

**IBM CARRER EDUCATION**

**MAIN PROJECT**

**DOMAIN NAME: JAVA**

**COLLEGE MANAGEMENT SYSTEM**

**Submitted By,**

**YESHA SHAH(D2D 1) , ZARANA NAKRANI (18082271003)**

**II Year – CBA (A Section)**

**GANPAT UNIVERSITY, AHMEDABAD.**

**Submitted To,**

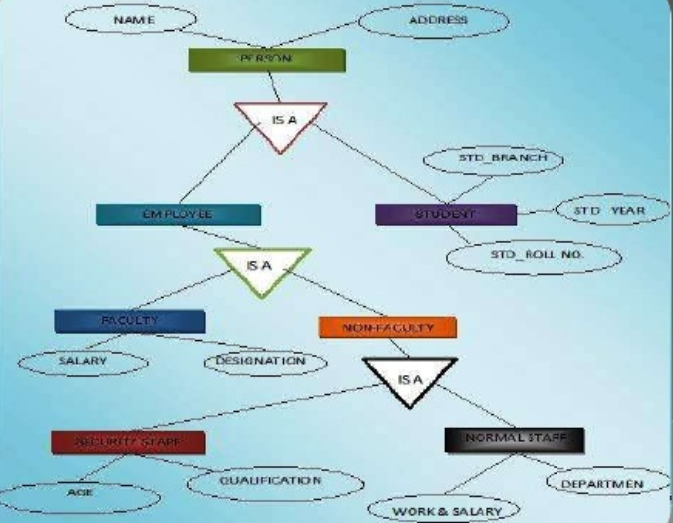
A. Saai Sanjeev Achaarya

IBM Software Technical Trainer



**COLLEGE MANAGEMENT SYSTEM**

**FLOWCHART**



**COLLEGE MANAGEMENT SYSTEM**

**SOFTWARE SPECIFICATIONS**

* JDK 1.5 or later
* Our software is platform independent since it uses java.
* Latest and updated antivirus
* Anti-hacking software
* Any RDBMS(Relational Database Management System)

**HARDWARE SPECIFICATIONS**

* 256 MB RAM
* Network Connections with 100 Mbps transfer rate.
* Server to control the database to be maintained by software.
* 1 Mb cache memory.
* 10 Gb free disk space.

**COLLEGE MANAGEMENT SYSTEM**

**DESCRIPTION**

**INTRODUCTION:**

This is a web-related application that permits us to approach the entire knowledge regarding the college, employees, students, faculties etc. This application is also called as institute management system. It offers an actual trip of the college campus. Here we would gain the recent knowledge regarding the students and employees.

**DESCRIPTION:**

This general application planned for aiding the students of an organization about details on the courses, subjects, classes, assignments, grades and time-table. It also allows the faculty to know his time-table, upload assignments and issue circulars to the pupil.

The administrator would maintain the accounts of the pupil and staff, prepares the time-table and upload the current information regarding the campus.

To reduce the headache of maintaining the record of students and teachers related documents.

To reduce the cumbersome job of maintaining several documents like : It will eliminate the delays in the generation of results and free updating of the students, this system will help in maintaining the records of absent students.

Searching will become more efficient and fast in comparison of manual searching.

It will also provide assurance that each employee of the college marked their attendance timely.

Overall it will reduce the cost and time of the college head in taking care of the college.

**ADVANTAGES:**

* Total control on the application by Administrator like assigning rights n so on.
* All the work pressure will be reduced by the use of this application.
* Availability & Accessibility of records 24X7 to authorized persons.
* Centralized data repository for trouble-free data access.
* Authenticated profile dependent access to data.
* Carefully designed & user friendly interface.
* High level data security.
* Freedom from maintaining lots of registers year by year.
* Elimination of people-dependent processes.
* Minimal data redundancy.

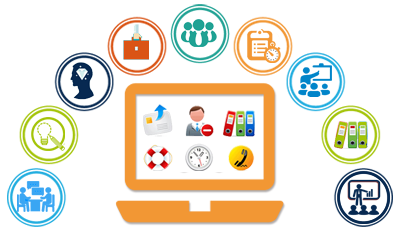
**DISVANTAGES:**

* This process is so much time consuming
* There is a threat for the record of the student and teachers, in this case, there might be a chance that person makes entry on someone else record.
* There is no proper way of getting to know about the events and extra curriculum activities happening in colleges.
* It might be the case that student tries to bribe the teachers to avoid the long queue.

**COLLEGE MANAGEMENT SYSTEM**

**PICTURES**







**COLLEGE MANAGEMENT SYSTEM**

**AIM:**

**To create a computerised library management system which:-**

* **Adds a Staff Details**
* **Adds a Course Details**
* **Adds a Time Table Details**
* **Adds a Hostel Details**

**PROGRAM:**

* **LOGIN PAGE CODE**

<html>

<head>

<meta name="viewport" content="width=device-width, initial-scale=1">

<style>

body {font-family: Arial, Helvetica, sans-serif;}

form {border: 3px solid #f1f1f1;}

input[type=text], input[type=password] {

width: 100%;

padding: 12px 20px;

margin: 8px 0;

display: inline-block;

border: 1px solid #ccc;

box-sizing: border-box;

}

button {

background-color: #4CAF50;

color: white;

padding: 14px 20px;

margin: 8px 0;

border: none;

cursor: pointer;

width: 30%;

}

button:hover {

opacity: 0.8;

}

.cancelbtn {

width: auto;

padding: 10px 18px;

background-color: #f44336;

}

.imgcontainer {

text-align: center;

margin: 24px 0 12px 0;

}

img.avatar {

width: 20%;

border-radius: 50%;

}

.container {

padding: 16px;

}

span.psw {

float: right;

padding-top: 16px;

}

/\* Change styles for span and cancel button on extra small screens \*/

@media screen and (max-width: 300px) {

span.psw {

display: block;

float: none;

}

.cancelbtn {

width: 100%;

}

}

</style>

</head>

<body>

<h2>Login Form</h2>

<form action="/action\_page.php" method="post">

<div class="imgcontainer">

<img src="avatar.png" alt="Avatar" class="avatar">

</div>

<div class="container">

<label for="uname"><b>Username</b></label>

<input type="text" placeholder="Enter Username" name="uname" required>

<label for="psw"><b>Password</b></label>

<input type="password" placeholder="Enter Password" name="psw" required>

<button type="submit">Login</button>

<label>

<input type="checkbox" checked="checked" name="remember"> Remember me

</label>

</div>

<div class="container" style="background-color:#f1f1f1">

<button type="button" class="cancelbtn">Cancel</button>

<span class="psw">Forgot <a href="#">password?</a></span>

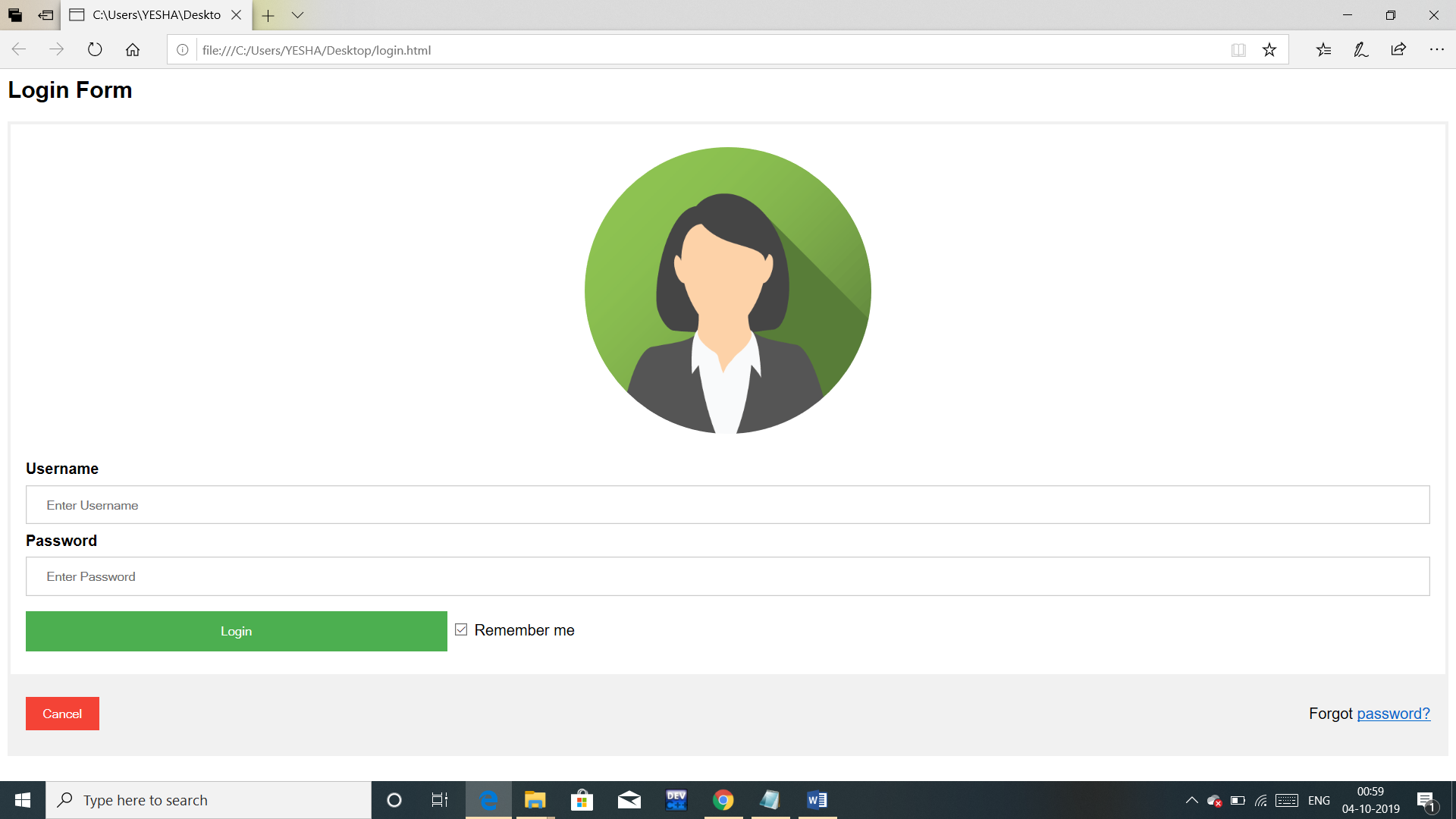
</div>

</form>

</body>

</html>

**OUTPUT**

****

* **STAFF DETAILS CODE**

class Employees

{

int id;

String name;

String state;

int contact\_no;

String qualification;

float salary;

int bonus;

Employees(int i,String n,String s,int c,String q,float sa,int b)

{

id=i;

name=n;

state=s;

contact\_no=c;

qualification=q;

salary=sa;

bonus=b;

}

void display()

{

System.out.println(id+" "+name+" "+state+" "+contact\_no+" "+qualification+" "+salary+" "+bonus);

}

}

class Employee

{

public static void main(String args[])

{

Employees e1=new Employees(1,"SpiderMan","Alaska",123456,"MBA",25000.98f,10000);

Employees e2=new Employees(2,"BenTen","California",216464,"MBA",30000f,9000);

Employees e3=new Employees(3,"BobBuilder","Colorado",648999,"BA",15000.56f,8000);

Employees e4=new Employees(4,"Iggle","Delaware",456974,"BBA",50000f,7000);

Employees e5=new Employees(5,"BumbleBee","Florida",656487,"MA",30000.34f,6000);

Employees e6=new Employees(6,"Mr.Maker","Alberta",456712,"BBA",70000.90f,5000);

Employees e7=new Employees(7,"ElmoPeter","Manitoba",123475,"MTECH",14000f,4000);

Employees e8=new Employees(8,"Yesha","Ontorio",457123,"PHD",100000f,15000);

Employees e9=new Employees(9,"Zoonie","Quebec",789452,"BTECH",25000f,8000);

Employees e0=new Employees(0,"BatMan","Nova Scotia",948756,"MA",17000.45f,3000);

e1.display();

e2.display();

e3.display();

e4.display();

e5.display();

e6.display();

e7.display();

e8.display();

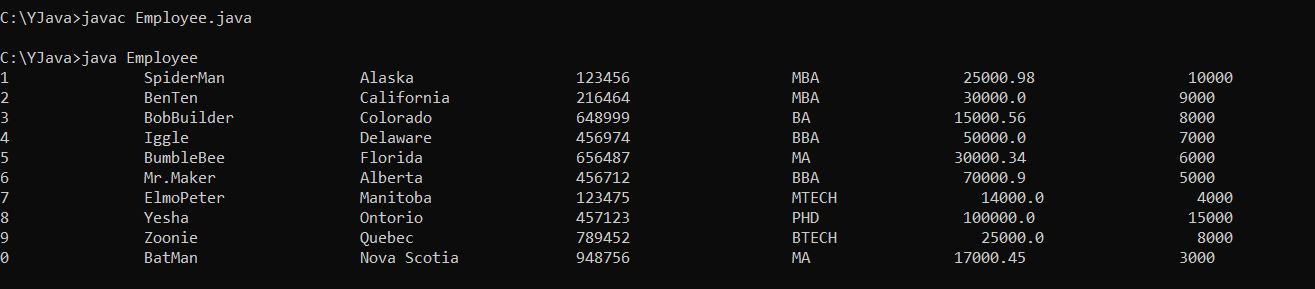
e9.display();

e0.display();

}

}

**OUTPUT**

****

* **COURSE DETAILS CODE**

import java.math.RoundingMode;

import java.util.LinkedList;

import java.util.List;

import java.util.Random;

import java.util.Scanner;

import java.math.BigDecimal;

public class Student

{

private String firstName;

private String lastName;

private String id;

private List<String> courses;

private BigDecimal tuition;

private Scanner keyboard = new Scanner(System.in);

private Student(String fName, String lastName)

{

this.firstName = fName;

this.lastName = lastName;

}

private Student()

{

}

private BigDecimal getTuition()

{

return tuition;

}

private void setTuition(BigDecimal money)

{

this.tuition = money;

}

private String getName()

{

return firstName + " " + lastName;

}

private void setFirstName(String firstName)

{

this.firstName = firstName;

}

private void setLastName(String lastName)

{

this.lastName = lastName;

}

private String getId()

{

return id;

}

private void setId(String id)

{

this.id = id;

}

private List<String> getCourses()

{

return courses;

}

private void setCourses(List<String> courses)

{

this.courses = courses;

}

private void makeID()

{

String grade;

boolean checked = false;

while (!checked)

{

System.out.println("Enter your school year 1. Freshman, 2. Sophomore, 3.Junior and 4. Senior ");

grade = keyboard.nextLine();

if (grade.length() == 1 && Integer.parseInt(grade) > 0 && Integer.parseInt(grade) < 5)

{

setId(grade.concat(randomString()));

checked = true;

}

else

{

System.out.println("The input you enter is incorrect please try again");

}

}

}

private String randomString()

{

String AB = "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz";

Random random = new Random();

int great = AB.length();

int temp;

String codeword = "";

for (int i = 0; i < 4; i++)

{

temp = (int) (random.nextFloat() \* great);

codeword = codeword.concat(Character.toString(AB.charAt(temp)));

}

return codeword;

}

private void payForCourses()

{

String answer;

BigDecimal payment;

BigDecimal moneyLeftOver;

while (getTuition().compareTo(BigDecimal.ZERO) > 0)

{

System.out.println("Your current balance is $" + getTuition());

System.out.println("Do you want pay off you balance right now");

answer = keyboard.nextLine();

if (answer.toLowerCase().equals("yes"))

{

System.out.println("How much would you like to pay right now");

if (keyboard.hasNextBigDecimal())

{

payment = keyboard.nextBigDecimal();

payment = payment.setScale(2, RoundingMode.HALF\_UP);

keyboard.nextLine();

if ((payment.compareTo(BigDecimal.ZERO) > 0) && payment.compareTo(getTuition()) <= 0)

{

moneyLeftOver = getTuition().subtract(payment);

setTuition(moneyLeftOver);

}

else if (payment.compareTo(getTuition()) > 0)

{

System.out.println("The value you have given is greater than your tuition");

}

else if (payment.compareTo(BigDecimal.ZERO) < 0)

{

System.out.println("You gave an negative number as a payment value. Please enter a positive value next time");

}

}

else

{

keyboard.nextLine();

System.out.println("You entered the wrong input so please input a number next time.");

}

}

else if (answer.toLowerCase().equals("no"))

{

break;

}

else

{

System.out.println("You gave the wrong input either enter yes or no");

}

}

}

private void chooseCourses(List<String> classes, int courseNumber)

{

switch (courseNumber)

{

case 1:

if (checkDups(classes, "History 101"))

classes.add("History 101");

break;

case 2:

if (checkDups(classes, "Mathematics 101"))

classes.add("Mathematics 101");

break;

case 3:

if (checkDups(classes, "English 101"))

classes.add("English 101");

break;

case 4:

if (checkDups(classes, "Chemistry 101"))

classes.add("Chemistry 101");

break;

case 5:

if (checkDups(classes, "Computer Science 101"))

classes.add("Computer Science 101");

break;

default:

System.out.println("You gave the wrong input");

break;

}

}

private void addCourses()

{

List<String> classes = new LinkedList<>();

setCourses(classes);

String answer;

int nextCourse;

BigDecimal size;

BigDecimal cost;

System.out.println("Do you want to add any courses? yes or no");

answer = keyboard.nextLine();

while (!answer.toLowerCase().equals("no"))

{

if (answer.toLowerCase().equals("yes"))

{

System.out.println("Which classes would you like to add now? Please choose from the following selection. " +

"Choose the number for the courses");

System.out.println("1. History 101");

System.out.println("2. Mathematics 101");

System.out.println("3. English 101");

System.out.println("4. Chemistry 101");

System.out.println("5. Computer Science 101");

if (keyboard.hasNextInt())

{

nextCourse = keyboard.nextInt();

keyboard.nextLine();

chooseCourses(classes, nextCourse);

} else {

System.out.println("You put in the wrong input: Enter a number 1 - 5 for each class");

keyboard.nextLine();

}

} else {

System.out.println("You put in the wrong input: Enter either yes or no next time");

}

System.out.println("Do you want to add any more courses?");

answer = keyboard.nextLine();

}

size = new BigDecimal(classes.size());

cost = new BigDecimal(600);

cost = cost.multiply(size);

setTuition(cost);

}

private boolean checkDups(List<String> list, String word)

{

for (String temp : list)

{

if (word.equals(temp))

{

System.out.println("You are already enrolled in that course");

return false;

}

}

return true;

}

private void displayInfo(Student[] studentList)

{

for (Student student : studentList)

{

System.out.println("Student Name: " + getName());

System.out.println("Student ID: " + student.getId());

if (student.getCourses().size() > 0) {

System.out.println("Student's Current Courses:" + student.getCourses());

} else {

System.out.println("Student's Current Courses: The student isn't enrolled in any courses");

}

System.out.println("Student's Current Balance: $" + student.getTuition());

System.out.println("------------------------------------------------------");

}

}

public static void main(String[] args)

{

try {

int size;

Scanner keyboard = new Scanner(System.in);

System.out.println("Please enter the number of students you wish to add to the system");

size = keyboard.nextInt();

keyboard.nextLine();

Student[] students = new Student[size];

Student student;

String firstName = "";

String lastName = "";

for (int i = 0; i < size; i++)

{

student = new Student(firstName, lastName);

students[i] = student;

System.out.println("Please enter your first name for Student ");

firstName = keyboard.nextLine();

student.setFirstName(firstName);

System.out.println("Please enter your last name");

lastName = keyboard.nextLine();

student.setLastName(lastName);

student.makeID();

student.addCourses();

student.payForCourses();

if (i == size - 1)

student.displayInfo(students);

}

} catch (NegativeArraySizeException e)

{

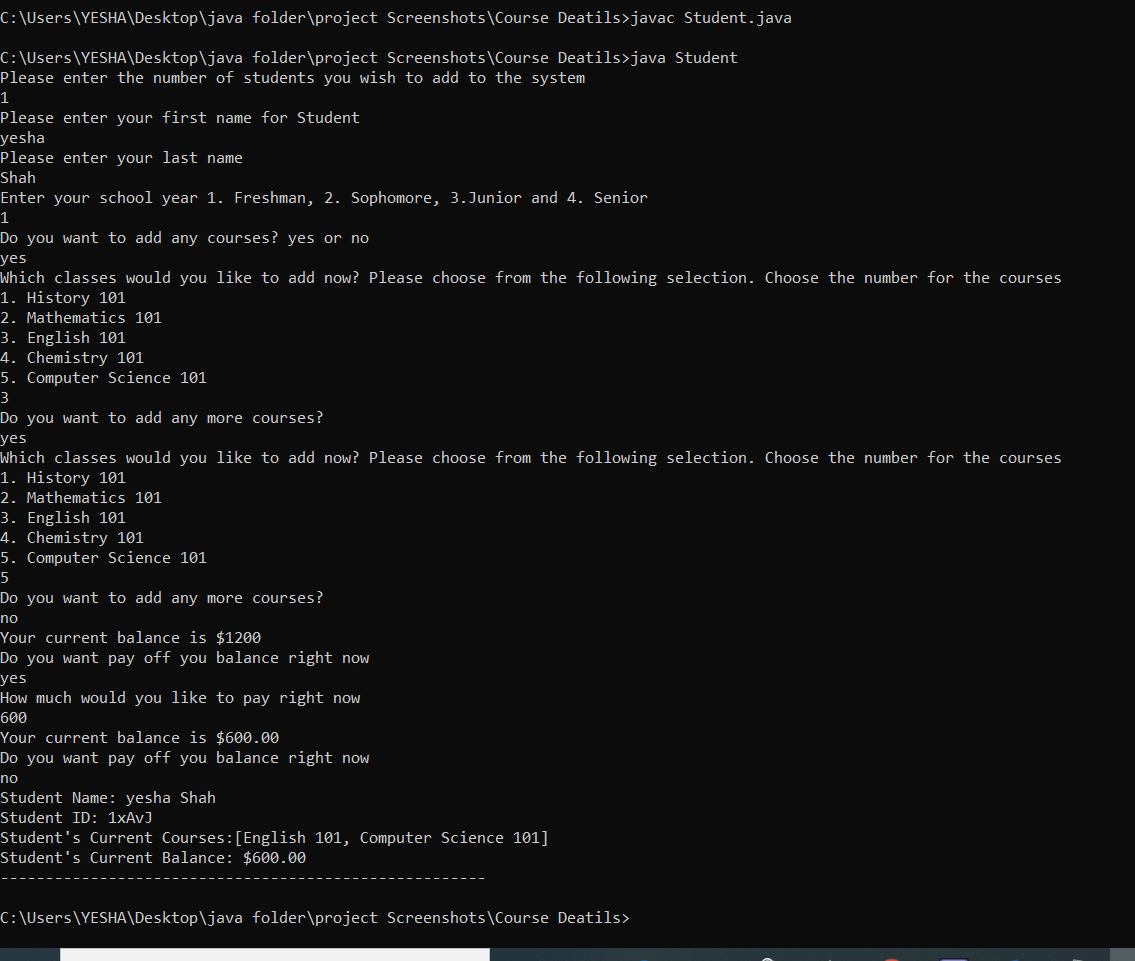
System.out.println("You can't use a negative number for size");

}

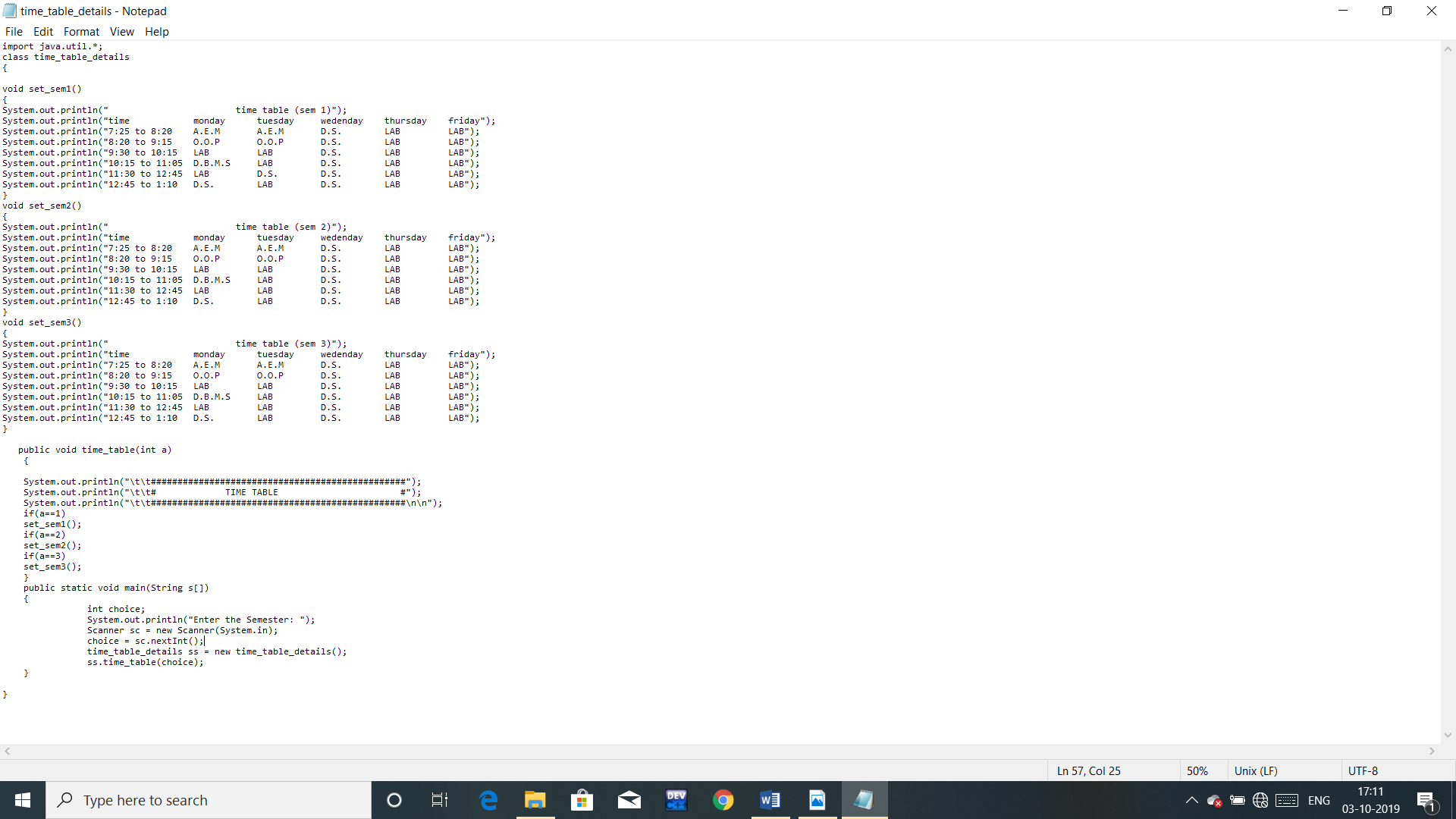
}

}

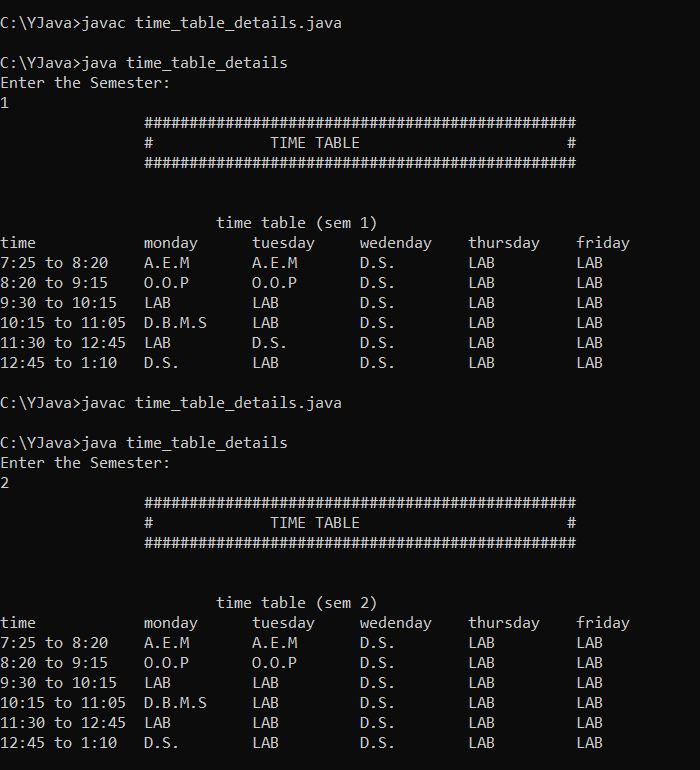
**OUTPUT**

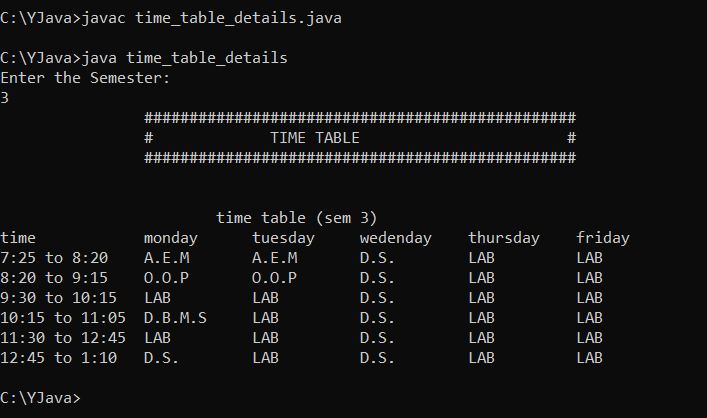
****

* **TIME-TABLE DETAILS CODE**



**OUTPUT**

****

****

* **HOSTEL DETAILS CODE**

import java.util.\*;

class hosteldetails

{

Scanner sc=new Scanner(System.in);

int city;

int a;

int b;

void heading()

{

System.out.println("\t\t\t\t\t\t\t\*\*\*\*\*\*\*\*\*\*Hostel Details\*\*\*\*\*\*\*\*\*\*");

System.out.println("\t\t\t\t\t\t\t\* \*");

System.out.println("\t\t\t\t\t\t\t\* \*");

System.out.println("\t\t\t\t\t\t\t\* \*");

System.out.println("\t\t\t\t\t\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

}

void info()

{

System.out.println("Hostel facility is available in..");

System.out.println("1) Ranip");

System.out.println("2) Gota");

System.out.println("At which place you are willing to avail the facitlity of Hostel");

city=sc.nextInt();

take();

}

void take()

{

if(city==1)

{

ranip();

}

else if(city==2)

{

gota();

}

else

{

System.out.println("Please choose the correct option");

info();

}

}

void ranip()

{

System.out.println("You have selected Ranip");

System.out.println("Do you want AC or Non AC");

System.out.println("For AC press 1");

System.out.println("For Non AC press 2");

a=sc.nextInt();

showrout();

}

void showrout()

{

if(a==1)

{

System.out.println("Rent=5000");

System.out.println("Maximum Sharing=4");

System.out.println("Food Facility=Available");

System.out.println("Laundry=extra charge(monthly=Rs.350)");

}

else if(a==2)

{

System.out.println("Rent=4000");

System.out.println("Maximum Sharing=4");

System.out.println("Food Facility=Available");

System.out.println("Laundry=extra charge(monthly=Rs.350)");

}

else

{

System.out.println("Please choose correct option");

ranip();

}

}

void gota()

{

System.out.println("You have selected Gota");

System.out.println("Do you want AC or Non AC");

System.out.println("For AC press 1");

System.out.println("For Non AC press 2");

b=sc.nextInt();

showgout();

}

void showgout()

{

if(b==1)

{

System.out.println("Rent=5500");

System.out.println("Maximum Sharing=4");

System.out.println("Food Facility=Available");

System.out.println("Laundry=extra charge(monthly=Rs.350)");

}

else if(b==2)

{

System.out.println("Rent=4500");

System.out.println("Maximum Sharing=4");

System.out.println("Food Facility=Available");

System.out.println("Laundry=extra charge(monthly=Rs.350)");

}

else

{

System.out.println("Please choose correct option");

gota();

}

}

public static void main(String args[])

{

hosteldetails hd=new hosteldetails();

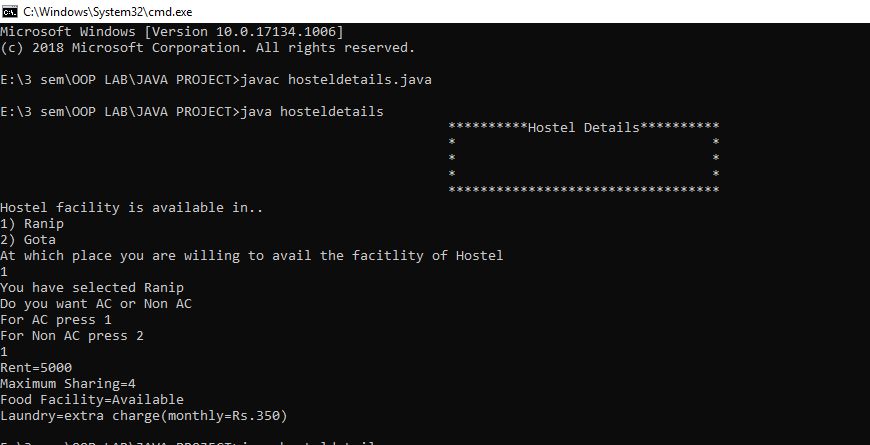
hd.heading();

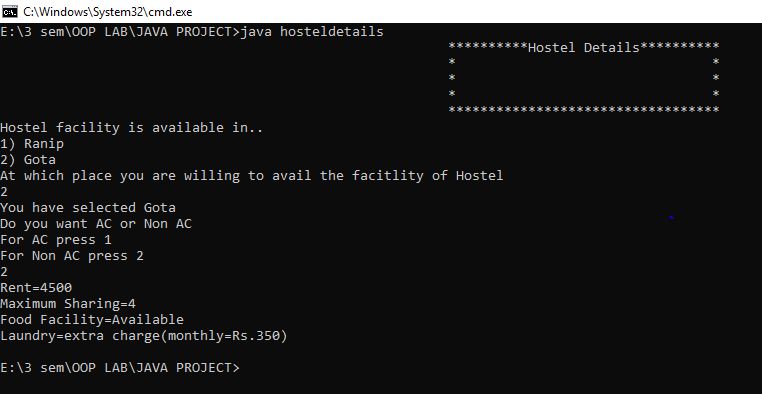
hd.info();

}

}

**OUTPUT**





**EXPLANATION ABOUT PROJECT:**

This project basically has 5 modules:

1. Login Page: it is used for to login the particular student.

2. Staff Details: it keeps the information of the faculties and other staff members.

3. Course Details: shows the information about course selection i.e. which course is to be selected by the student.

4. Time Table Details: It is used to show the time table with day and time for the student and faculties both.

5. Hostel Details: provide information about hostel for both boys and girls at different palces.

**REFERENCES:**

1. <https://www.slideshare.net>

2. Wikipedia(College Management System page)

3. Google Images

4. [www.muengineers.in](http://www.muengineers.in)

5. [www.timeoustechnetronic.com](http://www.timeoustechnetronic.com)

**CONCLUSION:**

1. A better, smart, time efficient and easy to manage alternative of existing ways of managing college/institute record.

2. This project provides a computerized version of college management system which will benefit the students as well as the staff of the College/Institute.