and columns is not constant, because it depends on the registration additions or deletions. We create a Vector column of size equal to the column count of the table required. It creates a Vector row while rs.next() is true. Vector row gets String of each column in order to store data of all columns in each row we create Vector data.

Each entry in Vector data is having each rows with all its columns information. On creating JTable with data and column on a simple JFrame will display SQL Table.

- **Exception Handling concepts:**
 - I. **NullPointerException** : thrown when an application attempts to use an object reference that has the null value.
 - II. **SQLException** :thrown to provide information on a database access error or other errors.
 - III. **IOException** : thrown when there is a failure during reading, writing and searching file or directory operations.
 - IV. **ClassNotFoundException**: thrown when the JVM tries to load a particular class and the specified class cannot be found in the class path.

MODULE 2 FUNCTIONALITY

Student Module

- **class St_Register**

This class inherits class First and builds a similar constructor as its parent class. We create an array of objects of JLabel to display “USN:”, “NAME:”, “BRANCH:”, “EMAIL-ID:”, “Total cgpa” and “BACKLOGS:”. We create objects of JTextArea to get the above details. On clicking the JButton “ADD”, we try to store these inputs given by user in SQL database. This is defined in the actionPerformed() method. On clicking “BACK” it creates an object of class menu. This class is implemented using multi-level inheritance, exception handling and method overriding.

- **class St_Edit**

This class inherits class First and builds a similar constructor as its parent class. We create two objects of JButton “DELETE” and “SEARCH” if the actionEvent is of “DELETE”, we execute an SQL query to delete the entry given as input in the SQL Table “STUDENT”. If it is of “SEARCH” we define of create an object of class St_Search.If it is of “Back”,an object of class Menu is created. If chosen “St_Display” , it creates an object if class display with an argument “SELECT * FROM STUDENT”. In this we have multilevel inheritance, method overriding and event handling.

- **class St_Search**

This class inherits class First and builds a similar constructor as its parent class. We create . We have created two objects of JRadioButton, three objects of JButton and two objects of JComboBox. We define in the event handling method to search using MySQL commands . We define individual, cgpa, branchwise and backlogwise. This will help to sort the students according to recruiters needs. In this we have multilevel inheritance, method overriding and event handling.

MODULE 3 FUNCTIONALITY

Training Module

- **class Trainee**

This class inherits class St_Register and builds a similar constructor as its parent class. We only reset the titles of the JLabels declared. The data given as the input at the JTextArea is saved to MySQL table Trainee. We assign a static number incrementor and concatenate a String with branch. This generates a unique Trainee ID. If chosen to "List", it creates an object of class Display with an argument "SELECT * FROM TRAINEE". If chosen to "Delete" we execute an SQL query whose TR_ID is given. In this we have multilevel inheritance, method overriding and event handling.

MODULE 4 FUNCTIONALITY

Recruitment Module

- **class Recruiter_Reg**

This class inherits class First and builds a similar constructor as its parent class. We create four objects of JLabel, two objects of JTextArea, two objects of JButton and one object of TextArea. We take inputs of company name, Drive date and the information regarding the company. We assign a static number incrementor and concatenate a String with branch. This generates a unique Trainee ID. The information about the company is taken from the TextArea and returning into a file "Company.txt", using an object of class BufferedWriter. If chosen "ADD" the entry is added to the database table. If chosen "BACK" a new object of class Menu created. If chosen "Drives" on the menu page, we create an object of class Display with an argument "SELECT * FROM REC". If chosen "Selected", it will ask for insert or view data. Selected is for selecting the students placed. The students placed can be inserted by creating an object of class Rec_Selected. In this we have multilevel inheritance, method overriding and event handling.

- **class Rec_Selected**

This class inherits class First and builds a similar constructor as its parent class. We create three objects of JLabel, two objects of JTextArea, one object of Calendar and two objects of JButton. We take the inputs of student registration number is placed in. If the student ID is not present, we display "Not Found" using JOptionPane. The student ID, Company name and the selected date (the date is obtained from object of the class Calendar) is inserted into MySQL table "SELECTED". If chosen "VIEW" on the menu paga, an object if class display is created with an argument "SELECTED * FROM SELECTED". The student placed is searched through St_ID and his or her details along with the company name and selected is written onto a file "Selected.txt". In this we have multilevel inheritance, method overriding and event handling.

MODULE 5 FUNCTIONALITY

Multi Threading

- **class newThread**

This is the subclass of class Menu and implements Runnable. This class is present on the other package MyThread. We create three threads of the same runnable object. Each thread counts the number of registrations in student, trainee and recruiters. The entries are displayed on the Menu Page by creating objects of JLabel. This class is instantiated when the JButton “Calculate” is clicked on the Menu Page. This class is abstract because it does not define the event handling method. The threads calculate the number of entries using SQL query "SELECT COUNT(*) FROM TABLE NAME". In this we have multiple and multilevel inheritance, exception handling, working with user defined packages, multithreading with synchronisation.

CHAPTER 5

RESULTS

5.1 WELCOME PAGE

On running the program,Welcome panel is displayed.

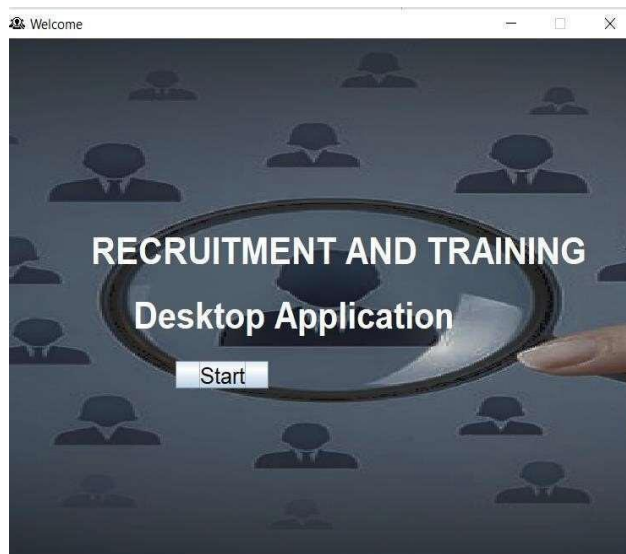


Fig no.6 Welcome page

5.2 LOGIN PAGE

On clicking "Start",Login panel is displayed.

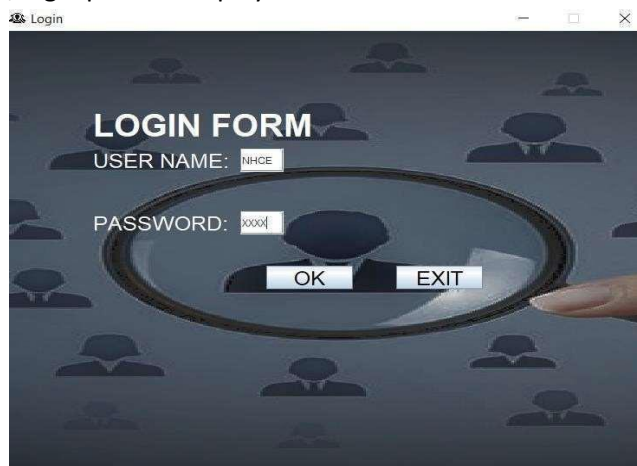


Fig no.7 Login page

On clicking “OK”,if username and password are right.Else warning message is displayed.

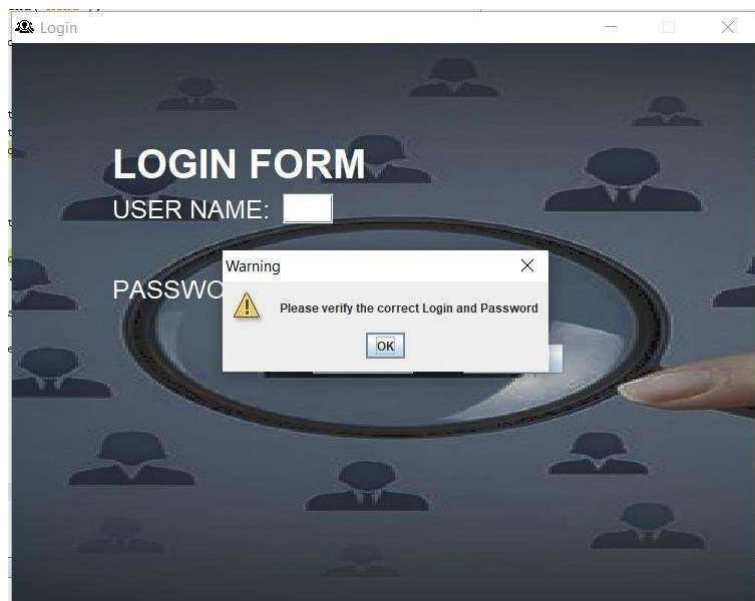


Fig no.8 Wrong Username or Password

If clicked on “Exit”,it enquires whether the user wants to quit when project is in progress.

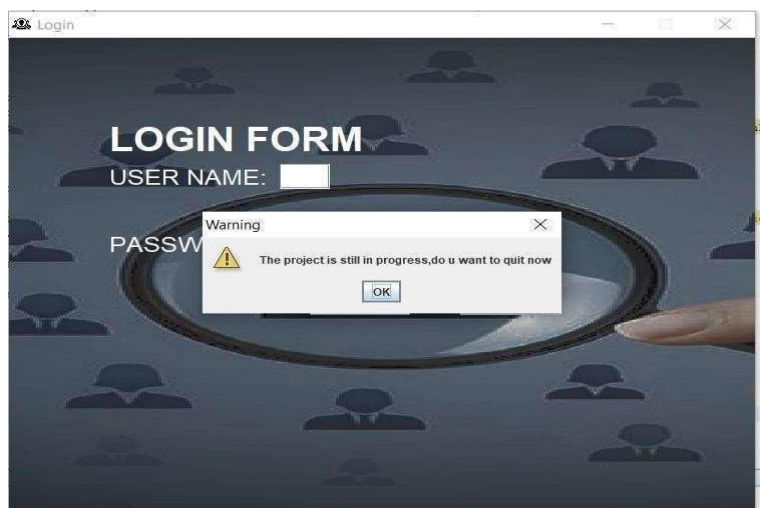


Fig no.9 Exiting Login page

5.3 MENU PAGE



Fig no.10 Menu Panel

On clicking "Calculate", number of registrations are displayed.



Fig no.11 Number of Registrations

5.4 STUDENT MENU

On clicking “Student”, Student Menu is displayed.

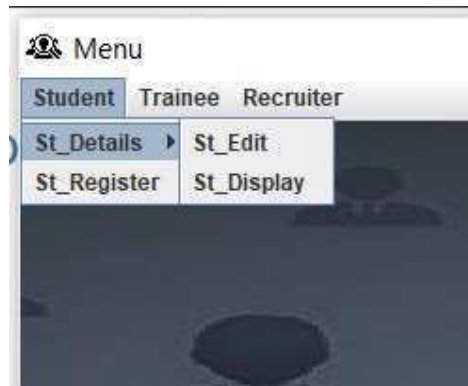


Fig no.12 Student Menu

On clicking “St_Register”, Student Registration form is displayed. It registers students, give them an ID for further procedures and stores in Student sql table.

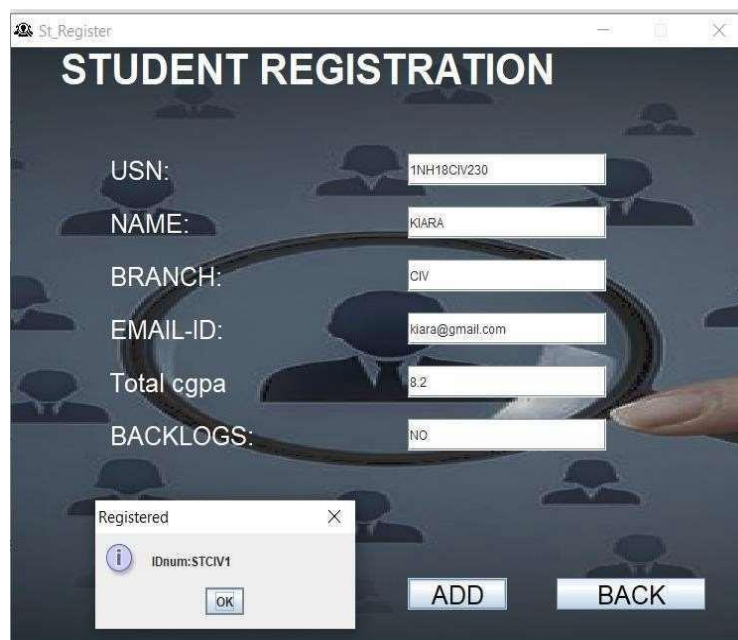
A screenshot of a software window titled "St_Register". The main area has a dark background with a pattern of silhouettes and a large magnifying glass graphic. The title "STUDENT REGISTRATION" is at the top. Below it are several input fields with labels: "USN:" (value: 1NH18CIV230), "NAME:" (value: KIARA), "BRANCH:" (value: CIV), "EMAIL-ID:" (value: kiara@gmail.com), "Total cgpa" (value: 8.2), and "BACKLOGS:" (value: NO). At the bottom right are two buttons: "ADD" and "BACK". A small "Registered" dialog box is open in the bottom left, showing an information icon, the text "IDnum:STCIV1", and an "OK" button.

Fig no.13 Student Registration

On clicking “St_Display”, list of registered students are displayed.

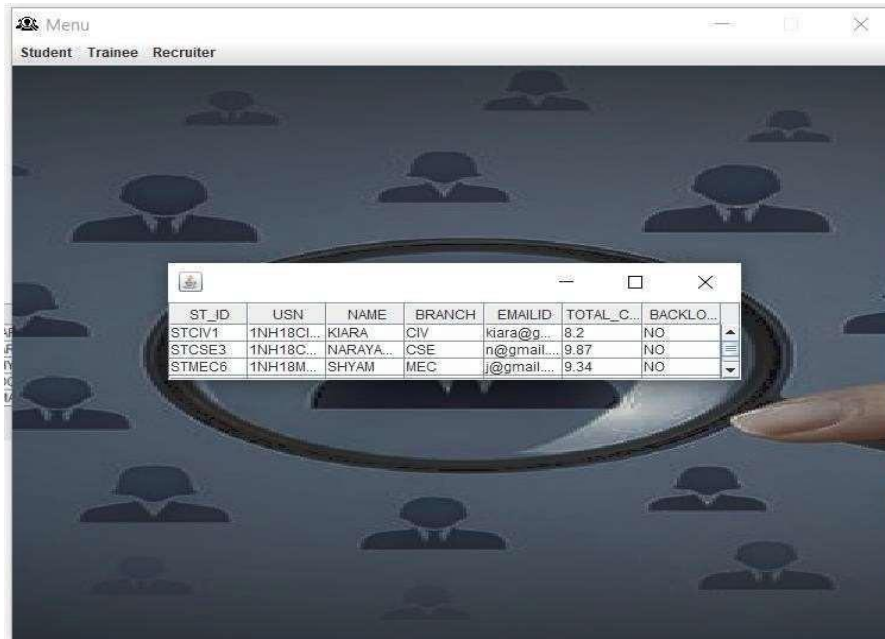


Fig no.14 Registered Students

On clicking “St_Edit”, an Edit Panel is opened.

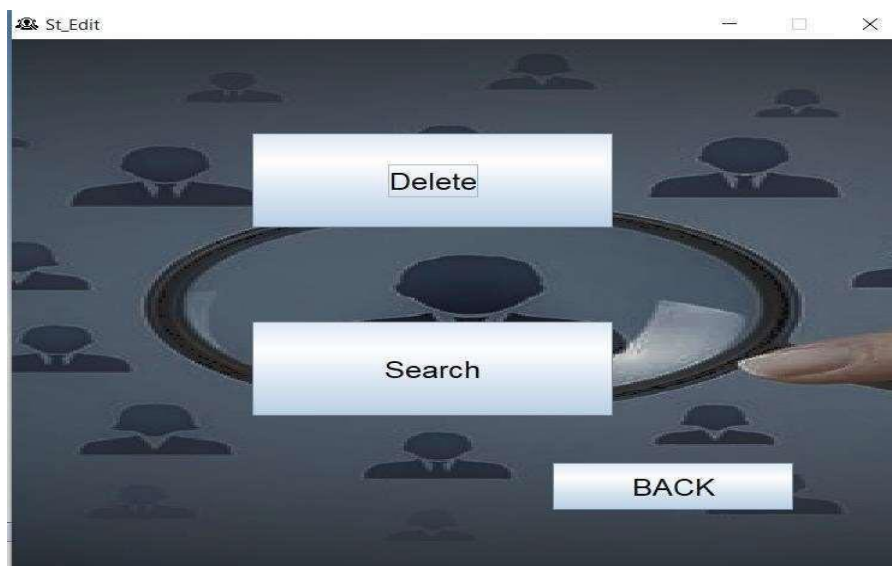


Fig no.15 Student Edit Panel

On clicking “Delete”, an input dialog box opens asking for which Student entry to be deleted. If present, the selected Student’s record will be erased. If not present, a message will be displayed “Not Found”.

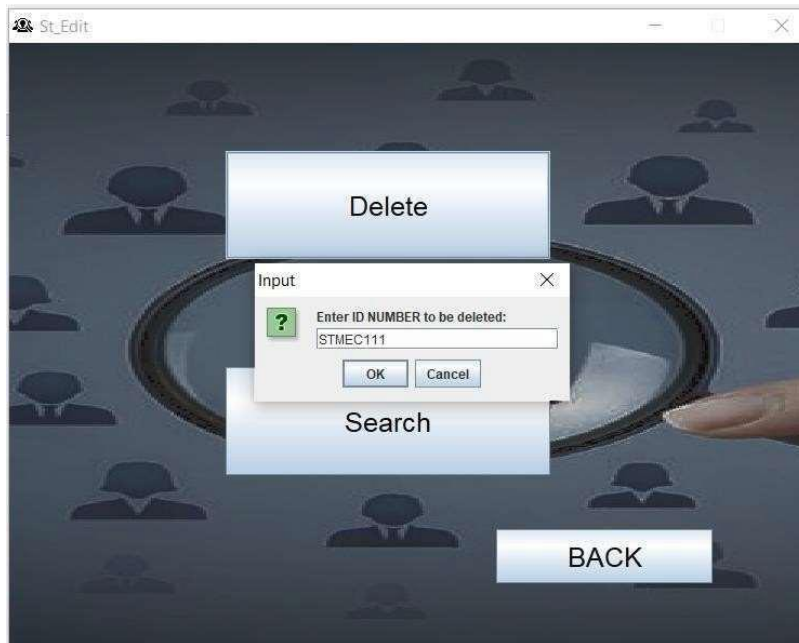


Fig no.16 Student Delete

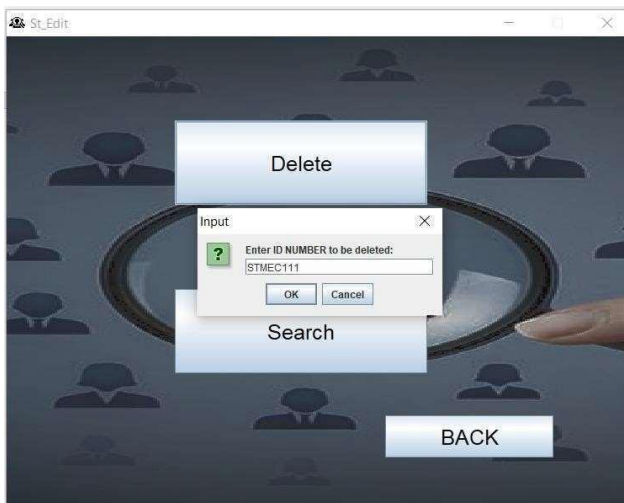


Fig no.17 Student Deleted

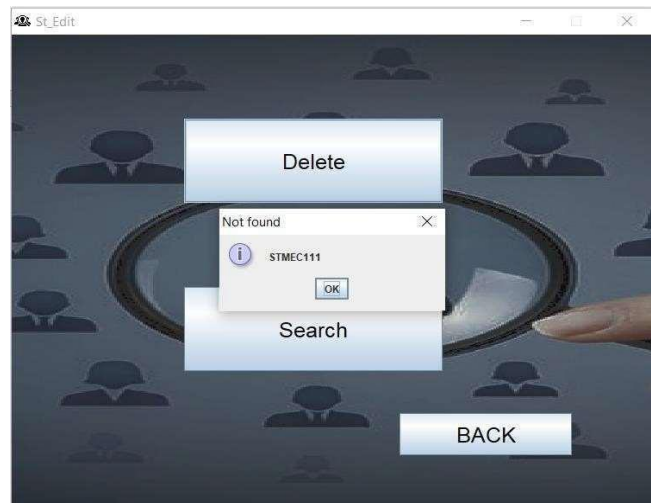


Fig no.18 Student to be deleted not found

On clicking “Search”, Search panel will be displayed. Searching is possible in different criteria.

a) Individual wise:

St_Search

☐ Individual Search ☐ CGPA WISE

Input

Enter ID NUMBER:

STEE15

OK Cancel

Backlogs

select

Branchwise

Select

BACK

If found:

St_Search

☐ Individual Search ☐ CGPA WISE

ST_ID	USN	NAME	BRANCH	EMAILID	TOTAL_C	BACKLOGS
STISE10	1NH18IS2	POORVI	ISE	p@gmail...	5.6	YES

Backlogs

select

Branchwise

Select

BACK

Fig no.20 Individual search successful

b) CGPA WISE

Displays the student with that minimum required cgpa and above that.

St_Search

Individual Search CGPA WISE

Backlogs Branchwise

select Select

BACK

Input

Enter minimum CGPA:

7

OK Cancel

Fig no.22 CGPA Search Input

St_Search

Individual Search CGPA WISE

Backlogs Branchwise

select Select

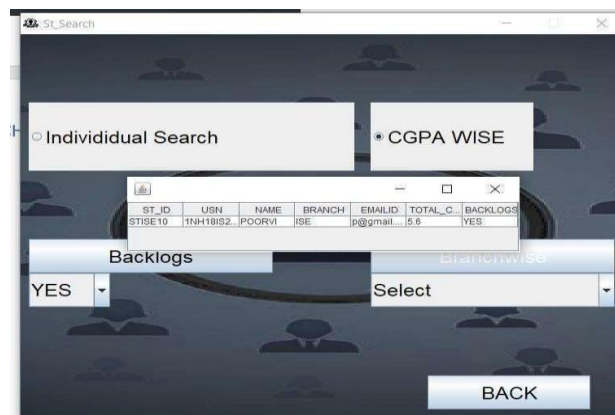
BACK

ST_ID	USN	NAME	BRANCH	EMAILID	TOTAL_C...	BACKLO...
STCIV1	1NH18CI...	KIARA	CIV	kiara@g...	8.2	NO
STCSE3	1NH18C...	NARAYA...	CSE	n@gmail...	9.87	NO
STMEC6	1NH18M...	SHYAM	MEC	j@gmail...	9.34	NO

Fig no.23 CGPA Search Output

c) BACKLOGSWISE

For example if chose "YES".

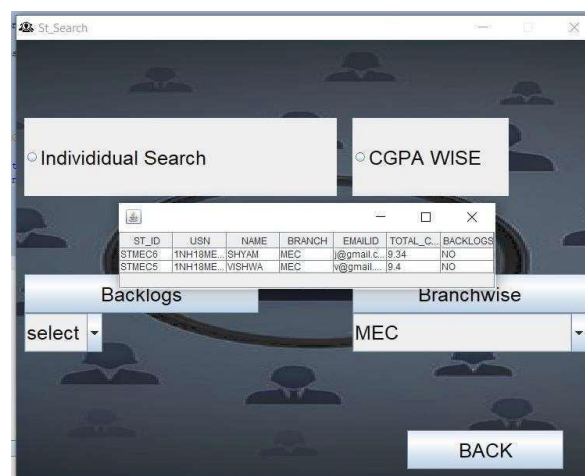


ST_ID	USN	NAME	BRANCH	EMAILID	TOTAL_C	BACKLOGS
STISE10	1NH18IS2...	POORVI	ISE	p@gmail...	5.6	YES

Fig no.24 Backlogwise Search

d) BRANCHWISE

For example ,if chose MEC



ST_ID	USN	NAME	BRANCH	EMAILID	TOTAL_C	BACKLOGS
STMEC0	1NH18ME...	SHYAM	MEC	j@gmail.c...	8.34	NO
STMEC5	1NH18ME...	WISHWA	MEC	v@gmail...	8.4	NO

Fig no.25 Branchwise Search

5.5 TRAINEE MENU

On clicking “Trainee”, Trainee Menu is displayed.



Fig no.26 Trainee Menu

On clicking “Register”, Trainee Registration form is displayed. It registers Trainees, give them an ID for further procedures and stores in Trainee sql table.

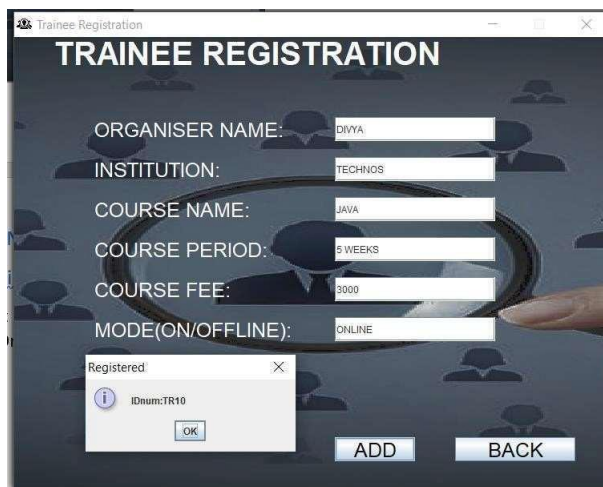
A screenshot of a web application window titled 'Trainee Registration'. The main heading is 'TRAINEE REGISTRATION'. Below the heading, there are several input fields with labels: 'ORGANISER NAME:' (with 'DIWYA' entered), 'INSTITUTION:' (with 'TECHNOS' entered), 'COURSE NAME:' (with 'JAVA' entered), 'COURSE PERIOD:' (with '5 WEEKS' entered), 'COURSE FEE:' (with '3000' entered), and 'MODE(ON/OFFLINE):' (with 'ONLINE' entered). At the bottom of the form, there are two buttons: 'ADD' and 'BACK'. A small modal window titled 'Registered' is open in the foreground, displaying an information icon, the text 'IDnum:TR10', and an 'OK' button.

Fig no.27 Trainee Registration Panel

On clicking “Delete”, an input dialog box opens asking for which Trainee entry to be deleted. If present, the selected Trainee’s record will be erased. If not present, a message will be displayed “Not Found”.

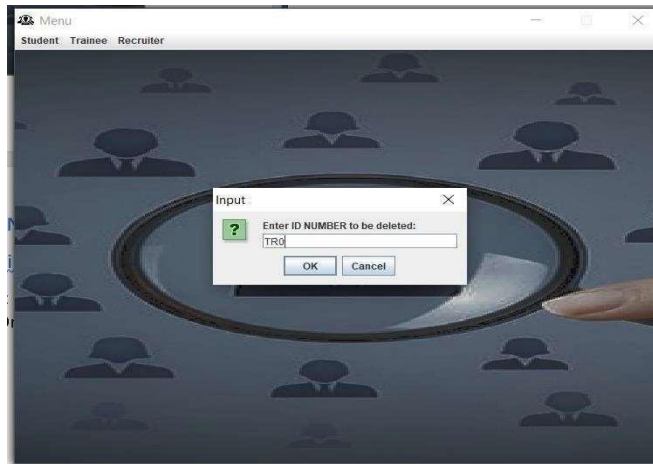


Fig no.28 Trainee Delete

If Not found:

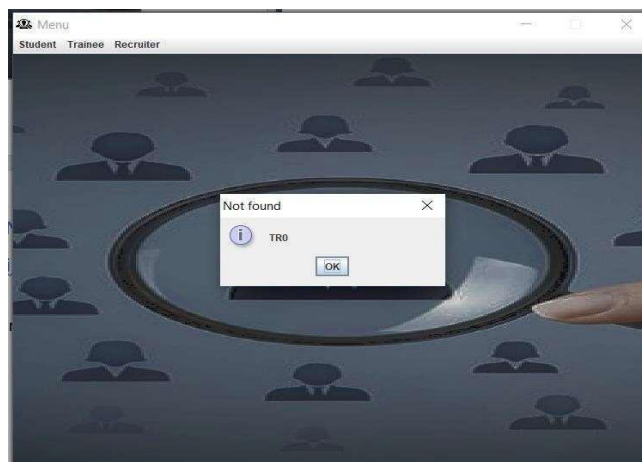


Fig no.29 Trainee Delete Unsuccessful

5.5 Recruiter Menu

On clicking “Recruiter”, Recruiter menu is displayed.



Fig no.30 Recruiter Menu

On clicking “REGISTER”, Recruiter Registration form is displayed. It registers Companies, give them an ID for further procedures and stores in Rec sql table. And the information regarding company and drive date is written in a file. This helps to display all the companies and its information in a file.



The screenshot shows a web application window titled "Recruiter Registration". The main form has a dark background with a circular pattern of silhouettes. It contains three input fields: "Name:" with the value "CICSO", "Drive Date:" with the value "24/02/2021", and "About:" with the text "OR FURTHER DEATILS REFER OFFICIAL WEBSITE.". Below the "About:" field is a scrollable text area. A small "Registered" dialog box is open, displaying "IDnum:REC9" and an "OK" button. At the bottom of the form are two buttons: "Add" and "Back".

Fig no.31 Recruiter Registration



The screenshot shows a text file named "Company - Notepad". The file contains the following text:

Company Name	Drive Date	Application Instructions
UBER	06-09-2021	APPLY ONLINE
QUORA	05-07-2020	DATA HANDLING SKILLS REQUIRED. APPLY ONLINE IN THE OFFICIAL WEBSITE.
MICROSOFT	26/08/2020	APPLY ONLINE.
CICSO	24/02/2021	WANTED COMPUTER TECHNIAN. APPLY ONLINE. FOR FURTHER DEATILS REFER OFFICIAL WEBSITE.

Fig no.32 Text file on Company Information

On clicking “Drives”, a list of companies along with drive dates are displayed.

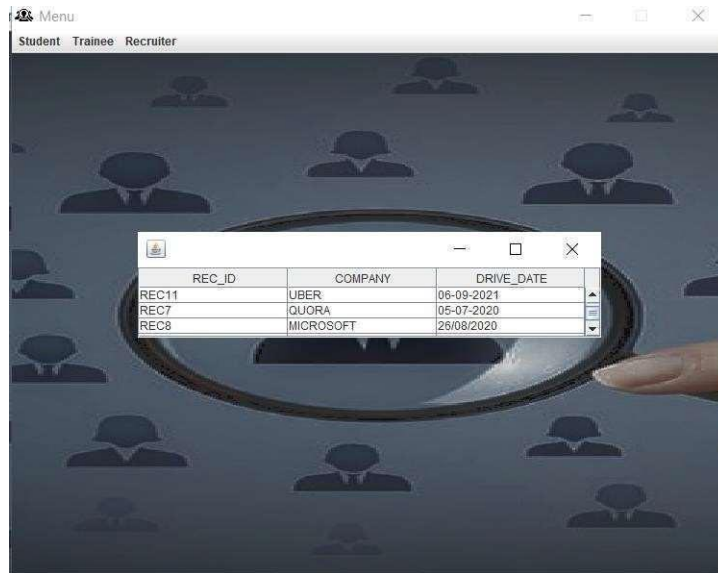


Fig no.33 Drives

On clicking “Selected”->”Insert”, a panel opens to add the student who is placed in a company. His/Her entry is deleted from Student sql table and is added in the Selected sql table. His/Her record is written onto a text file along with placement details.

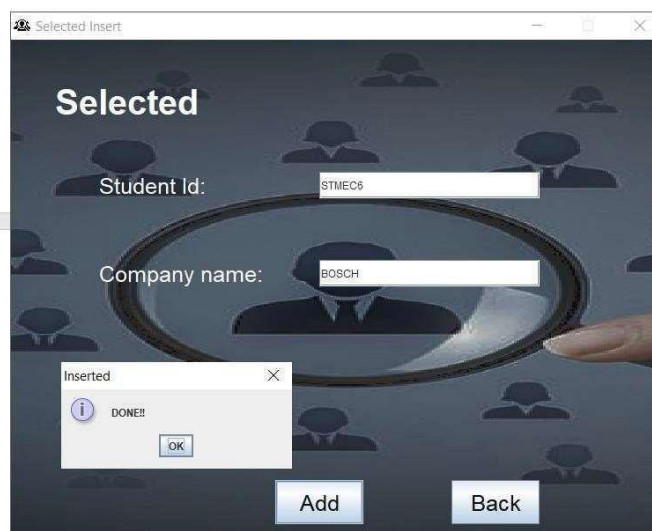


Fig no.34 Placed Student Input



Fig no.35 Text file on Placed Students

On clicking “Selected”->”View”, displays placed students with self updating date.



Fig no.36 Placed Students

CHAPTER 6

CONCLUSION

Recruitment and Training Desktop Application is a simple desktop application which helps an institution to manage data of student, trainees and recruiters .It is an application developed for only one or one department who to manage records of the students,trainees and the companies. It is used for registration of the above mentioned field members .It is used as an help desk which can filter the required data when one comes to its approach.It helps us to save the required data in the created in our databases and also in the files (better readability).The Identity Numbers helps to secure data regarding privacy.We get number of registrations at any point of time while working with application.It also helps to make separate records of registered and places students.All its functionalities makes on easier to work with placement and training data.

REFERENCES

- GeekForGeeks
- Telusko youtube channel
- JavatPoint
- Tutorial Point