

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**STUDENT ADMISSION PORTAL**

**A Mini Project Report**

**Submitted By**

**PAVAN KULKARNI**

*IN PARTIAL FULFILMENT OF THE AWARD OF*

**BACHELOR IN ENGINEERING**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**CERTIFICATE**

This is to certify that the mini project work titled

**STUDENT ADMISSION PORTAL**

**Submitted in partial fulfillment of the degree of Bachelor of Engineering**

**To**

**PAVAN KULKARNI| 1NH19CS409**

***During Even semester 2019 – 2020***

***For 19CSE48***

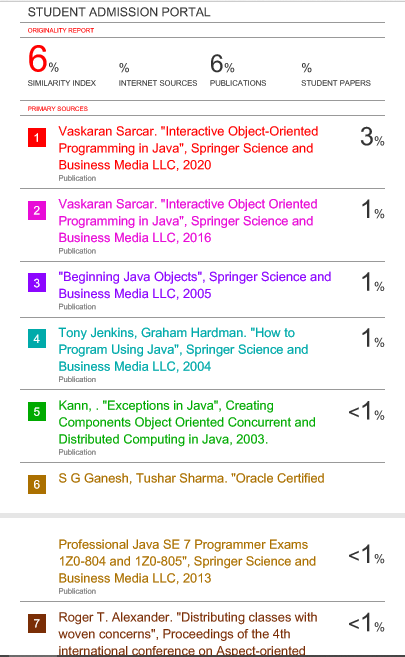
**SIGNATURE OF REVIEWER SIGNATURE OF HOD**

**SEMESTER END EXAMINATION**

**Name of examiner Signature with date**

**MS.Asha Borah**

**PLAGARISM REPORT**

****

**ABSTRACT**

Every year, students in thousands queue up for collecting admission / application

Forms and then again for submitting the admission / application forms. This leads to

Problems in managing the applications, resulting in annoyed parents and students alike.

Institute incurs expenditure on printing of admission forms, and in absence of any reliable

Forecasting system on how much application it is going to receive it may overspend by

Printing excessive application forms or may fall in crisis if there is excess demand of application forms. Chances of form misplacement are very high, because of huge number of

Applications. Wastage of institute's resources due to involvement of people / teachers in

Form collection.

The goal of Online Admission System is to computerize the admission management

System structure, its related operation, and functionality to rectify these weaknesses. Also,

The purpose is to provide support to the administration and admission seeking candidates

By providing a faster, transparent, and easy way of maintaining records and utilize them for

Reference and further proceedings.

Student online admission starts with tab 'Apply Now'. Student collects all required

Information and document, check all the check boxes, and proceed for next step. Read the

Notes given on next screen. Enter the details in the respective fields and confirm the

Details. On confirmation, it generates applicant number. The credentials will be sent on

Registered email address.

**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 Soja Rani,** Designation, 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.

**CONTENTS**

**Abstract I**

**Acknowledgement II**

**Contents III**

1. **Introduction**
   1. Problem Definition [8]
   2. Objectives [6]
   3. Methodology to be Followed [7]
   4. Expected Outcomes [9]
   5. Requirement Specifications [12]
2. **Object Oriented Programming Concepts**
   1. Classes [13]
   2. Objects [13]
   3. Inheritance [13]
   4. Polymorphism [15]
   5. Abstract Class [16]
   6. Exception Handling [17]
3. **Design**
   1. Algorithm [19]
4. **Implementation**
   1. Module 1: ADMIN LOGIN [20]
   2. Module 2: SERVER [21]
   3. Module 3: DISPLAY [22]
   4. Module 4: REGISTER [23]
   5. Module 5: STUDENT PROFILE [23]
5. **Results**
   1. Sample Output [24]
6. **Conclusion** [28]
7. **References** 
   1. Books [29]

**CHAPTER 1**

# INTRODUCTION

## A. STUDENT ADMISSION PORTAL

The goal of Online Admission System is to computerize the admission management system structure, its related operation, and functionality to rectify these weaknesses. Also, the purpose is to provide support to the administration and admission seeking candidates by providing a faster, transparent, and easy way of maintaining records and utilize them for reference and further proceedings.

Student online admission starts with tab 'Apply Now'. Student collects all required information and document, check all the check boxes, and proceed for next step. Read the notes given on next screen. Enter the details in the respective fields and confirm the details. On confirmation, it generates applicant number. The credentials will be sent on registered email address.

## B. PROBLEM STATEMENT

Every year, students in thousands queue up for collecting admission / application forms and then again for submitting the admission / application forms. This leads to problems in managing the applications, resulting in annoyed parents and students alike.

Institute incurs expenditure on printing of admission forms, and in absence of any reliable forecasting system on how much application it is going to receive it may overspend by printing excessive application forms or may fall in crisis if there is excess demand of application forms.

Chances of form misplacement are very high, because of huge number of applications. Wastage of institute's resources due to involvement of people / teachers in form collection.

## C. EXPECTED OUTCOMES

Software is computer-based and can record plenty of data. We can search, forecast, record any achieved task, etc. within no time. This system helps the student as well as the administrative department of an organization for planning and execution of a task in a methodical manner. They can plan at what time they can undertake the college list, student details display,result,update etc in the organization.

## D. REQUIREMENTS

**HARDWARE REQUIREMENTS**

* Processor: Intel core i3 or above
* RAM: 2GB or more

**SOFTWARE REQUIREMENTS**

* Operating system: Windows 7/8/10
* Language: java
* Compiler: javac, eclipse

**CHAPTER 2**

# OBJECT-ORIENTED PROGRAM

## A. CLASS

A class is like a blueprint or a template for creating objects in java. It defines the state or behavior of the object created. A class can have any number of variables and methods of various types to access to different values.

Each class has a constructor, it can be of type default or parameterized. These constructors are used to initialize objects with default values. A class can also inherit characteristics from other classes.

A class declaration can include these in order:

Access Modifier -> Class name -> Superclass -> Interface -> Body There are various types of classes such as:

* Nested class
* Anonymous class  Lambda expressions **Syntax:**

class Classname

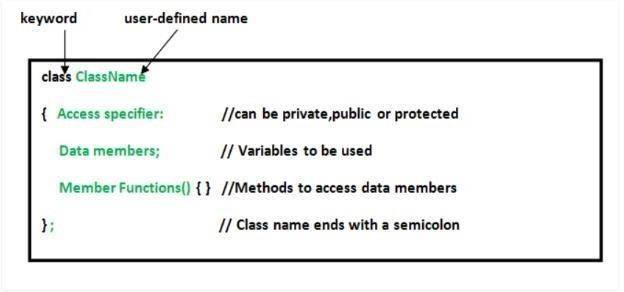


Fig 2.1 (a) Class example

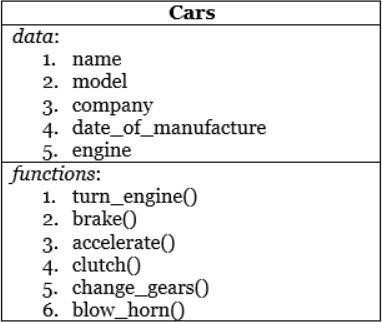


Fig 2.1 (b) Members of a class

## B. OBJECT

An object is a self-contained entity that has a state, behavior and identity.

Object allocates memory for the template class. It defines the behavior of the class. It is a specimen or instance of class used to invoke or execute any of the methods or features of the class for which object is created.

An object contains physical as well as logical entity whereas a class does not. Memory or storage allocation takes place for a class when object is created. The methods and the variables of a class are accessed using objects.

**Syntax:**

Classname objectname;

Classname reference\_variable=new Classname();

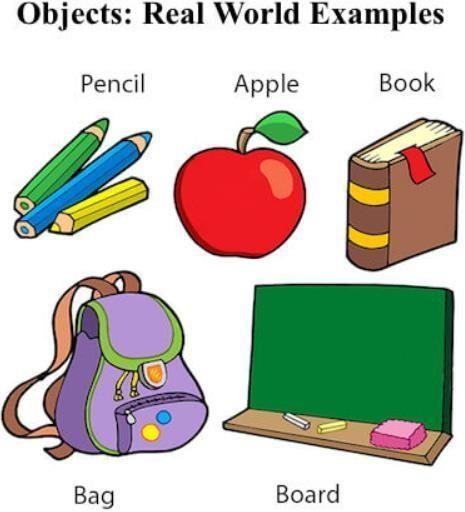


Fig 2.2 (a) Objects example 1

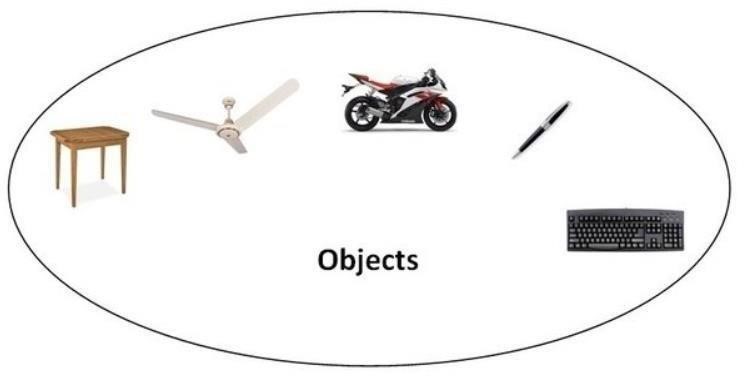


Fig 2.2 (b) Objects example 2

## C. INHERITANCE

Inheritance is an oops concept in java that allows us to define a class from an existing class. The keyword ‘extends’ is used for inheritance.

* Superclass: The parent/base class from which attributes, methods are inherited.
* Subclass: The child/derived class which inherits attributes, methods.

### 1. SINGLE INHERITANCE

Single inheritance is when a class inherits properties from one class only. All the attributes except private members are inherited or extended by child class from parent class. class A

{

}

class B extends A

{

}

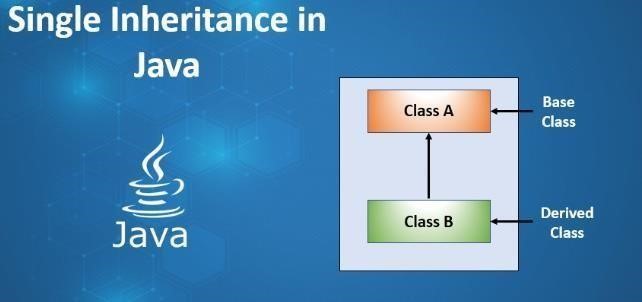


Fig 2.3 (a) Single Inheritance block diagram

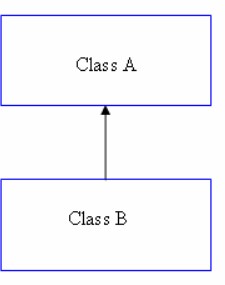


Fig 2.3 (b) Single inheritance

### 2. MULTILEVEL INHERITANCE

Multilevel inheritance is when a class inherits properties from derived class. This derived class becomes the parent for the new child class. It allows accessing of grandparent class attributes by the child class also.

class A

{

}

class B extends A

{

}

class C extends B

{

}

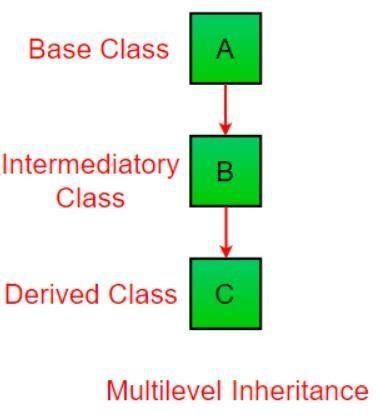


Fig 2.3 (c) Multilevel inheritance block diagram

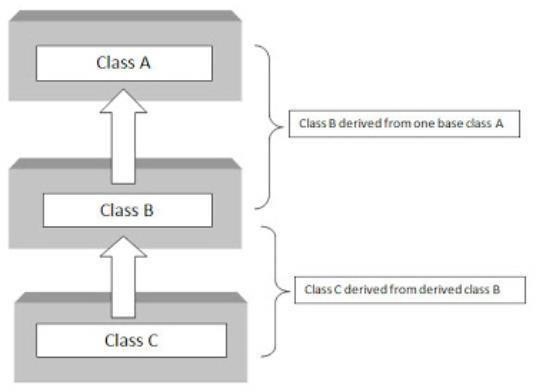


Fig 2.3 (d) Multilevel inheritance diagram and features

### 3. HIERARCHICAL INHERITANCE

Hierarchical inheritance is when a class is inherited two or more classes. In this type of inheritance all of the super class’s sub classes inherit same attributes of the parent class. class A

{

}

class B extends A

{

}

class C extends A

{

}

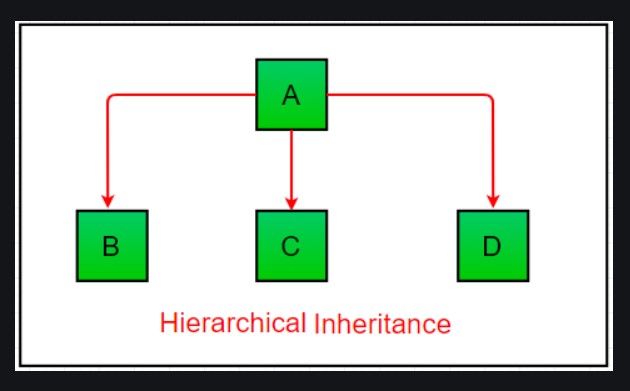


Fig 2.3 (e) Hierarchical inheritance

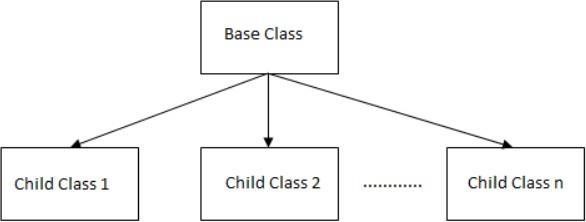


Fig 2.3 (f) Hierarchical inheritance block diagram

## D. POLYMORPHISM

Polymorphism is a very important concept in object-oriented programming. It means that the same object, method or operator acts differently in different cases. Types of polymorphism are:

* Run-time polymorphism
* Compile-time polymorphism

Run-time polymorphism is done using method overriding.

**Method Overriding**: It means that different methods have same syntax and return

type. class A{ }

class B extends A{ public void display(){

System.out.println(“Method1”);

}} class C extends A{ public void display(){

System.out.println(“Method2”); }}

Compile-time polymorphism is done through method overloading and operator overloading.

**Method Overloading**: It means that different methods with same name differ in number, type or sequence of arguments passed in them.

class A{

public void display(int a){

System.out.println(a);

}} class B{

public void display(String

s){ System.out.println(s);

}} class C{

public void display(int x, int y){

System.out.println(x+” “+y); }}

**Operator overloading**: ‘+’ operator is used for concatenation as well as addition operator.

System.out.println(m+”ways”);

System.out.println(a+b);

**Final methods**: final methods are those methods which cannot be modified by any other class and hence data is protected. The keyword “final” is used as a prefix to the name of the method, example, final visual(){ }

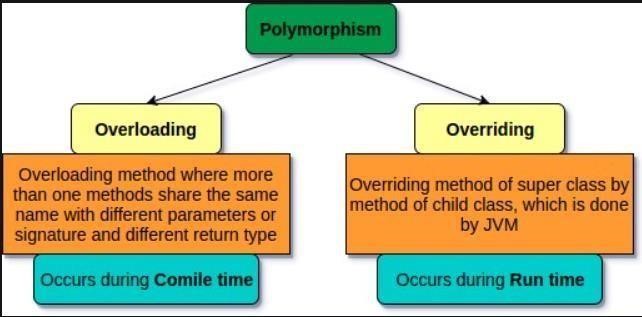
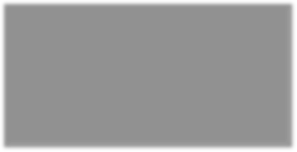


Fig 2.4 (a) Polymorphism types explanation

## E. I/O FUNCTIONS IN JAVA

I/O functions in java are used to process the input and give output. It uses concept of Streams to operations fast. We can also perform file handling in java using these streams.

* System.out
* System.in
* System.err
* OutputStream  InputStrea m Example:

Scanner s=new

Scanner(System.in);

System.out.println(“Hello”);

System.err(0);

## F. PACKAGES IN JAVA

Packages in java is to encapsulate a group of classes, sub-packages and interfaces. It prevents naming conflicts and also provides controlled access and are known as data encapsulation.

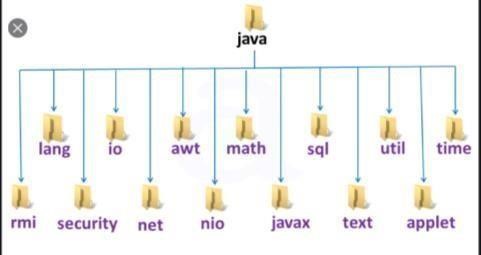


Fig 2.6 (a) Packages example

## G. EXCEPTION HANDLING

An exception is the error that occurs during the successful execution of a program. These errors need to be handled and this process is known as exception handling. There are two types of exceptions: checked and unchecked.

Checked exception: These are the exceptions that arrive at the time of compilation, checked by compiler and needs to be handled by the user.

Unchecked exception: These are the errors that occur during execution process and are ignored by compiler because they occur due to logical errors, programming bugs or improper uses of APIs. These errors are known as runtime exceptions.

All the exceptions in java are a subclass of java.lang.Exception class.

Different parts in Exception handling are try, catch, throw, throws, throwable and finally. Every try block should have a catch or multiple catch statements. Finally is used to execute any statement that needs to be executed after catching exception and throw, throws and throwable are for throwing the error occurred.

Example:

try{

//protected code

}

catch(ExcepitonType e){

//Exception caught

}

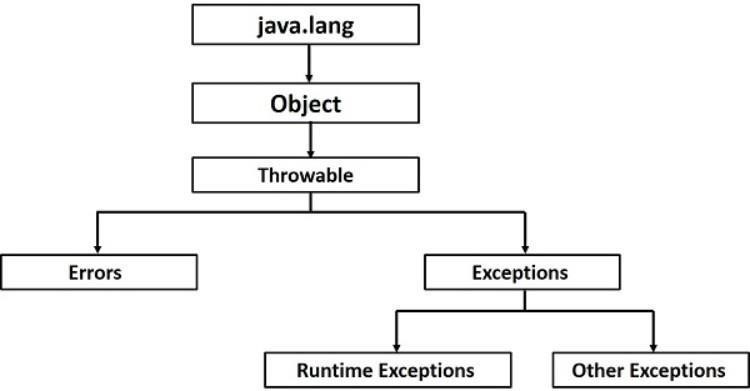


Fig 2.7 (a) Exception handling hierarchy

**CHAPTER 3**

# DESIGN

## A. ALGORITHM

Step 1: START

Step 2: Admin portal

Step 3: If credentials are valid: -

1.1 Search

1.2 Update

1.3 display student details

Step 4:STUDENT PORTAL

4.1 Register

4.1.1 Enter Personal details

4.2 If Login.

4.2.1 Display Personal Details

4.2.2 Pay Fees

4.2.3 Check admission Status

4.2.4 Update Personal Details

STEP 5: STOP

**CHAPTER 4**

# IMPLEMENTATION

## PARTIAL CODE

### A. ADMIN LOGIN

public void run()

{

try

{

admin=new AdminInfo();

AdminLogin frame = new AdminLogin(); frame.setVisible(true);

}catch (Exception e)

{

e.printStackTrace();

}

}

}

)

;

}

### B. MODULE 2

btnPutChoice.addActionListener(new ActionListener())

{ public void actionPerformed(ActionEvent e2) {

try { output.reset();

}

catch (SocketException e1)

{

JOptionPane.showMessageDialog(null,"Server Down");

return;

}

catch (IOException e) {

e.printStackTrace();

}

if(textPane.getText().isEmpty())

{

|  |  |  |
| --- | --- | --- |
| blPleaseEnterThe.setVisible(true); Timer t=new Timer(2000,new)  ActionListener()  actionPerformed(ActionEvent e) |  | {  { |
| lblPleaseEnterThe.setVisible(false); |  | }  });  t.start(); return; |

}

text1=textpane.getText();

p1.Setvisivle(true); textPane.setText(null); textPane.SetVisible(false);

btnPutChoice.setvisible(false);

iblCollegeNae.setVisible(false);

} }}; add (btnPutchoice); if(!flag) { p1.setVisible(true); textPane.setVisible(false); btnPutChoice.setVisible(false);

}}

### A. MODULE 3

public void student\_list()

{

int rows = model.getRowCount(); for(int i = rows - 1; i >=0; i--)

{

model.removeRow(i);

}

Iterator<StudentInfo> i=ServerCollection.students.iterator(); while(i.hasNext())

{

student=i.next();

model.addRow(new

Object[]{student.ID,student.name,student.address,student.mob,student.do b,student.marks,student.result});

}

}

### B. MODULE 4

public void run()

{

try

{

Student=new StudentInfo();

StudentLogin frame = new StudentLogin(); frame.setVisible(true);

}catch (Exception e)

{

e.printStackTrace();

}

}

});

}

/\*\*

### C. MODULE 5

public void run()

{

try

{

studentregistration=new StudentregistrationInfo();

StudentregistrationLogin frame = new StudentregistrationLogin(); frame.setVisible(true);

}catch (Exception e)

{

e.printStackTrace();

}

}

});

}

.

**D.** **MODULE 6**

public void run()

{

try

{

checkresult=new CheckResultInfo(); Checkresult frame = new Checkresult(); frame.setVisible(true);

}catch (Exception e)

{

e.printStackTrace();

}

}

});

}

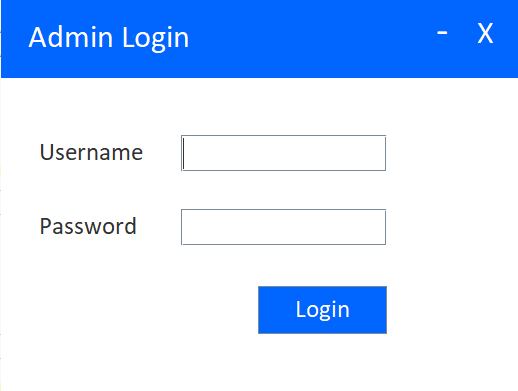
/\*\*

\* Create the frame.

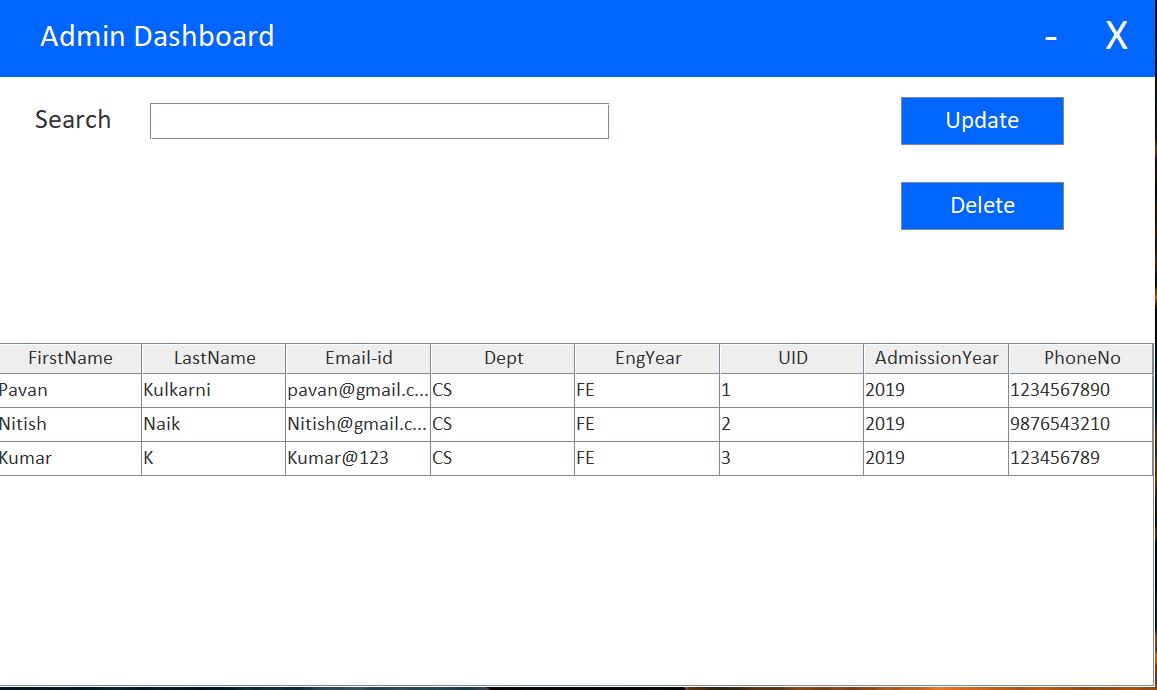
**CHAPTER 5**

# RESULT

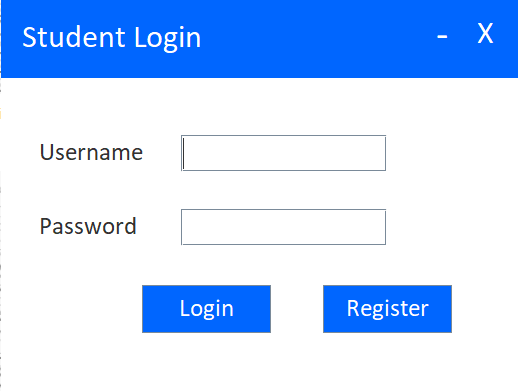
**ADMIN LOGIN SYSTEM**



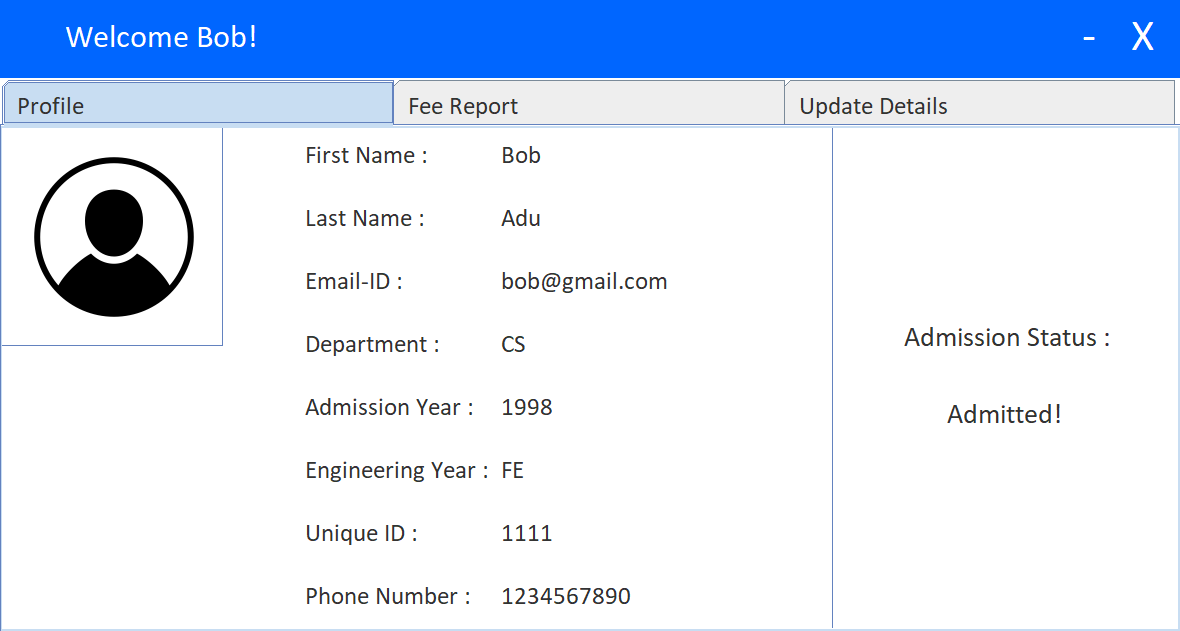
**ADMIN** **DASHBOARD**



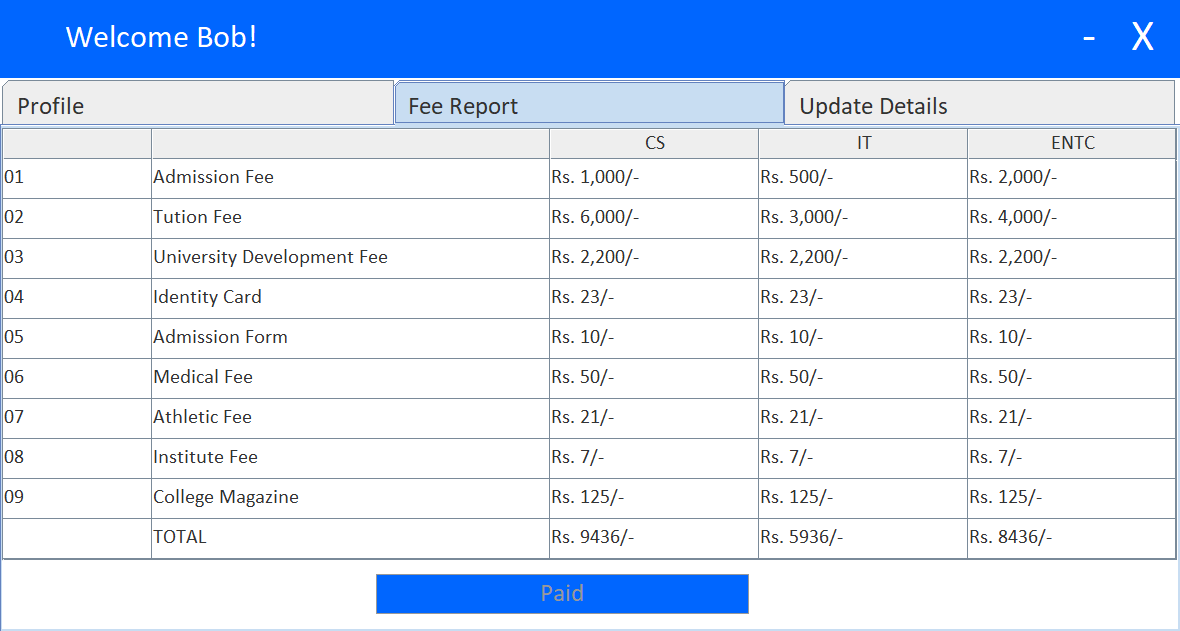
**STUDENT LOGIN**



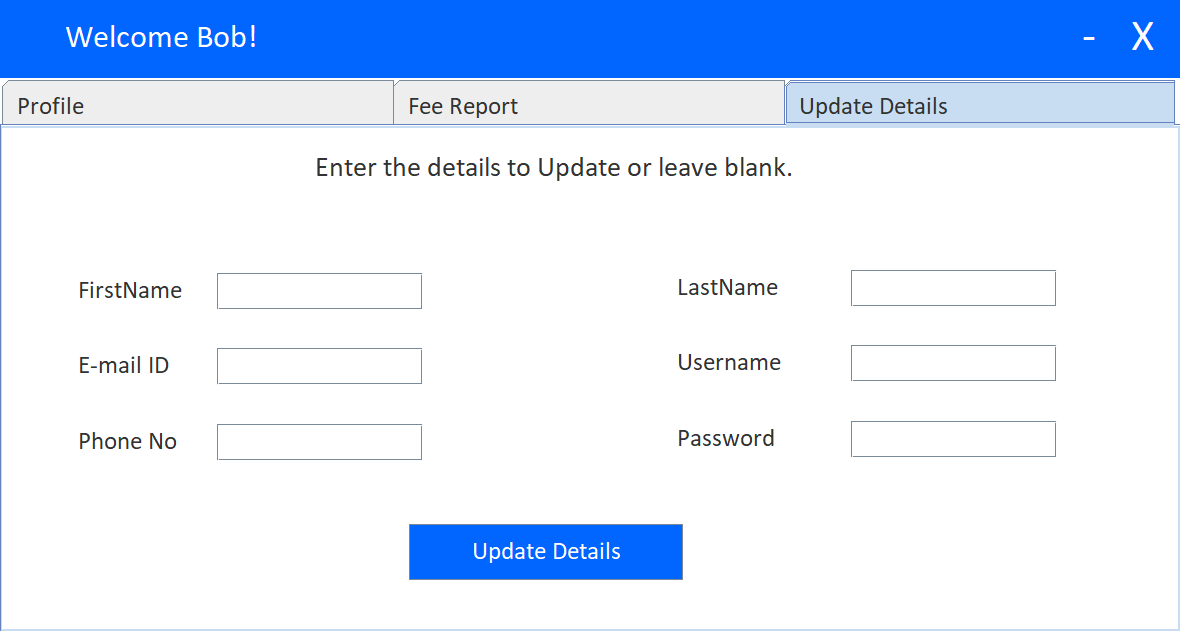
**STUDENT PROFILE**



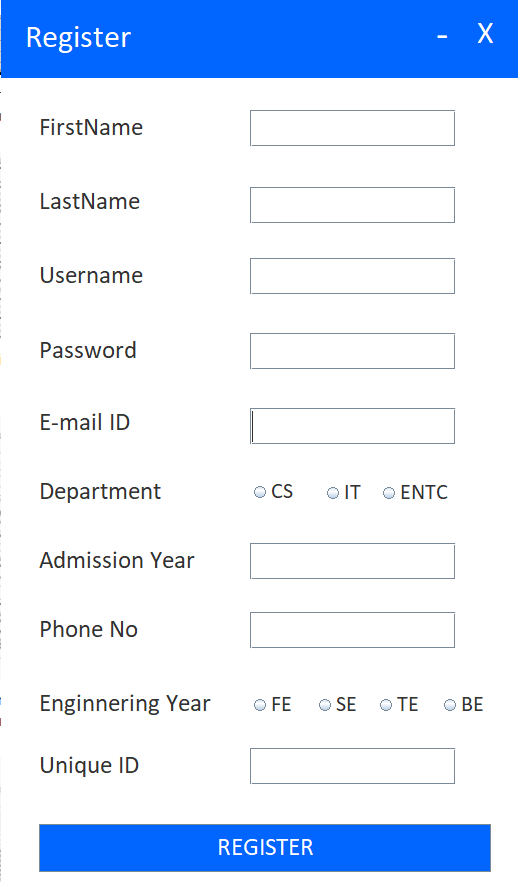
**FEE REPORT**



**STUDENT UPDATE**



**STUDENT REGESTRATION**



**CHAPTER 6**

# CONCLUSION

This mini-project will serve the needful purposes required to maintain records in Student admission portal. This helps the officials to maintain details of various Students thus, making it easier to keep a record of all the needful services to be done on them timely. These data can be read and updated anytime. It records the college list,cut off college,display student list,display college list etc.

## CHAPTER 7

**REFERENCE**

**Java: The Complete Reference**, SE 11, Eleventh Edition, Herbert Schildt (December 2018), McGraw-Hill.

This book explains all the concepts of java in detail, like, data types, variables, arrays, operators, classes, objects, methods, inheritance, interfaces, packages, swing, exception handling, multithreading, I/O classes, networking, servlets, etc. and is a very helpful guide to the students.

1. [**www.geeksforgeeks.org**](http://www.geeksforgeeks.org/)
2. [**www.javatpoint.com**](http://www.javatpoint.com/)
3. [**https://stackoverflow.com**](https://stackoverflow.com/)
4. [**https://codeforwin.org**](https://codeforwin.org/)