MINI
Project Report
2020-2021

PROBLEM STATEMENT:

DESIGN A COURSE REGISTRATION APPLICATION FOR THE STUDENTS TO REGISTER FOR ELECTIVE AND OPEN ELECTIVE SUBJECTS.

~The application must have a provision to upload the syllabus and auto cut-off strength for each course

College: CMR INSTITUTE OF TECHNOLOGY
Branch: COMPUTER SCIENCE ENGINEERING
Team Number: TEAM 77
Name: C. K. AMRUTHA (1CR19CS032)
DADIREDDY SAI KUMAR REDDY(1CR19CS035)
DARSHAN R (1CR19CS037)
DUSI UJWAL(1CR19CS051)
Guide: Mrs. SAVITHA N J

TABLE OF CONTENTS

Abstract

- A course registration system (CRS) created using object oriented programming.

REQUIREMENTS:

- The CRS stores the following information about each course:
    1. course name
    2. course ID
    3. max # of students registered in the course
    4. current # of registered students
    5. list of names of students being registered in the course
    6. course instructor
    7. course section #
    8. course location
- The CRS allows two types of users: Admin and Student
- The CRS allows the Admin to perform the following tasks:
    1. Course Management
        - Create a new course
        - Delete a course
        - Edit a course (except for course ID and name)
        - Display info for a course (by course ID)
        - Register a student (allows admin to add student w/o assigning to a course)
        - Exit
    2. Reports
        - View all courses (for every course display list of enrolled students' names, ids, # of students registered, max # of students allowed to be registered)
        - View all full courses
        - Write to a file list of full courses
        - View names of students registered to specific course
        - View list of courses a student is registered in (given first & last name of student)

- Sort courses on current # of students registered
- Exit

- The CRS allows the Student to perform the following tasks:

1. Course Management
    - View all courses
    - View all courses that are NOT full
    - Register in a course (student enters course name, section, student's full name)
    - Withdraw from a course (student enters their name, course name)
    - View all courses student is registered in
    - Exit

- The CRS must implement the following design:

1. An *Interface* for admin class with signatures of methods used by admin.
2. An *Interface* for student class with signatures of methods used by student.
3. Both Admin and Student classes inherit from class User. User has the following class members:

    1.Username 2.Password 3. first & last name

4. When program is launched, the CRS must read all courses' information from a file (MyUniversityCourses.csv).
5. Assume there is only one Admin in the program.
6. Serialization & deserialization will be used to read the csv file and write into a new file.

Learning Objectives

The basic objective of taking up this project was to increase our knowledge in the language of Java. Also, this project gave an edge for us as a team to explore Java not only as programming language but as a tool to easily solve problems. Netbeans was used throughout the project.

We also explored Xampp server, MySQL commands, Apache these increased the scope of our project and helped towards completion and knowledge growth.

After the completion of the experiment our team and I, with the help of our mentor are confident enough that the model created by us is ready for the practical use. Throughout the course of this project, we have learned a lot of things concerned with Java, MySQL commands etc. Altogether, we would say we have come across various things during this course and learned and explored a lot of things that are out of our engineering scope.

Tasks Handled

Some of the main tasks in order to complete this project are given below:

Technical	Non-Technical
Setting up the technicalities	Research
Downloading required packages.	Innovation
Complete some online course	Team building
Coding	Scope of up-gradation
Debugging	Report Writing
Execution	Power point presentation
Development	
Testing	

WORK DISTRIBUTION

C. K. AMRUTHA	DADIREDDY SAI KUMAR REDDY	DARSHAN R	DUSI UJWAL
Research Execution Report-Writing Coding	Research Execution Planning Presentation Coding	Research Execution Planning Presentation Coding	Research Execution Presentation Coding Debugging Testing

As, seen above the tasks had been equally distributed and handled in a systematic manner. We divided our work according to the strengths we had which helped in executing the model perfectly.
There are minor other tasks like coordinating, communicating with people who are professional coders and programmers for better execution, which I have not considered here.

Implementation

First we have created a project named Student

>>Under default package we have added JFrame Form named as Login

- Once we obtain a form we add a panel to put our choices,buttons,text etc..

Login-

- In this we have a logo of the institute.
- labels namely Email and Password with text field to fill in respectively.
- buttons namely Login and Cancel.

The login button successfully takes you to the Home page whereas Cancel button allows you to exit. We will create our 2nd JFrame Form named as Home Page for admin. Once we obtain a form we add a panel to put our choices, buttons, text,etc..

Home Page-

- We have a label named ADMIN PORTAL.
- We have a menu bar that contains.

File – where we have Home and Logout as items

- Also we have buttons namely
  1. Student
  2. Admin
  3. Courses

- After selecting **STUDENT** in the admin portal again we get three options as follows
  1. ADD STUDENT
  2. OPERATION
  3. SHOW STUDENTS

- After selecting **ADMIN** in the admin portal again we get three options as follows
  1. ADD ADMIN
  2. OPERATION
  3. SHOW ADMIN

- After selecting **COURSE** in the admin portal again we get three options as follows
  1. ADD COURSE
  2. OPERATION
  3. SHOW COURSES

We will create our 3rd JFrame Form named as **Add Student** i.e, subclass of student

- It contains various labels namely
  1. Name
  2. Father's Name
  3. City
  4. Blood Group
  5. Phone No
  6. Sem
  7. Email ID
  8. Password
- All these labels except Blood Group have their respective text fields to fill in the details whereas for Blood Group we have used a combo box for selecting out from multiple options(ex: A+ve,B+ve..)
- Then have two buttons namely –
  - Submit – To submit the entered details and proceed
  - Cancel – to exit from the Add Student Form

We will create our 4th JFrame Form named as Operation i.e. subclass of student

- It contains various labels namely
    1. ID
    2. Name
    3. Father's Name
    4. City
    5. Blood Group
    6. Phone No
    7. Sem
    8. Email ID
    9. Password
- We have buttons namely Update, Delete, Search, Back.

We will create our 5th JFrame Form named as Show students i.e. subclass of student

- It contains a table with a list of items-
    1. ID
    2. Name
    3. Father's Name
    4. City
    5. Blood Group
    6. Phone No
    7. Sem
    8. Email ID
    9. Password

We will create our 6th JFrame Form named as Add Admin i.e. subclass of ADMIN

- It contains various labels namely
    1. Name
    2. Email ID
    3. Password
- We have two buttons namely
    1. Submit
    2. Cancel

We will create our 7th JFrame Form named as operation i.e. subclass of ADMIN

- It contains various labels namely
    1. ID
    2. Name
    3. Email ID
    4. Password
- We have four buttons namely
    1. Update
    2. Delete
    3. Submit
    4. Cancel

We will create our 8th JFrame Form named as show admins i.e. subclass of ADMIN

- It contains table with following items-
    1. ID
    2. Name
    3. Email ID
    4. Password
- We have a button
    1. Back

We will create our 10th JFrame Form named as ADD COURSE in ADMIN PORTAL

- In this add course section,

We added a combo box for selection of type of course[PEC/PCC]

- Later we have many text fields to enter title of the course, duration, maximum seats and to add syllabus.
- We added two buttons
  1. Submit
  2. Cancel

We will create our 11th JFrame Form named as Course Operation i.e. subclass of course

- we have many text fields to edit title

of the course, duration, maximum seats
and to add syllabus.

- We have buttons namely Update, Delete,

Search, Back**.**

We will create our 12th JFrame Form named as Show Courses i.e, subclass of course

- We have four tables namely PCC, PCC syllabus, PEC, PEC syllabus.
- We have a Back button.

We will create our 13th JFrame Form named as View courses i.e. subclass of student.

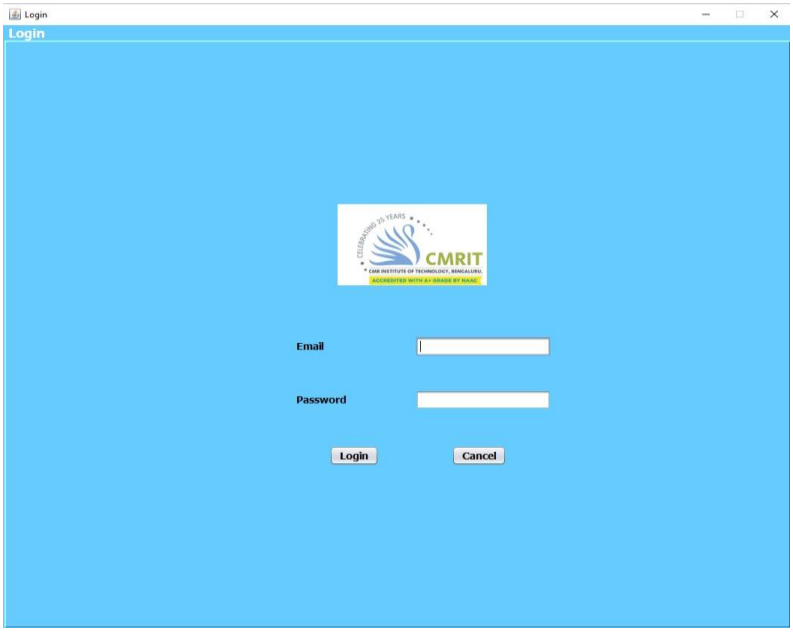
- Here we have three tables namely PCC, PCC syllabus, PEC syllabus for opted course.
- We have a Back button.

We will create our 14th JFrame Form named as Course registration i.e, subclass of student.

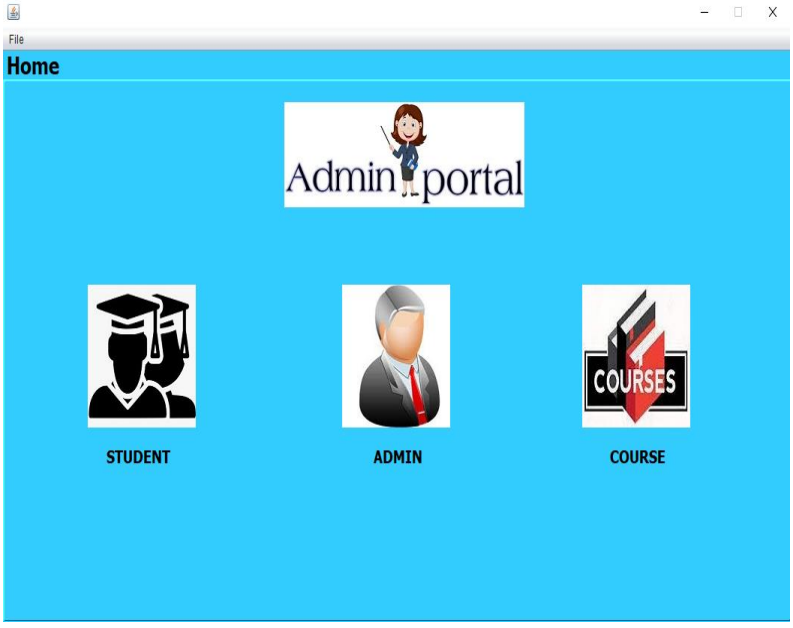
- Here we have two tables namely PEC and PEC syllabus.
- There is a text field to enter the couse id that to be registered.
- We have Register, Back buttons.

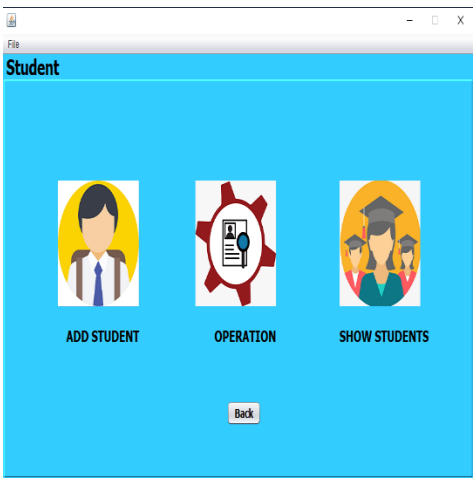
SCREENSHOTS

Login Page

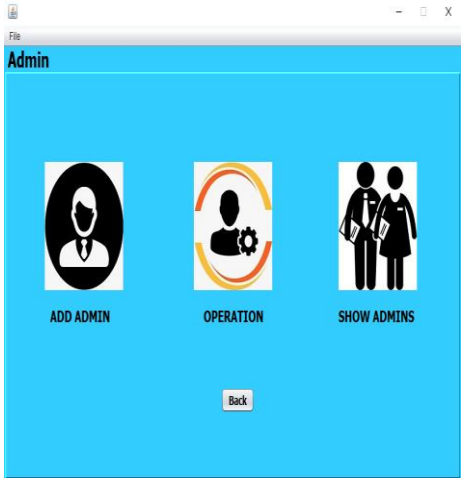


Home Page

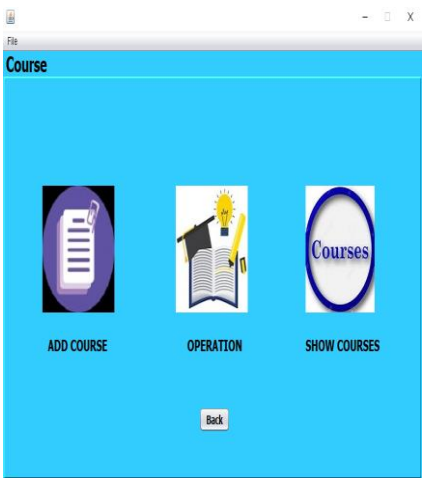




`Student Page

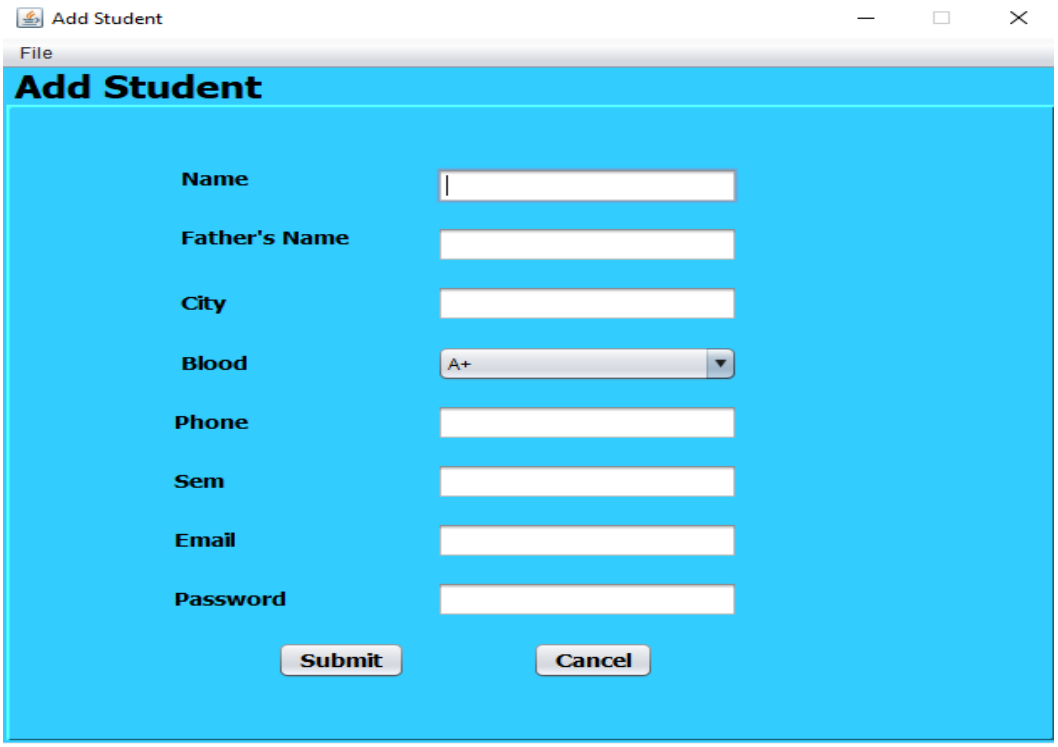


Admin Page



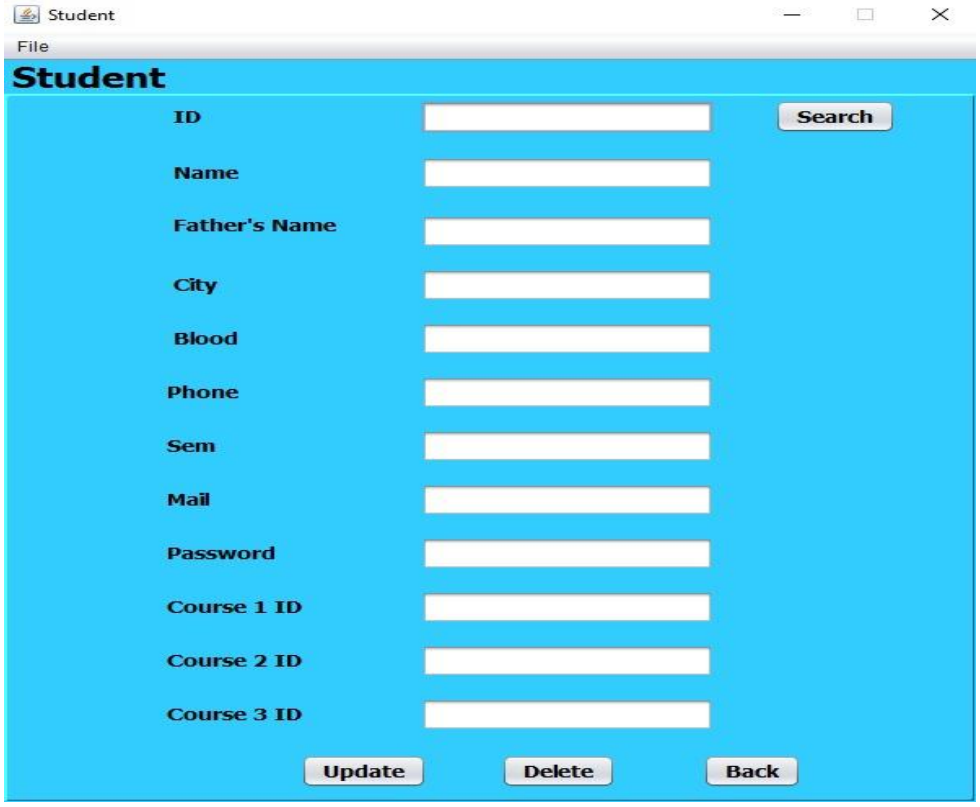
Course Page

In Student Page:

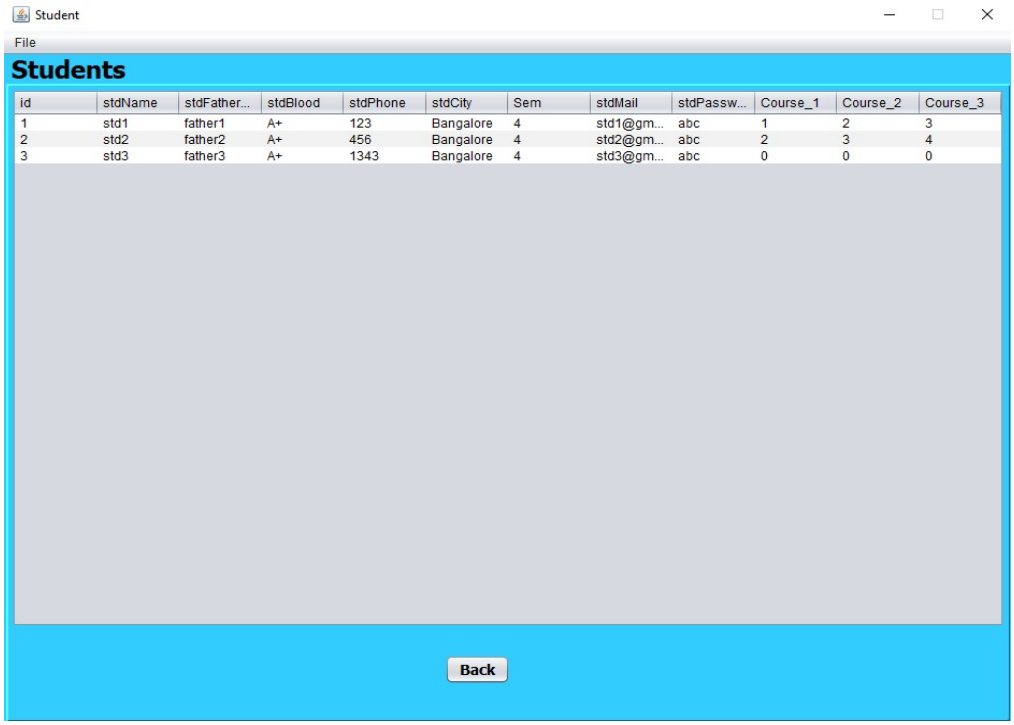


To add a student

Under Operations

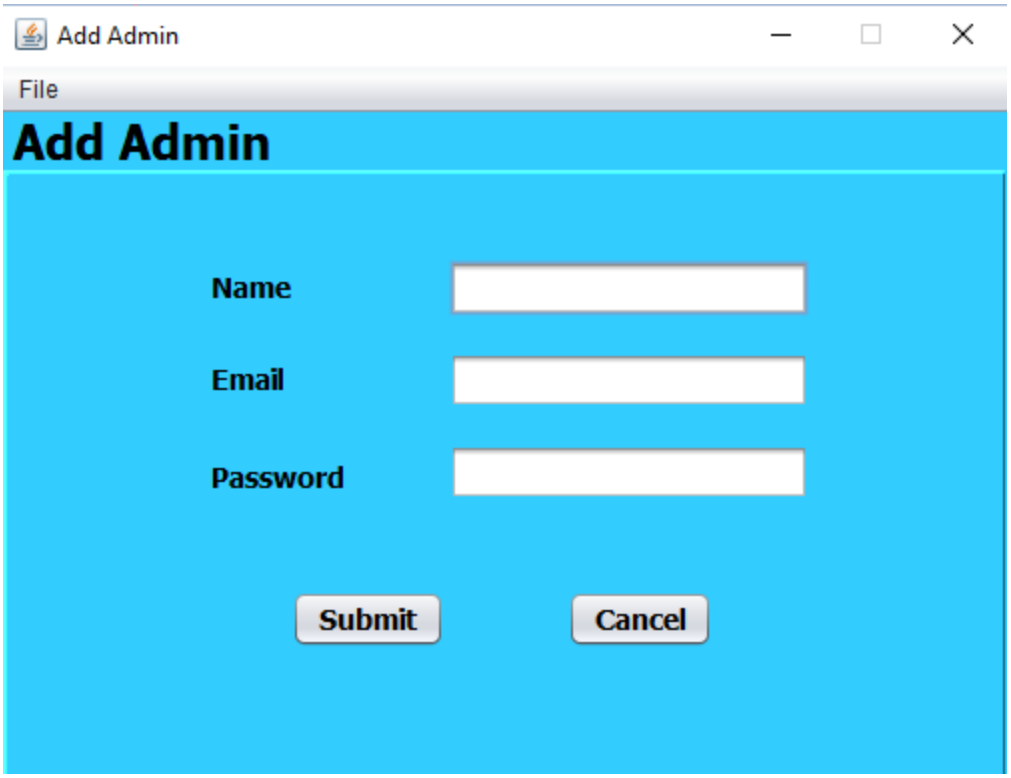


Show Students

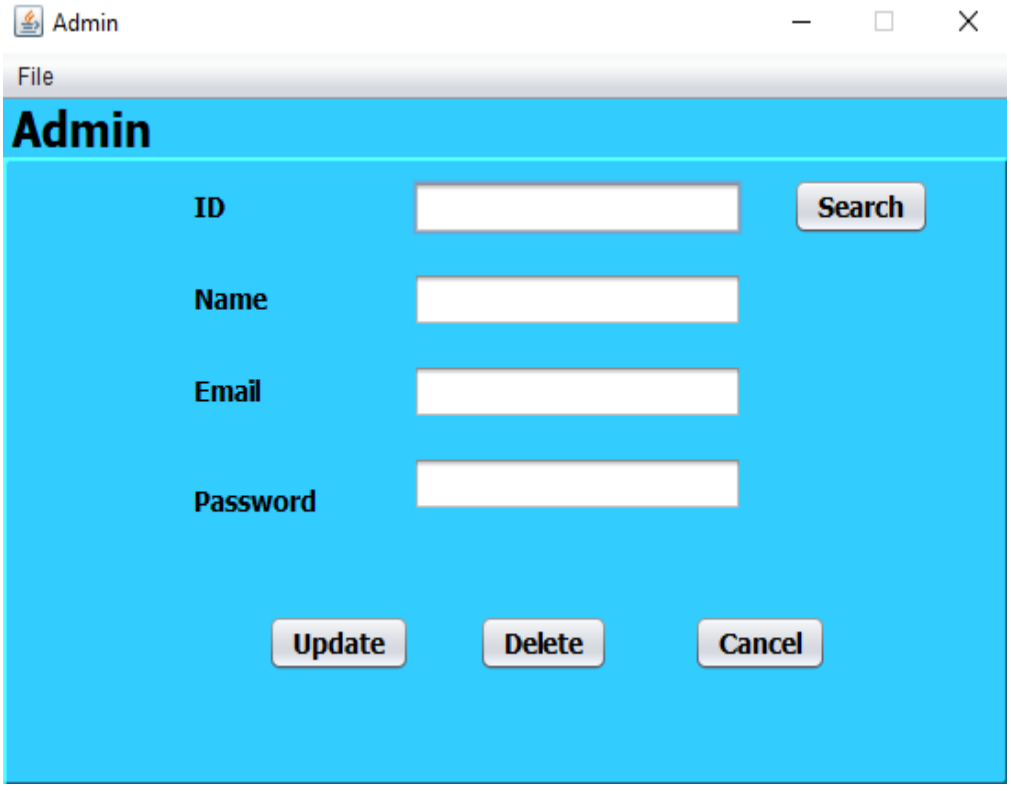


In Admin Page

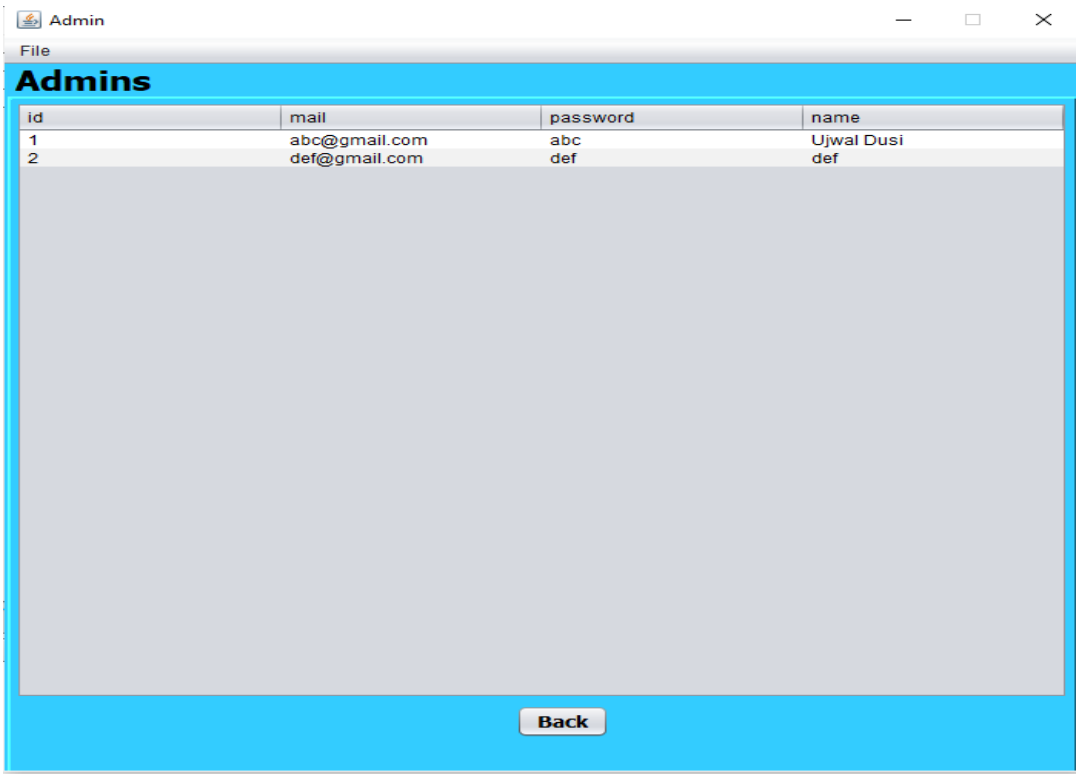
Add a admin



Under Operations

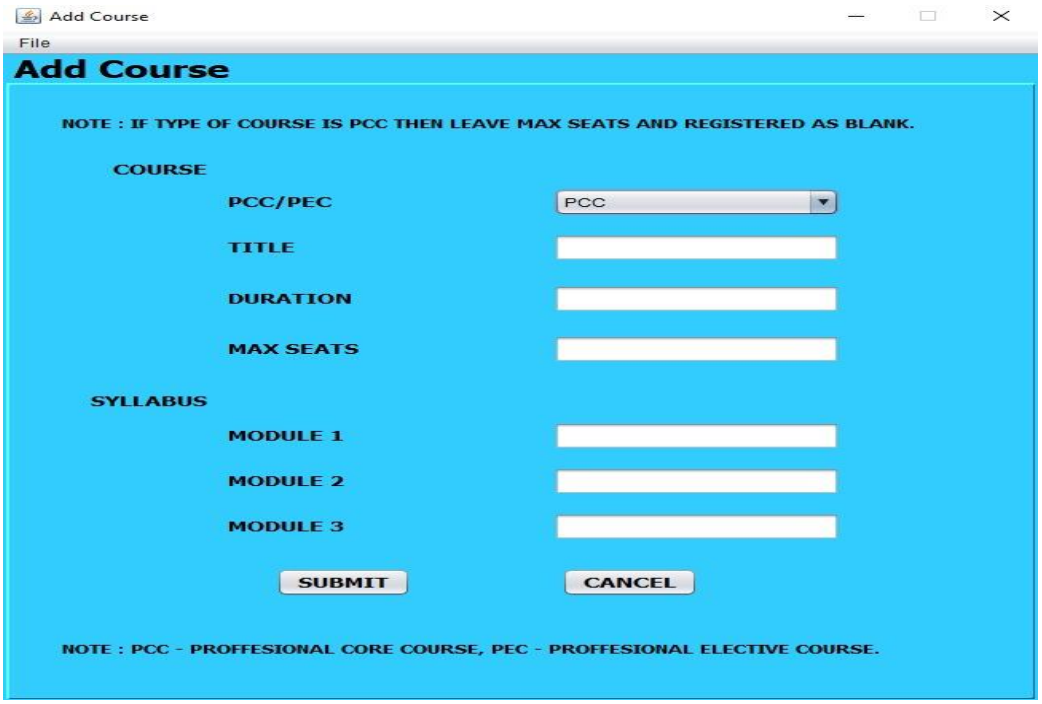


Show Admin list



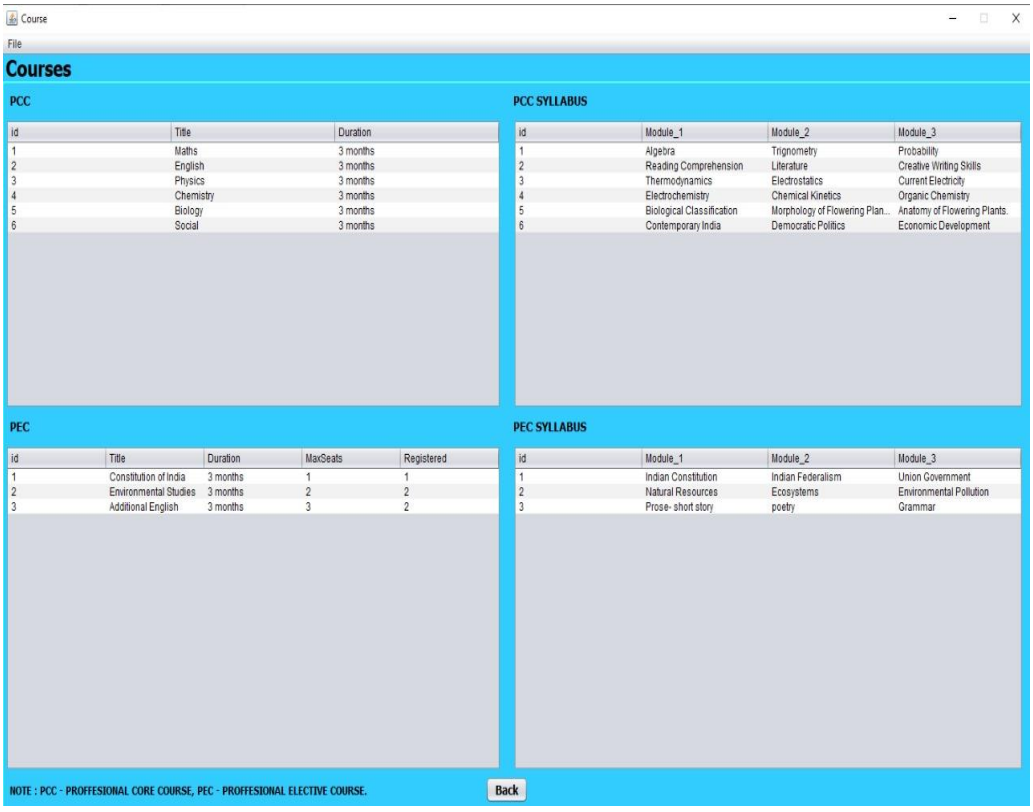
In Course Page

Add course

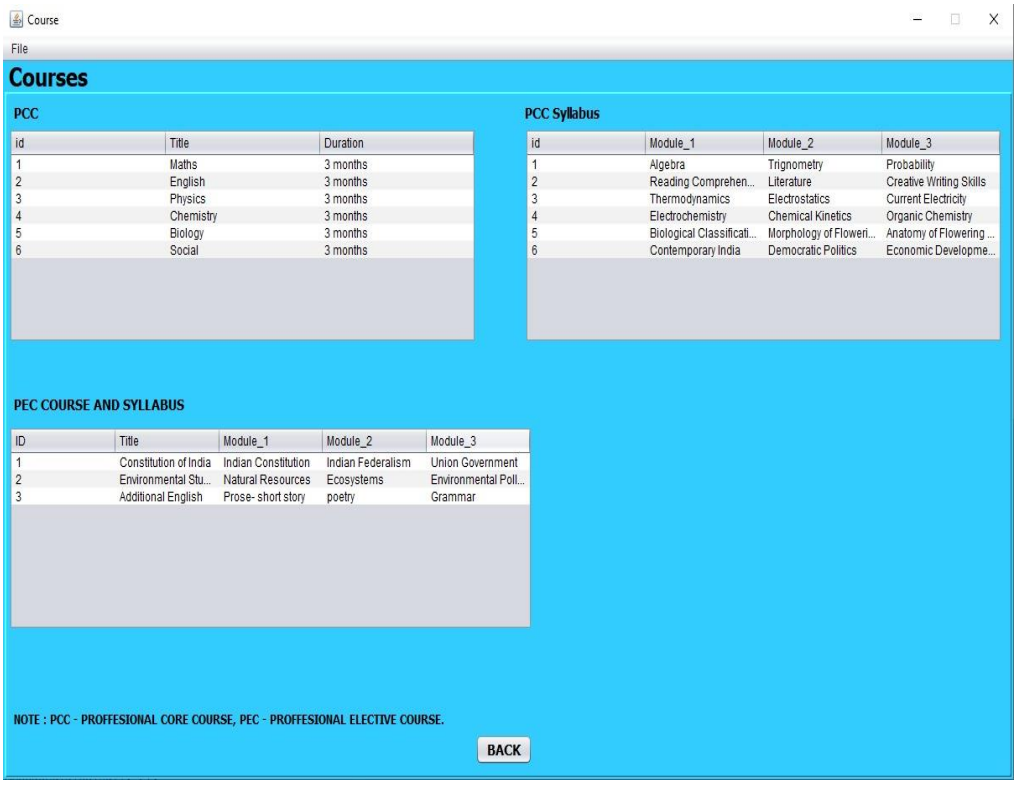


Under Operations

Show Courses



Courses List



Conclusion

A COURSE REGISTRATION APPLICATION FOR THE STUDENTS TO REGISTER FOR ELECTIVE AND OPEN ELECTIVE SUBJECTS IS DESIGNED. IT CONTAINS A PROVISION TO UPLOAD THE SYLLABUS AND CUT-OFF STRENGTH FOR EACH COURSE. THIS PROJECT IS NOW FULLY READY TO BE DEPLOYED.

References

https://youtube.com/playlist?list=PLMa5a9Dh6SlgcnYfJCFsJsrnoHWopDvWa

https://github.com/jkh394/Course-Registration-System