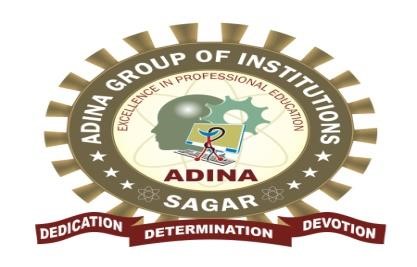
**ADINA INSTITUTE OF SCIENCE & TECHNOLOGYSAGAR (M.P.)**

# Department Of Computer Science and Engineering



**Session: Dec 2021**

**A Minor Project Report**

# ON

**“Student Management System”**

**Submitted in partial fulfilment for the requirement of Bachelor’s Degree in**

**Computer Science and Engineering**

## RAJIV GANDHI PRODYOGIKI VISHWAVIDHYALAYA, BHOPAL

**(University of Technology in Madhya Pradesh)**



**Submitted by: -**

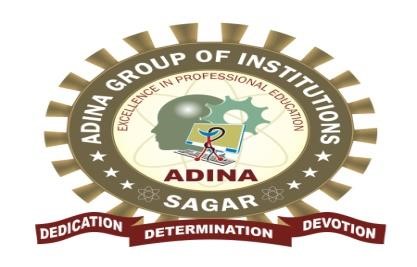
**0612CS191033**

|  |  |  |
| --- | --- | --- |
| **Dr.Anshul Gangele** | **Mr. Rajneesh Pachouri** | **Mr**. **Anurag jain** |
| **Principal** | **H.O.D.** | **Project Incharge** |
| **A.I.S.T. Sagar** | **CSE deptt.** | **CSE deptt.** |

## ADINA INSTITUTE OF SCIENCE &TECHNOLOGYSAGAR (M.P.)

**Department Of Computer Science and Engineering**

**Session: 2021-22**



### Approval Certificate

**This is hereby certified that 0612CS191033 of Computer Science & Engr. Department of this college have submitted the project “Student Management System “for the partial fulfilled his/her Bachelor Degree under the guidance of Asst. Prof. Anurag Jain**

**We wish all the best for their future.**

**ACKNOWLEDGEMENT**

I express my deep sense of gratitude to **Asst. Prof Rajneesh Pachouri** Head, Department of Computer Science and Engineering, Adina Institute of Science and Technology Sagar (M.P), whose kindness valuable guidance and timely help encouraged me to complete this work.

A special thank to **Asst. Prof. Anurag Jain** who helped me in completing the minor project by exchanging her interesting ideas, thoughts which made this work easy and accurate. I am indebted to him for the valuable time he has spared for me during this work.

I express my thanks to **Dr.Anshul Gangele(Principal)** for extending his support. I am also grateful to **Dr. Sunil Jain (Director**), who provided all the official facilities and guidance to me. I also thank the authors whose works have been consulted by me during the research.

I would also thank my institution and my faculty members without whom this project would have been a distant reality.

I take this opportunity to express my regards and obligation to my father and other family members whose support and encouragement I can never forget in my life. At last but not the least I want to thank my best friend who appreciated me for my work and motivated me and finally to god who made all the things possible.

**Nisharg Gupta,**

## DECLARATION

I/We, hereby declare that the work, which is being presented in the Project Report entitled **“Student Management System”** is my/our own work, carried out under the noble guidance of my/our guide **Prof. Anurag Jain**, Department of Computer Science & Engineering, Adina Institute of Science and Technology, Sagar (M.P.).

The matter embodied in this Minor Project Synopsis work has not been submitted in any other university for the award of any degree.

**Nisharg Gupta 0612CS191033**

**Approved by**

|  |  |  |
| --- | --- | --- |
| **Dr. Anshul Gangele** | **Mr. Rajneesh Pachouri** | **Mr. Anurag Jain** |
| **Principal** | **H.O.D** | **Project Incharge** |
| **A.I.S.T** | **CSE deptt.** | **CSE deptt.** |

## CERTIFICATE

This is to certify that the Minor project entitled “**Student Management System**” the Bonafede Minor project work carried out independently **Nisharg Gupta** student of Bachelor of Technology in **Computer Science Engineering** from Adina institute of Science and Technology Sagar, affiliated by **RGPV Bhopal (M.P)**

**INTERNAL EXAMINER** **EXTERNAL EXAMINER**

**Table of content:**

1. **Acknowledgement……………………………………………………………………...6**
2. **Declaration……………………………..……………………………………………….7**
3. **Certificate……………………………………………………………………………….8**
4. **Software Requirement…………………………………………………………………9**
5. **Pros and Cons…………………………..………………………………………………9**
6. **Data flow Diagram…………………….………………………………………………10**
7. **Software Development Model……….………………………………………………..11**
8. **Relational Table………………………..……………………………………………...12**
9. **Software working and explanation…....……………………………………………...21**
10. **Project Detail…………………………………………………………………………..23**
11. **Relational between table……………….……………………………………………...24**
12. **Conclusion……………………………………………………………………………...25**

**STUDENT MANAGEMENT PROJECT with DATABASE**

**Requirement:**

* JDK i.e., Java Development kit in order to run Software you must need this.
* POSTGRESQL JDBC DRIVER
* RAM >=8 GB
* HARD DISK >=400 MB

**PROS:**

* Fast
* Require Low Space
* Easy
* Lot to learn (which we studied till now we implement that)

**CONS:**

* Minor
* Easy Algorithm
* Not so much user friendly

**Features**

* Login Page
* Home Page
* School Year Page
* Grade List Page
* Student List
* Subjects List
* Curriculum List
* Academic Record
* Promote Candidates
* Candidate List
* Faculty Page
* Management Page

**DATA FLOW DIAGRAM:**

**Management**

**DATABASE**

…

…….

………..

. .

. .

……….

………..

. .

. .

**STUDENT FACULTY**

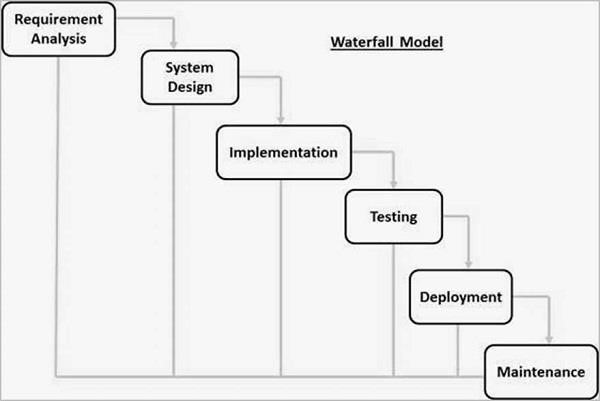
**Software Model used to develop the project –**

**Water fall model----**

The Waterfall Model was the first Process Model to be introduced. It is also referred to as a linear-sequential life cycle model. It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases.

The Waterfall model is the earliest SDLC approach that was used for software development.

The waterfall Model illustrates the software development process in a linear sequential flow. This means that any phase in the development process begins only if the previous phase is complete. In this waterfall model, the phases do not overlap



Firstly, we gather information that required to develop the software, after that we develop Schema

of Project and Algorithm for coding.

As waterfall Model say’s we do the same thing First we make one class completely then after we switch to another class

**Schema**

**Software Management System Tables**

These tables below provide the complete database tables details such as Field Name, Descriptions, data types, character lengths.

**Table Name: students**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Description** | **Type** | **Length** |
| sid (PK) | Student ID Number | Int | 11 |
| sname | Student First Name | Varchar | 30 |
| sDOB | Student Last Name | Varchar | 30 |
| sBranch | Student Branch | Varchar | 30 |
| syear | Student Year | Int | 11 |
| MobieNo | Student Contact Number | Int | 11 |
| castype | Student cast | varchar | 3 |
| joindate | Student enrol Date | Date |  |
| gender | Student Gender | Int | 11 |
| Feesstatus | Fees Status paid or unpaid | varchar | 6 |
| Attendance | Attendance day wise | boolean |  |
| scity | Student city | varchar | 50 |
| sstate | Student | Varchar | 30 |
| saddr | Student | Varchar | 100 |
| spassword | Student | varchar | >8 |

**Table Name: Faculty**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Description** | **Type** | **Length** |
| fId (PK) | Staff ID Number | Int | 11 |
| fname | Staff Name | Varchar | 30 |

|  |  |  |  |
| --- | --- | --- | --- |
| fDOB | Faculty DOB | Varchar | 30 |
| fsalary | Staff Salary | Int | <=45000 |
| faddr | Staff Address | Varchar | 255 |
| gender | Staff Gender | Varchar | 30 |
| Fpassword | Login password | varchar | >=8 |
| Fcity | Faculty city | varchar |  |
| Fstate | Faculty state | varchar |  |
| Joindate | Joining date | Date |  |
| Currentdate | Current Date | Date |  |
| fexperience | Experience | Int |  |
| mobileno | Mobile number | Int | 10 |
| post | Current post of faculty | varchar | 10 |
| attendance | Faculty attendance | Boolean |  |
| salarystatus | Salary status | varchar | 6 |

**Table Name: Fees Structure**

|  |  |  |  |
| --- | --- | --- | --- |
| **Field Name** | **Description** | **Type** | **Length** |
| yearly | i.e., ‘First’,’Second’,’Third’,’Fourth’ | varchar | 11 |
| Branch | i.e.,’CS’,’ME’,’CSE’,’EC’ | varchar | 4 |
| SC\_Boy | SC boy fees | Int | <=45000 |
| ST\_Boy | ST boy fees | Int | <=50000 |
| OBC\_Boy | OBC boy fees | Int | <=25000 |
| SC\_Girl | SC girl fees | Int | <=50000 |
| ST\_Girl | ST girl fees | Int | <=55000 |
| OBC\_Girl | OBC girl fees | Int | <=45000 |

**Algorithm-**

Student can

* See his Information
* Update some information

Faculty can

* See his Information
* Update some information

Management can

* Update existing Faculty and Student can Enroll new Faculty and Students

So, from this we can conclude that Student and Faculty can see or update database.But Management only update database.

**DATABASE CODE:**

CREATE DATABASE adina

WITH

OWNER = postgres

ENCODING = 'UTF8'

CONNECTION LIMIT = 15;

create schema if not exists AdinaInfo;

set search\_path to public;

create table Information

(

courseCode BIGSERIAL not null primary key,

course varchar(15) not null,/\*Btech Mtech etc\*/

branch varchar(15) unique not null,/\*CS ME CSE \*/

courseDuration smallint not null check(courseDuration>2),

seat smallint not null check(seat<=60),

numberOfFaculty smallint not null check(numberOfFaculty=60)

);

insert into Information (course,branch,courseDuration,seat,numberOfFaculty) values

(

'Btech',

'CSE',

4,

60,

60

);

create table feesStructure

(

yearly varchar(20) not null, /\*First Second Third Fourth\*/

branch varchar(15) not null, /\*CS ME CSE \*/

SC\_Boy int not null check(SC\_Boy<=45000) default 45000,

ST\_Boy int not null check(ST\_Boy<=50000) default 45000,

OBC\_Boy int not null check(OBC\_Boy<=25000) default 20000,

SC\_Girl int not null check(SC\_Girl<=50000) default 45000,

ST\_Girl int not null check(ST\_Girl<=55000) default 45000,

OBC\_Girl int not null check(OBC\_Girl<=45000) default 45000,

primary key(yearly,branch),

foreign key(branch) references Information(branch) on delete cascade

);

select \* from information;

insert into feesStructure values

(

'Fourth',

'CSE'

);

select \* from feesstructure;

create table Faculty

(

/\* Count == Information(numberOfFaculty) \*/

fId bigserial NOT NULL primary key,

fName varchar(80) not null, /\*Error\*/

fDOB date,

fSalary int not null check(fSalary<=45000) default 45000,

fPassword varchar(12) not null check(length(Fpassword) >= 8),

fcity varchar(30),

fstate varchar(50),

fAddr varchar(100),

joinDate date not null,

currentDate date,

fExperience smallint not null,

mobileNo int8 not null,

post varchar(78),

attendance bool,

gender varchar(6),

salarystatus varchar(6)

/\*

True means present

False means Absent

Count number of True based on 1 month and calculate percentage which give percentage

\*/

);

create table Students

(

sId bigserial NOT NULL primary key,

sName varchar(85) not null,

sDOB date,

sBranch varchar(30) not null,

scholarship bool,

gender varchar(6) not null,

castype varchar(10), /\*SC ST OBC General\*/

spassword varchar(12) not null check(length(spassword) >= 8),

scity varchar(30) not null,

sstate varchar(50) not null,

sAddr varchar(100),

joinDate date not null,

currentDate date,

semester smallint not null,

mobileNo int8 not null,

Attendance bool,

feestatus varchar(6),

SYear varchar(10),

foreign key(sBranch) references Information(branch)

);

insert into faculty (

fname,

fdob,

fsalary,

fpassword,

fcity,

fstate,

faddr,

joindate,

currentDate,

fexperience,

mobileno,

post,

attendance,

gender,

salarystatus

)

values

(

'Rajneesh Pachauri',

'2002-01-08',

26000,

'Karma123@',

'Sagar',

'Madhya Pradesh',

'Neha nagar makronia sagar',

'2008-02-16',

current\_date,

5,

9300707154,

'Asst Professor',

'y',

'male',

'unpaid'

);

insert into Students

(

sName,

sDOB,

sBranch,

scholarship,

gender,

castype,

spassword,

scity,

sstate,

sAddr,

joinDate,

currentDate,

semester,

mobileNo,

Attendance,

feestatus,

Syear

)values

(

'Kula Lumper',

'2002-06-12',

'CS',

'true',

'male',

'SC',

'Karma123@',

'Sagar',

'Madhya pradesh',

'bada bazaar sagar madhya pradesh',

'2021-02-12',

current\_date,

5,

9516750048,

'y',

'unpaid',

'First'

);

select \* from faculty;

select \* from feesstructure;

select \* from information;

select \* from students;

create table Fattendance

(

fid integer,

DateP date,

Attendance boolean,

foreign key(fid) references Faculty(fid) on delete cascade

);

create table Sattendance

(

sid integer,

DateP date,

Attendance boolean,

foreign key(sid) references students(sid) on delete cascade

)

**Data is important while developing an application. In order to store data or retrieve data efficiently we used database.**

**The database which we used is postgres because it has**

* Open-Source DBMS.
* Diverse Community.
* Function.
* ACID and Transaction.
* Diverse indexing techniques.
* Flexible Full-text search.
* Diverse kinds of replication.
* Diversified extension functions.
* Variety of datatype

**Here are the List of Modules that the Student Management System should have:**

* **Student Information Management**: The data that are gathered form the student information must be secured in this system. These data were composed of the student basic information and academic status.
* **Enrolment Management**: This is the process where the system takes care of the requirements provided by the student enrolees. They will require the birth cert of the enrolees as well as their academic status that will serve as their basis if the student is new or continuing.
* **Grade and Subject Management**: This module designates the instructors per subjects to every student or enrolees.
* **Fee and Payment Management**: Management of fee and payment is also covered in student management system. The fee and payment information is processed from the enrolment and then the parents or students will be updated about their accounts to avoid conflicts.

# Step to work with Postgres-

1.

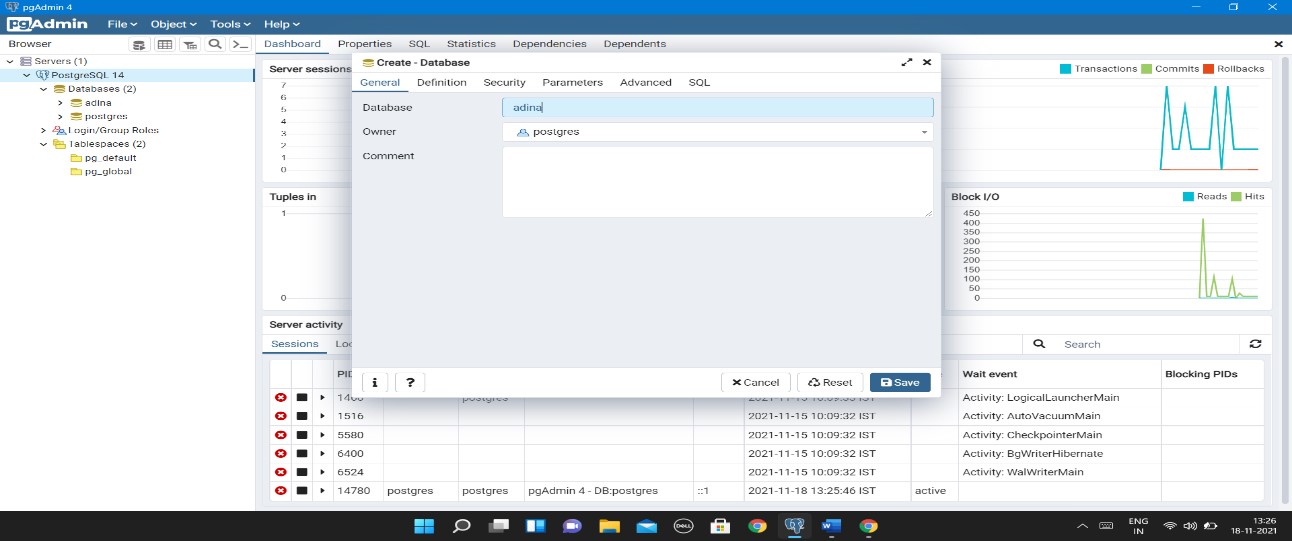
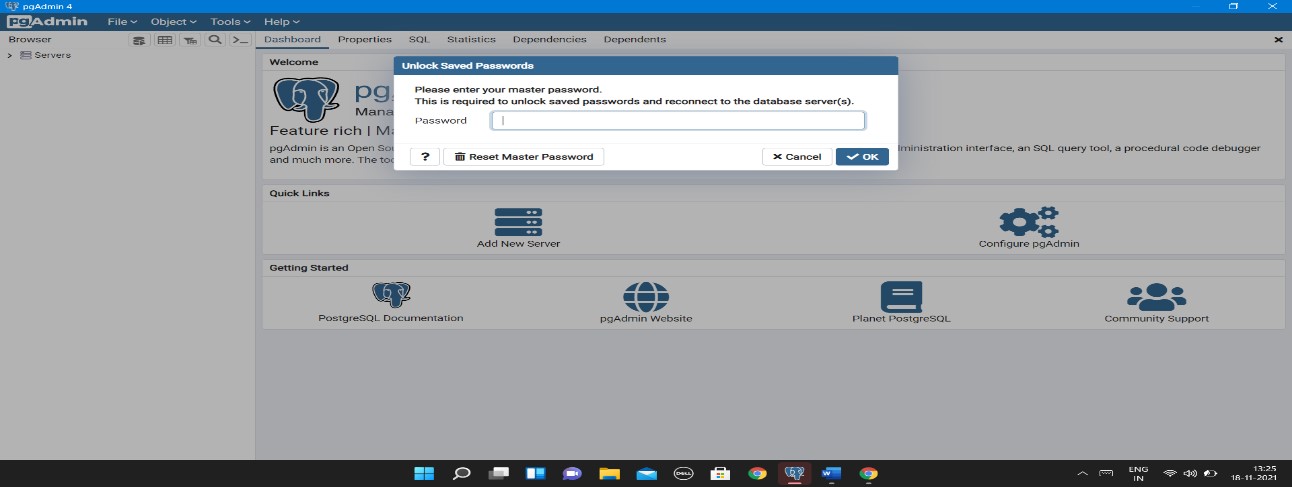
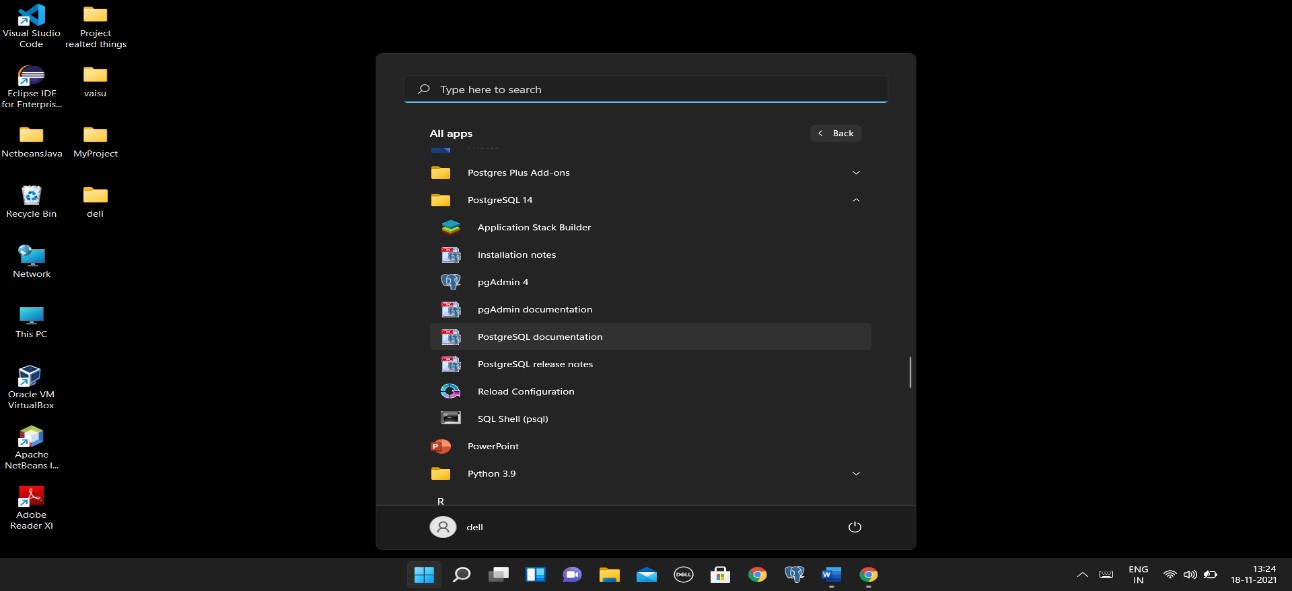
Download and install postgres.

2.

Open “pgadmin” password is “root”

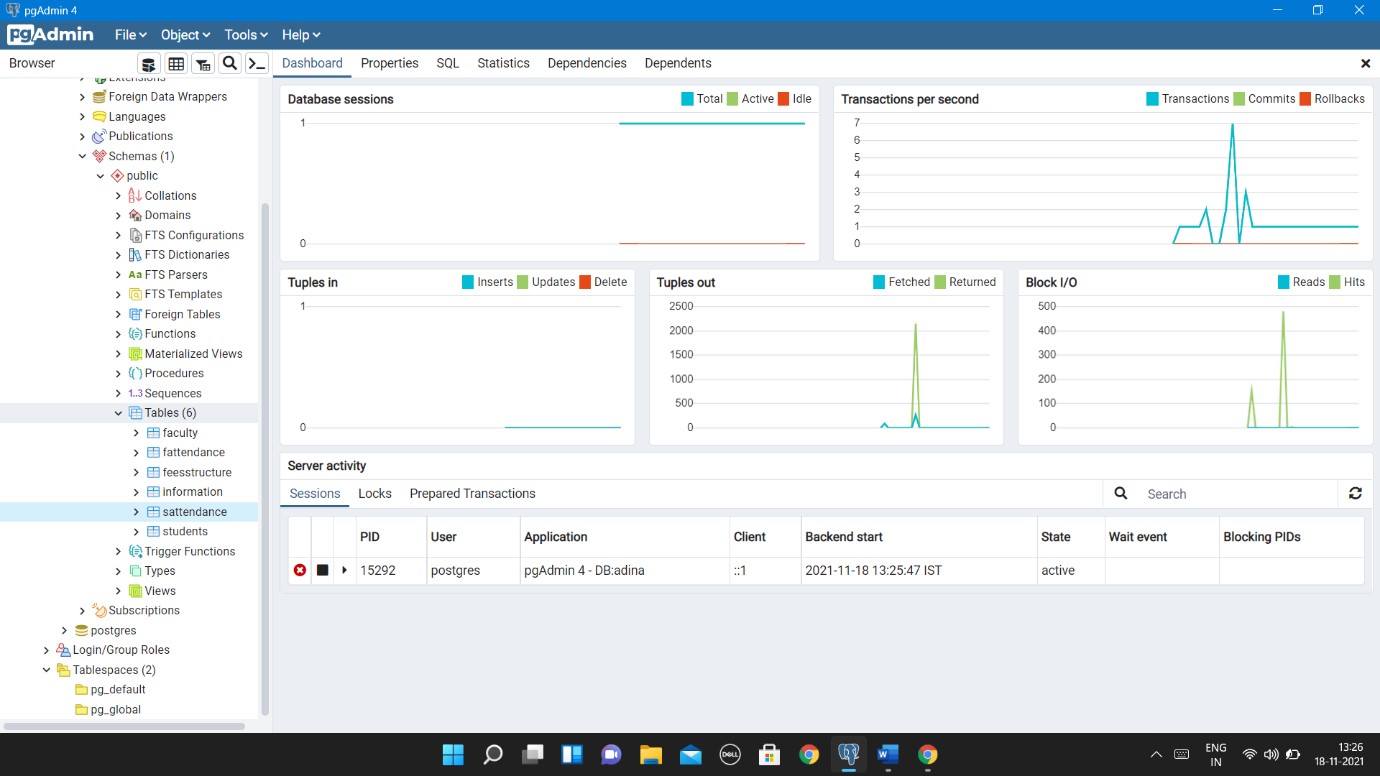
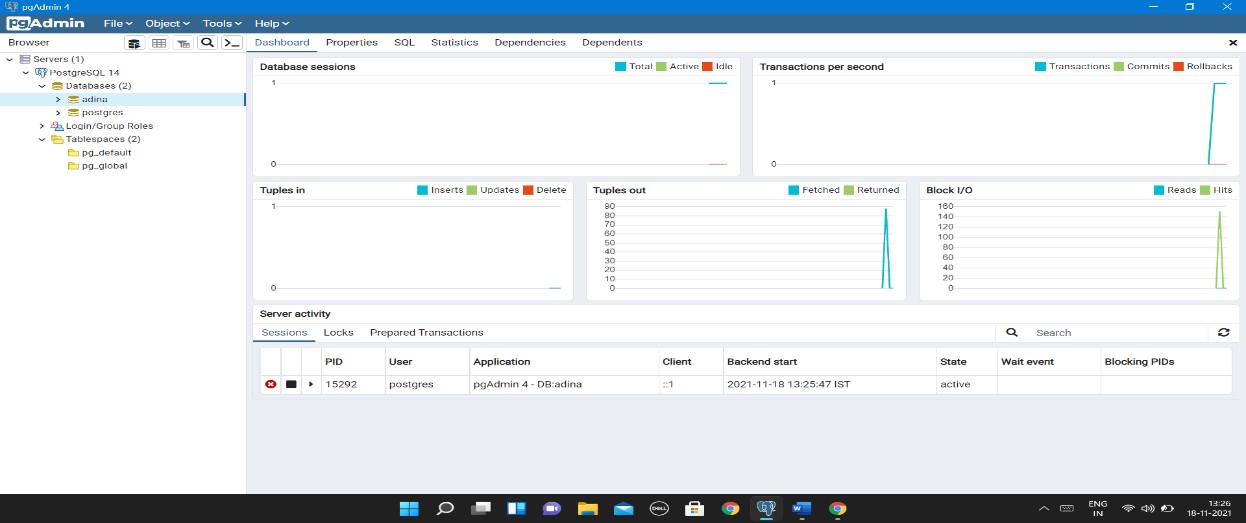
3.

Created Database named “adina”



4.

Created multiple tables.

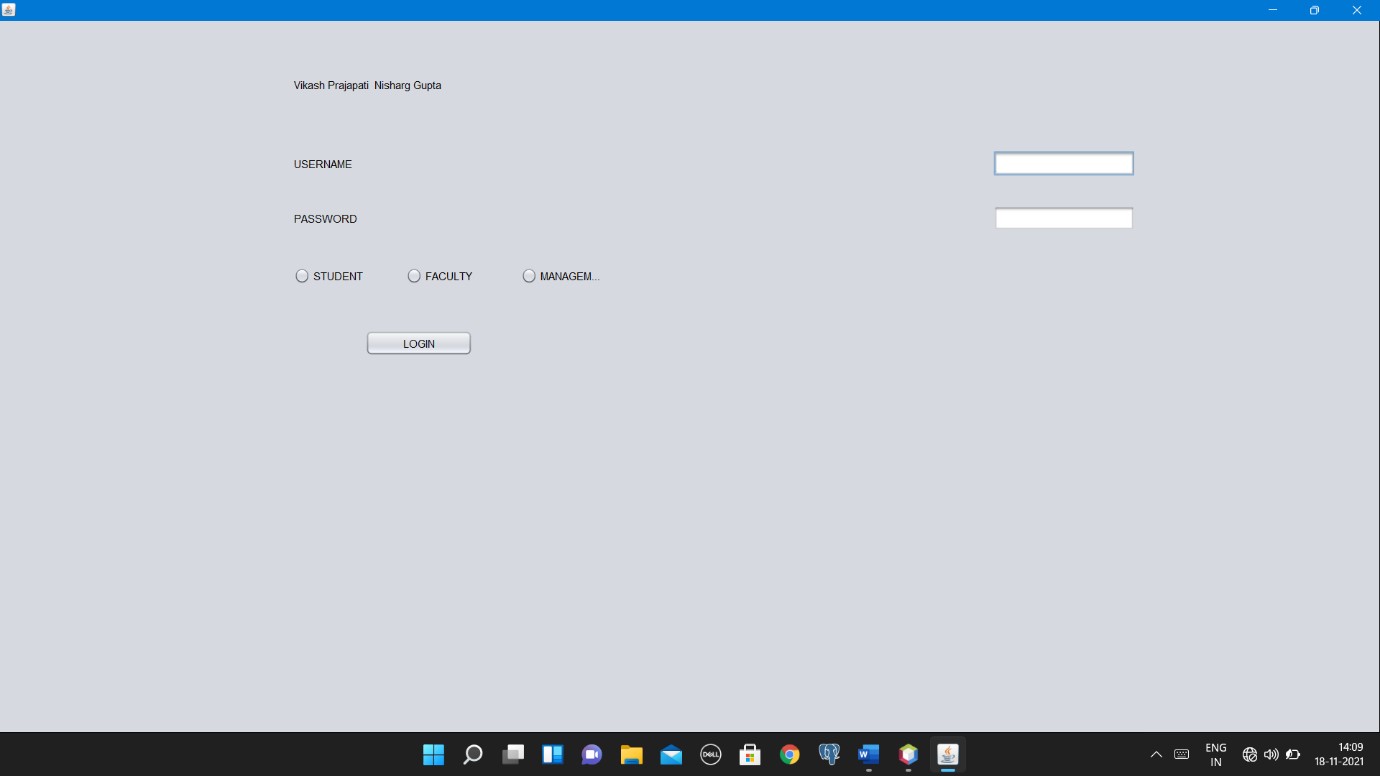


1. Table that we created are
   * faculty for faculty information
   * fattendance for attendance of faculty(on day based).
   * feestructure for structure of fees for student who enroll in college.
   * information for information of college what course it will offer which branch etc.
   * students for storing student who enroll.
   * sattendance for storing students’ attendance (on day based).

1. After that we developed GUI based front end interface with the help of java swing.

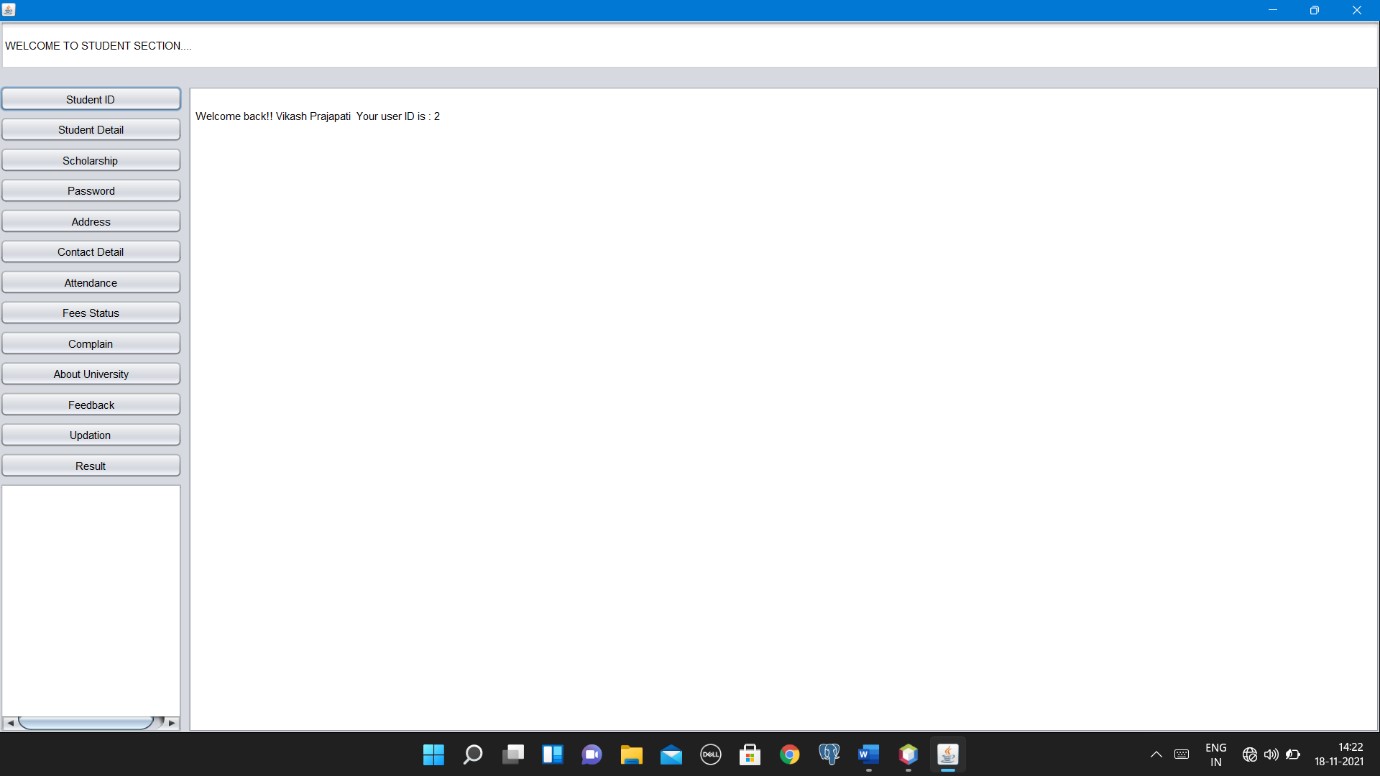
We created multiple classes for each interface.

Basically, we have 3 interface student, faculty and management

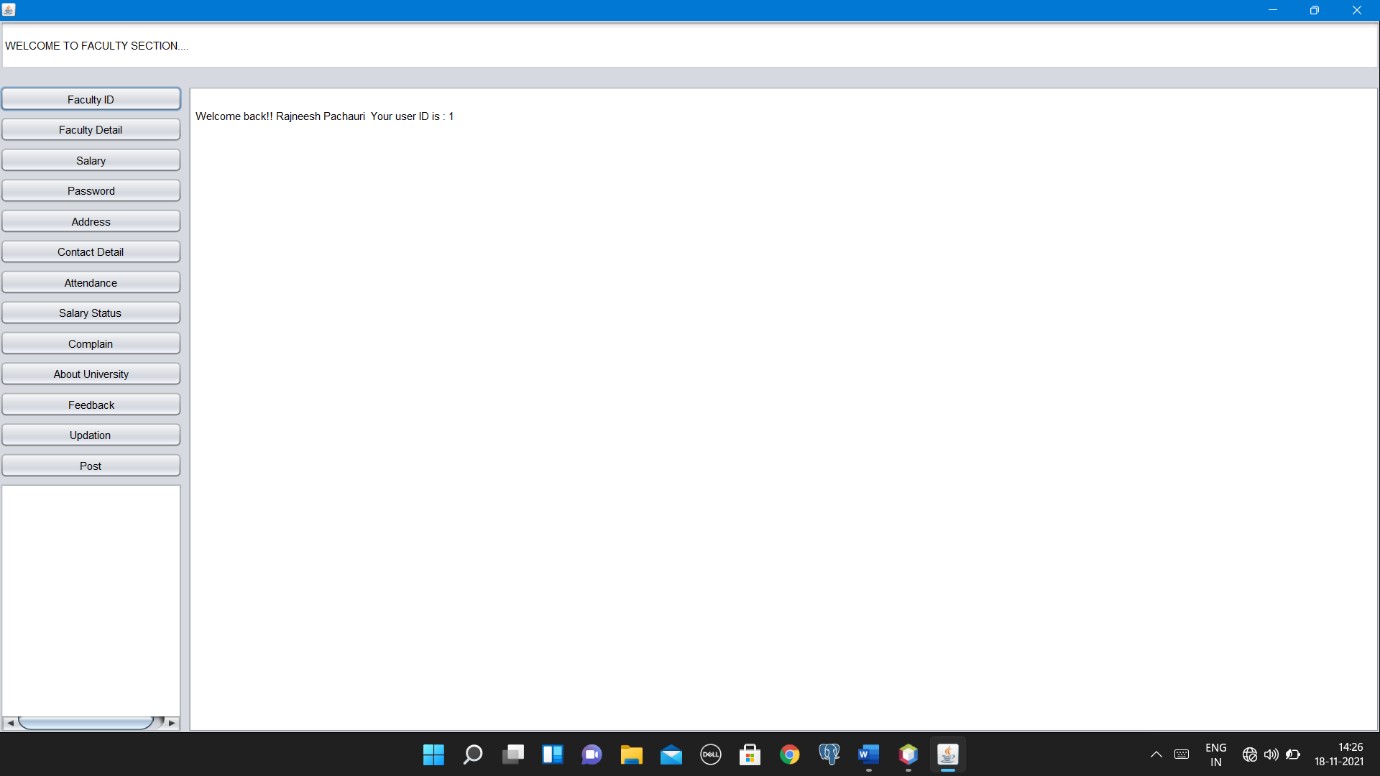


It requires userid and password and type.

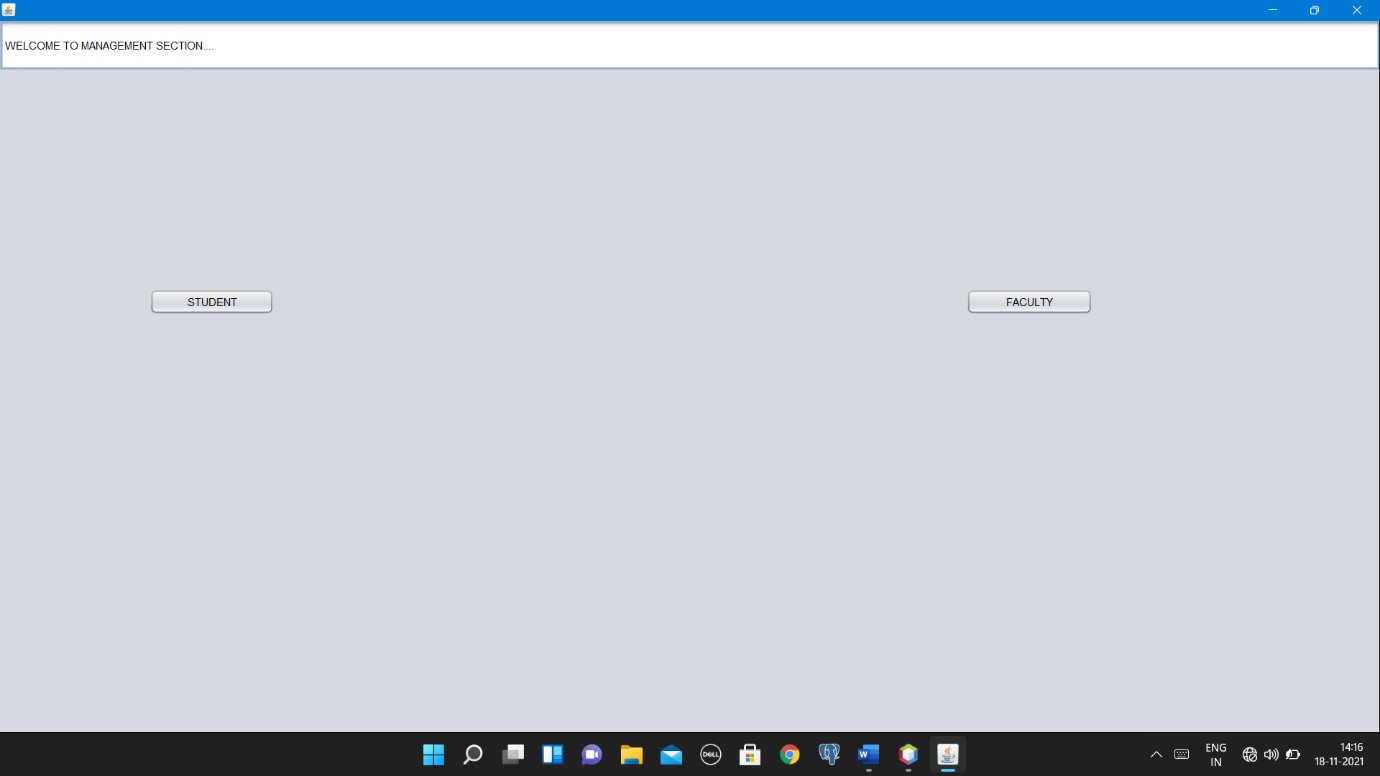
If it will login as student only after if userid and password exist into database then new interface has been open like this.



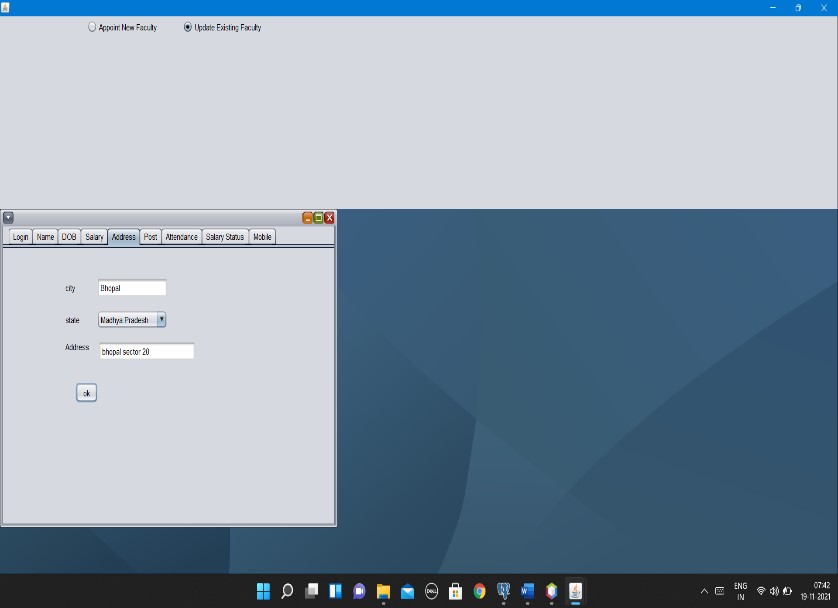
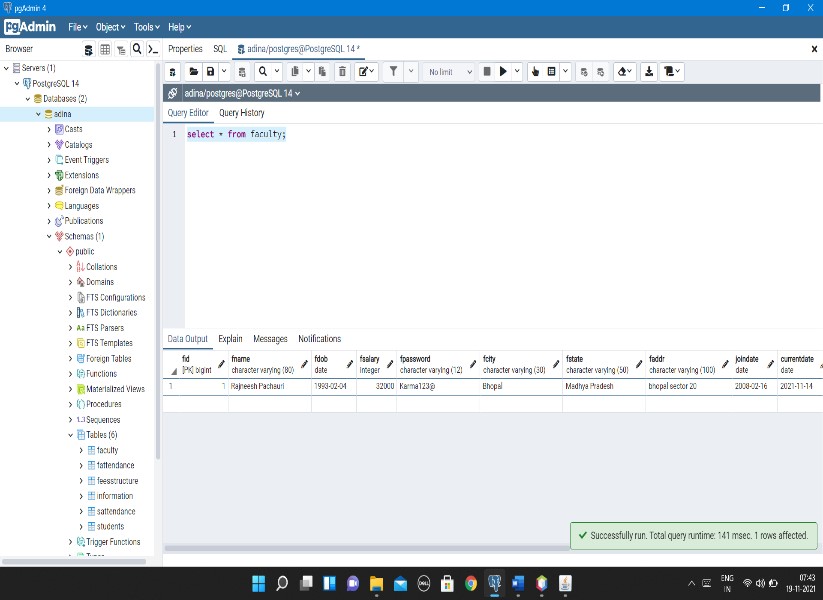
If it will login as faculty only after if userid and password exist into database then new interface has been open like this.



If it will login as management then interface has been open like this.



After update what happen let see.



As wee see fcity change to Sagar to Bhopal. In the similar way we can change name, address, DOB, Post, Pay Scale, Mobile number etc.

# Project Detail: -

**We Develop this project with the help of java swing GUI widget toolkit.**

|  |
| --- |
| import static java.awt.EventQueue.invokeLater; import java.awt.Font; import java.awt.event.ActionEvent; import java.awt.event.MouseEvent; import java.sql.Connection; import java.sql.DriverManager; import java.sql.SQLException; import java.sql.PreparedStatement; import java.sql.ResultSet; import java.time.LocalDate; import java.util.Calendar; import java.util.TimeZone; import java.util.logging.Level; import java.util.logging.Logger; import javax.swing.AbstractListModel; import javax.swing.JFrame; import javax.swing.ButtonGroup; import javax.swing.DefaultComboBoxModel; import javax.swing.GroupLayout; import javax.swing.JButton; import javax.swing.JCheckBox; import javax.swing.JComboBox; import javax.swing.JDesktopPane; import javax.swing.JFormattedTextField; import javax.swing.JInternalFrame; import javax.swing.JLabel; import javax.swing.JList; import javax.swing.JPanel; import javax.swing.JPasswordField; import javax.swing.JRadioButton; import javax.swing.JScrollPane; import javax.swing.JTabbedPane; import javax.swing.JTable; import javax.swing.JTextArea; import javax.swing.JTextField; import javax.swing.LayoutStyle;  import javax.swing.LayoutStyle.ComponentPlacement; import javax.swing.UIManager.LookAndFeelInfo;  import static javax.swing.UIManager.getInstalledLookAndFeels; import static javax.swing.UIManager.setLookAndFeel; import javax.swing.UnsupportedLookAndFeelException; import javax.swing.WindowConstants; import javax.swing.table.DefaultTableModel; import javax.swing.table.TableModel |

**Above Package has been Required to Develop this Project.**

1. Now Let talk About Classes, So Total 24 classes has been Required including main class also.
2. Total 6106 lines of code has been there.

# Relationship between Student, Faculty, Management, Information, and Fees Structure Table-

Faculty

Fid

F

name

course

FeesStructure

Course

Fees

Information

course

Adina

Students

Sid

Sname

course

**Conclusion: things get after developing the software.**

1. Lot to learn about project development strategies.
2. How to work on ideas
3. Project implementation and procedure
4. Problem solving skill by solving bugs and error.
5. Approach to solve problem.
6. Determination of bugs.
7. Enhancing quality of product.