**Introduction:**

**The project we are currently working on is the student database management system. The system is able to add, update, delete and auto enroll students to their corresponding courses depending on their level, term and department.**

**Present System: The present system is able to add, update and delete student and faculty information. It is able to enroll students automatically into courses and after confirming payment is registering them and including them in the class attendance list.**

**The enrollment system is fully automated. Also if we update the term and level, the students will be enrolled into new courses by matching with their term and level.**

**Also, students can not register unless they have completed their payment which is also automated.**

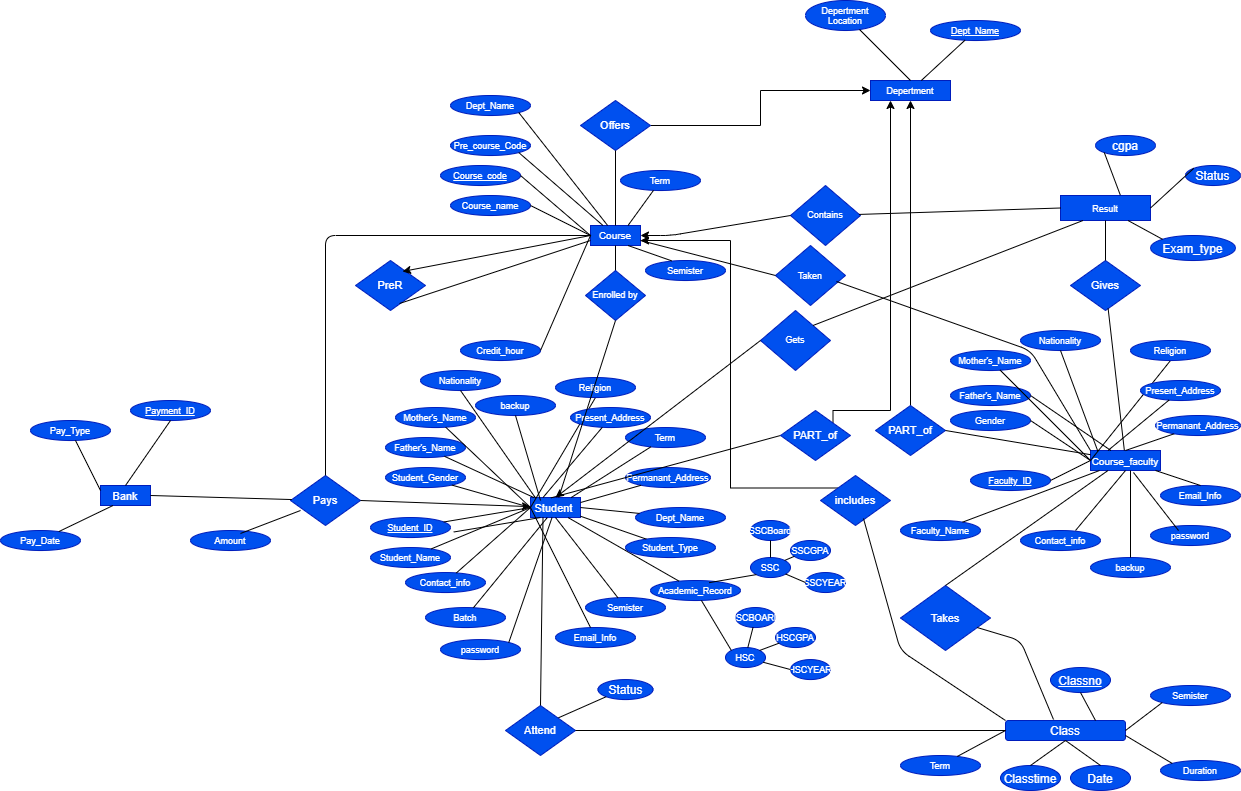
**Problems were mostly faced during creating procedures in oracle. Also some of the queries, mostly join operations were giving undesired values. The use of date variable between java and oracle was also another notable problem as oracle was taking dd-mmm-yyyy format and java was taking yyyy-mm-dd format.**

**The present condition is incomplete. We have yet to include reset password feature, publish result and create class feature for faculty.**

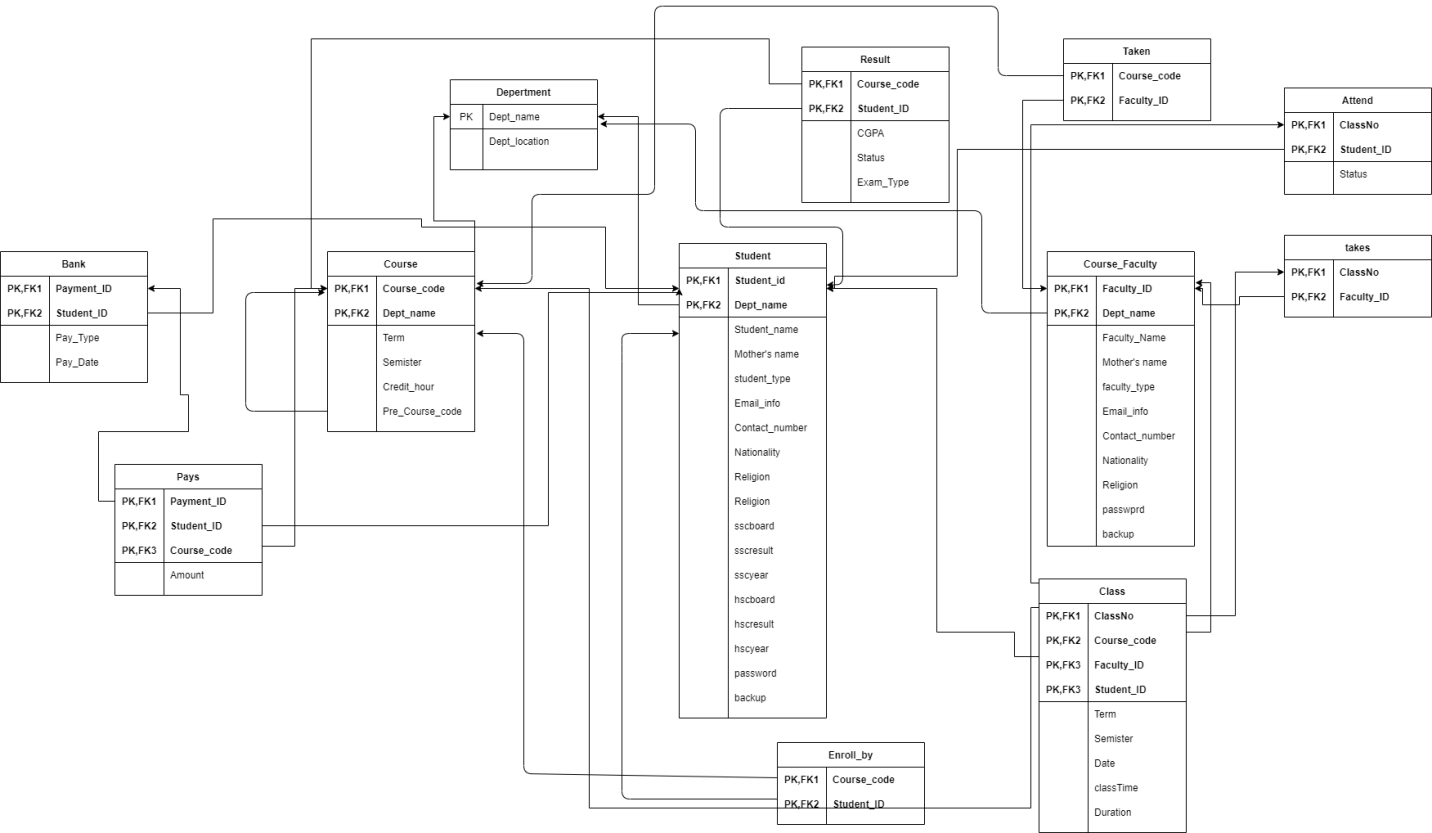
**Proposed System:**

**Conceptual design:**

**E-R Diagram**

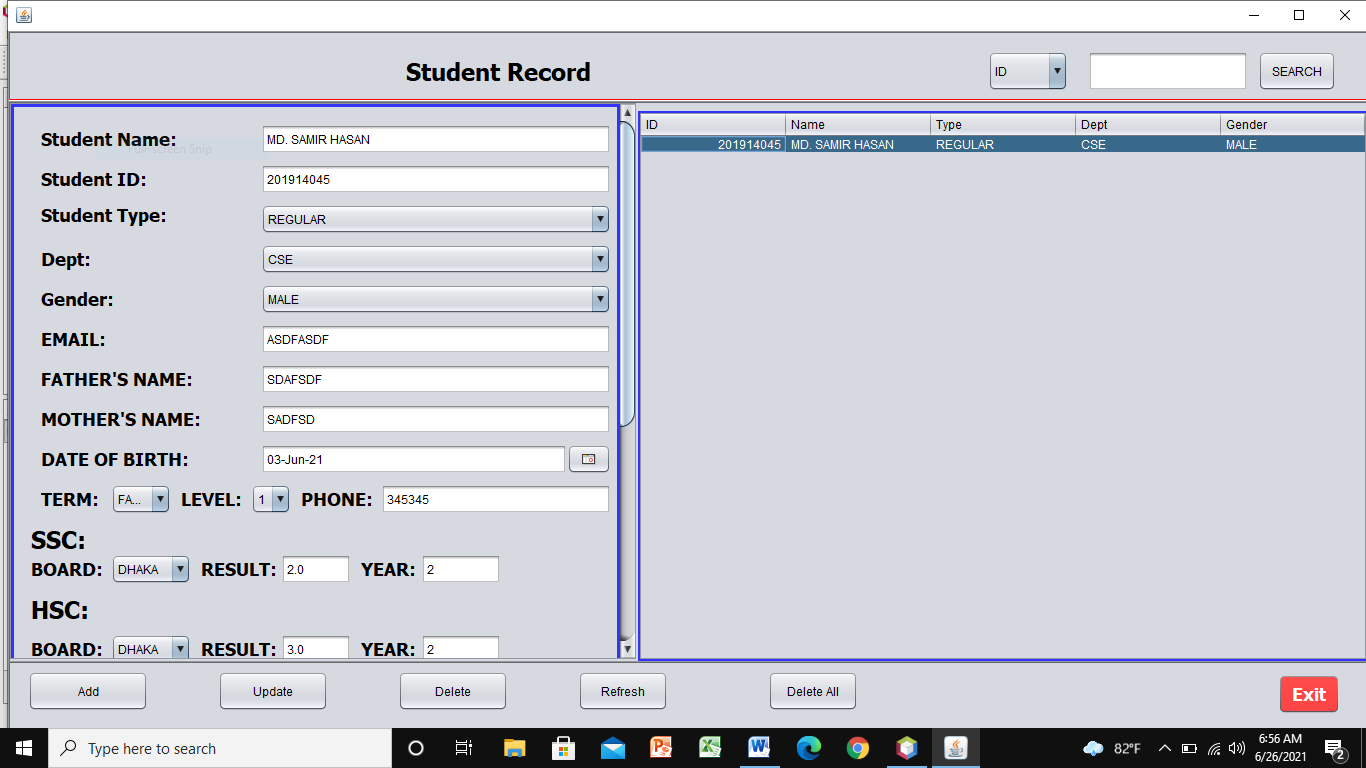
****

**Schema**

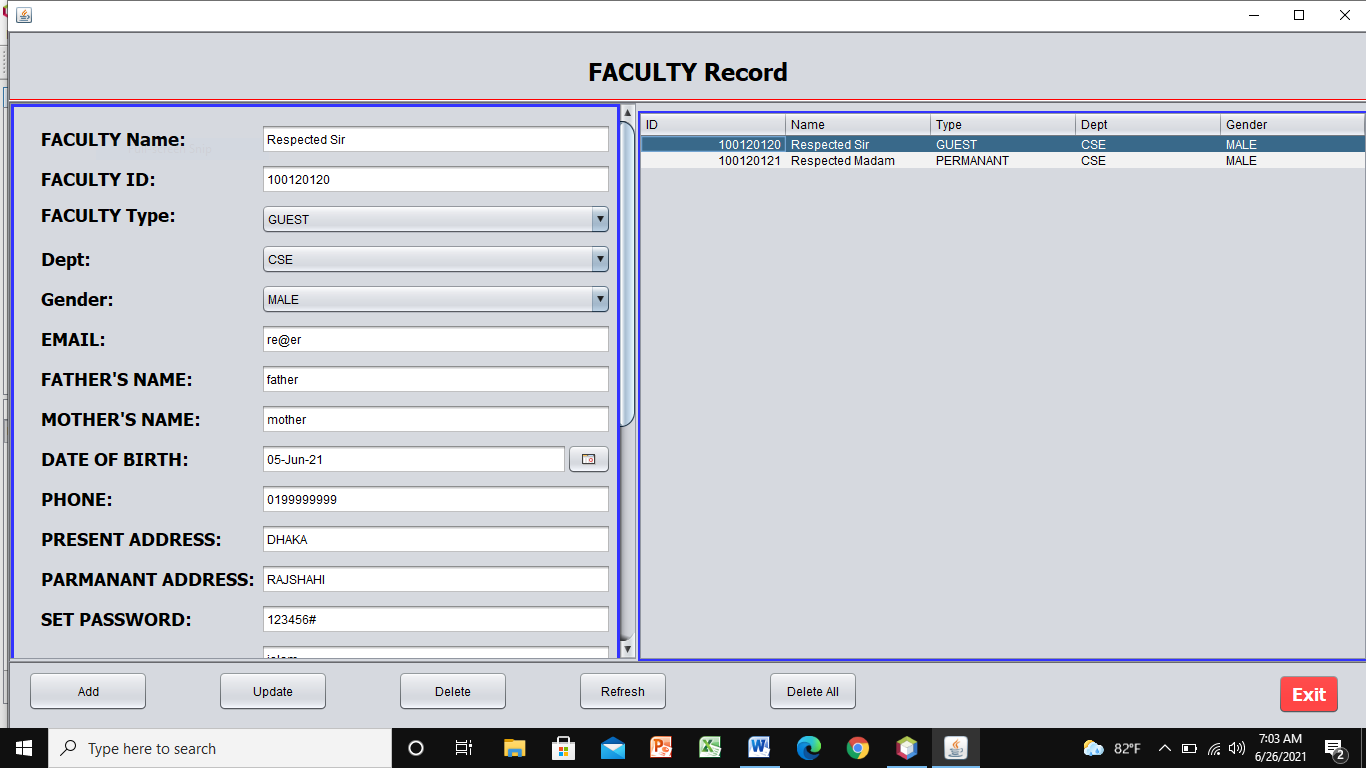
****

**Development of the system:**

**Student Editor:**

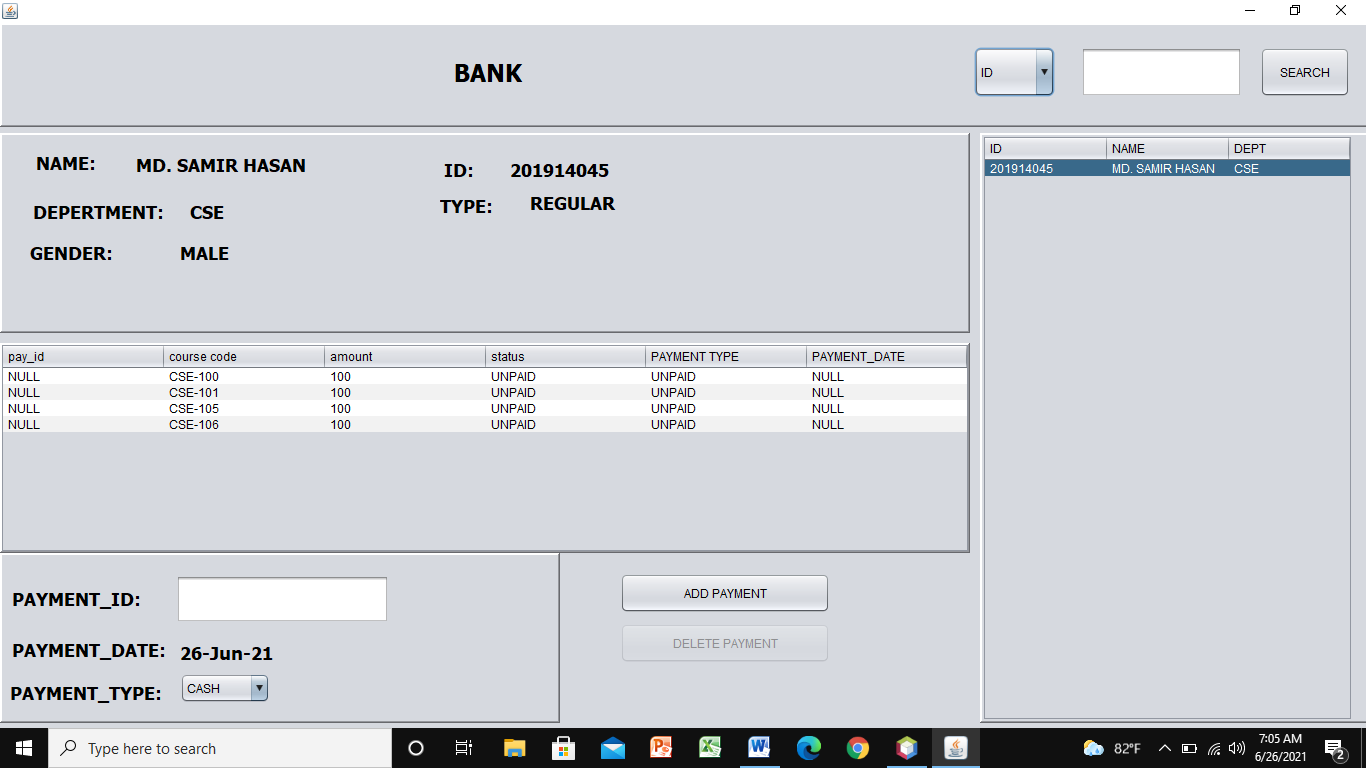
****

**Here, we are able to add, update, delete and search for student information. Adding student automatically enrolls them to courses. Updating term and level enrolls them into new courses.**

**Faculty editor:**

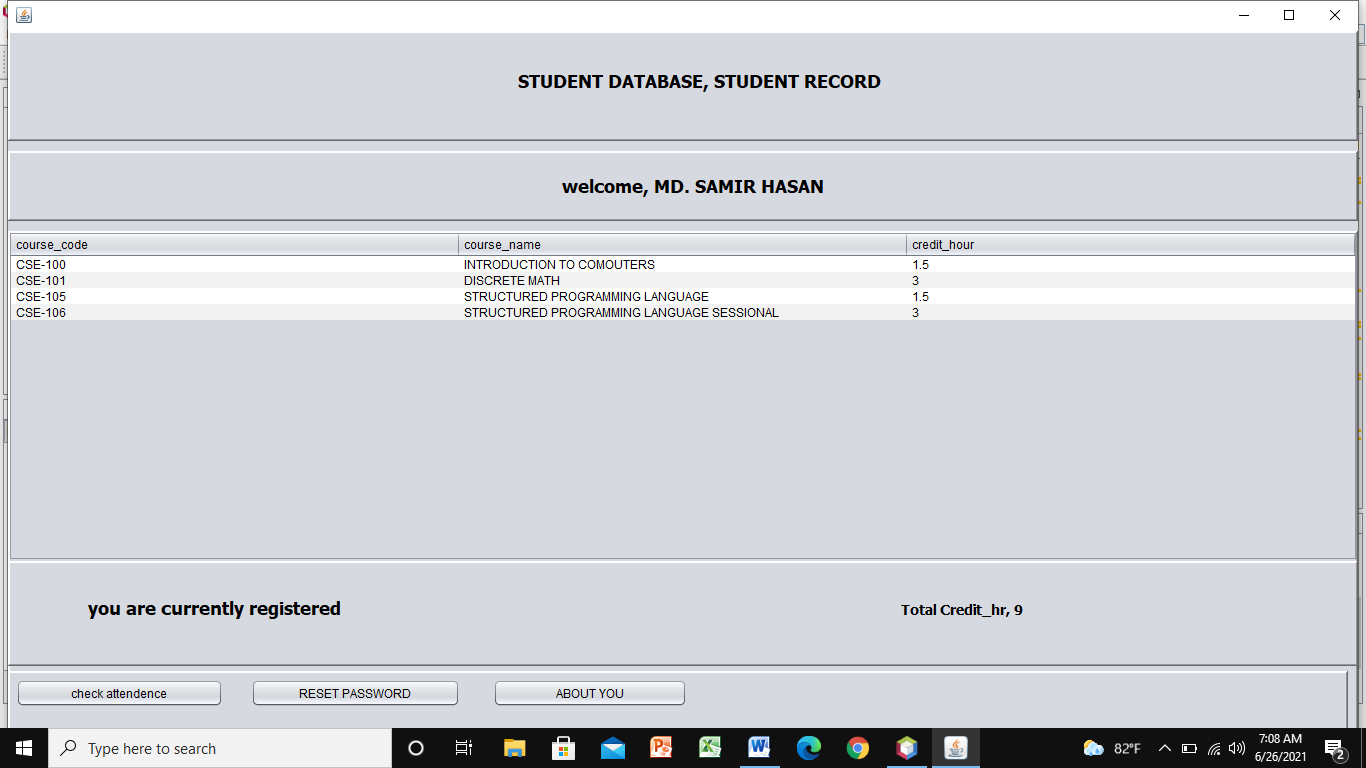
**Just like student editor, we are able to add, search, update or delete any faculty information.**

**Bank:**

****

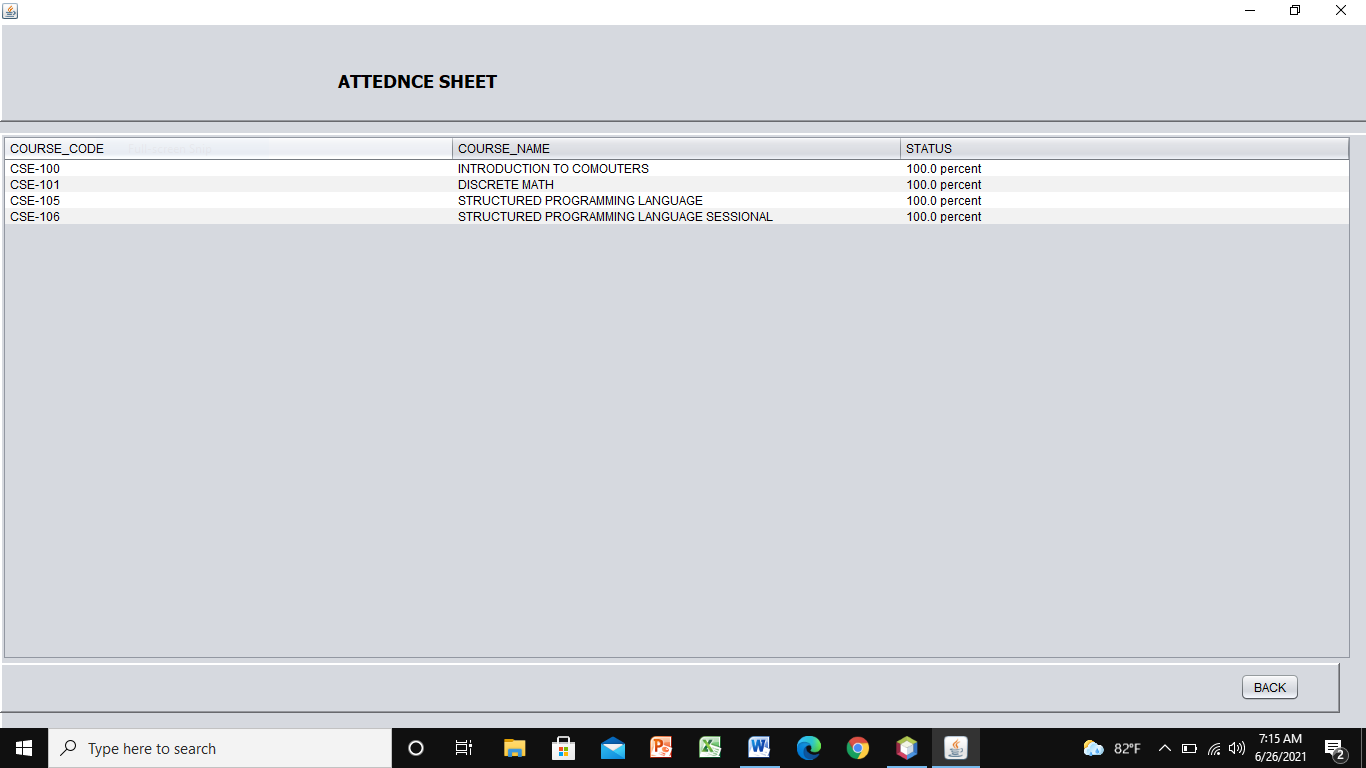
**The bank contains the information of both paid and unpaid student. It is able to show the student the amount he or she has to pay for each course. The payment date is automatically generated. The bank will not ask for payment again if already paid.**

**Student Module:**

****

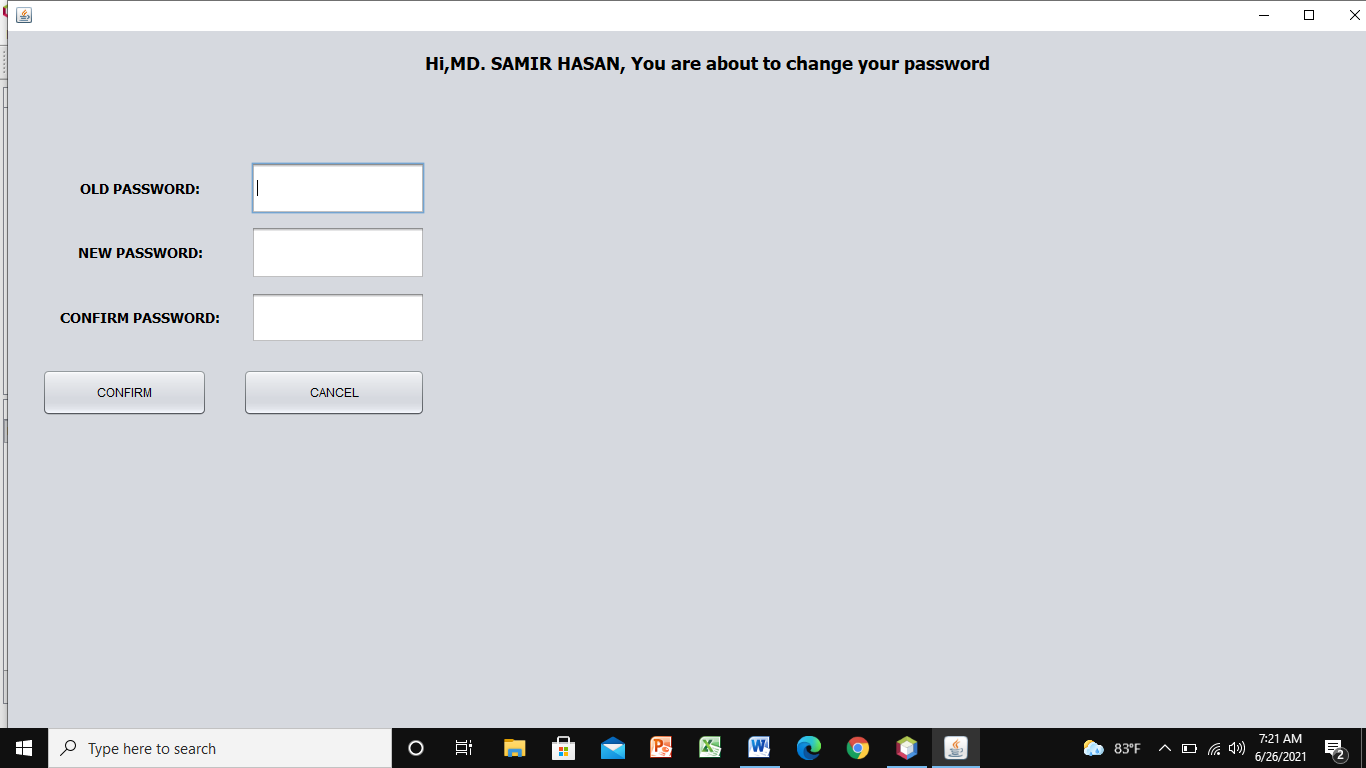
**It is the home page of a student. Here a student can see if he is currently registered, or paid but not registered or unpaid but enrolled. He or she may view their attendance if they are registered, reset their password to their security fitting.**

**Check attendance:**

****

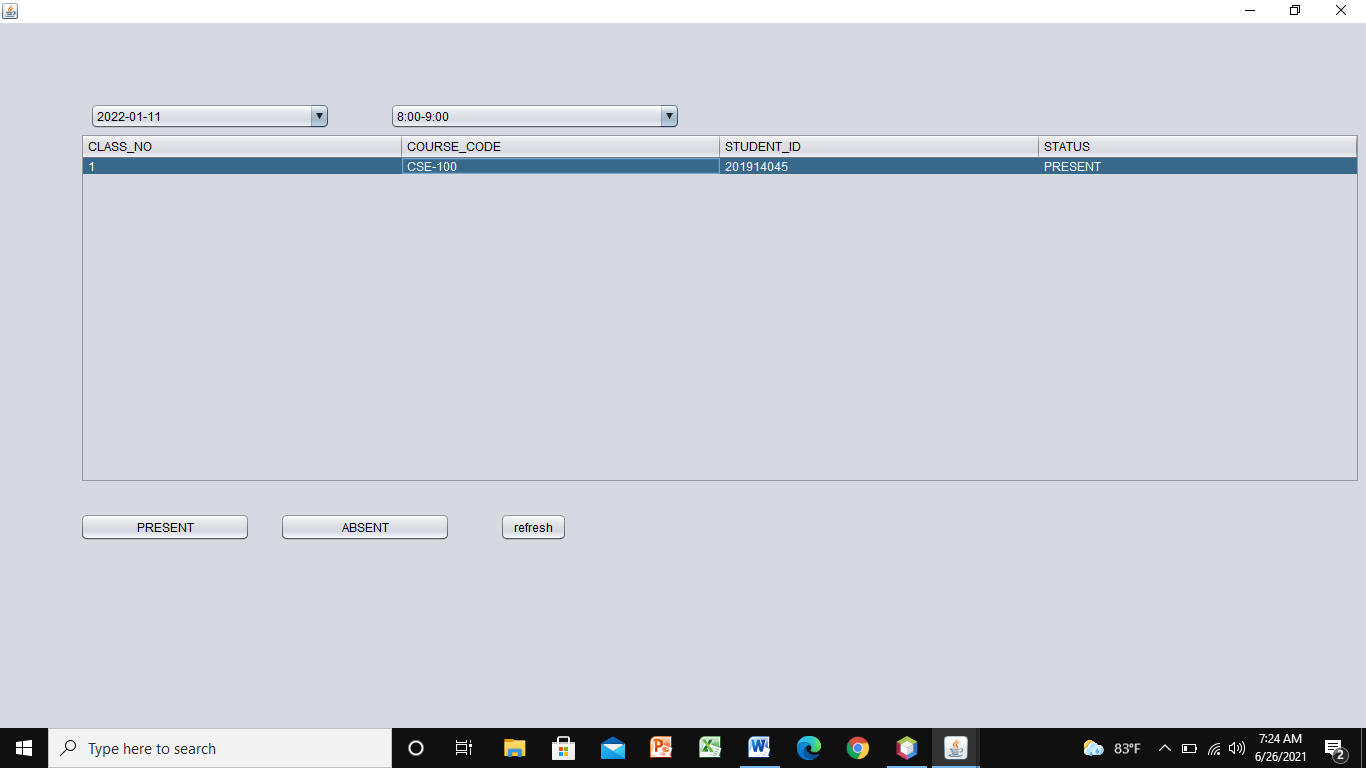
**A registered student may choose to view his attendance status anytime using the attendance sheet.**

**Reset Password:**



**Students can update their password using this feature.**

**Give Attendence:**



**After logging in, faculty can update the attendance of a student.**

**System Requirements:**

|  |  |
| --- | --- |
| **Tools:** | **Netbeans IDE 12.0** |
| **OS:** | **Windows 10** |
| **Front end:** | **Java** |
| **Back End:** | **Oracle 11g** |

**Limitation and future expansion: The system can not yet enroll courses based on pre-requisite courses. It can not yet publish any results. We shall be working on these features in near future.**

**Conclusion: The database system we have designed is still at its infant stage. We shall be able to improve its features more, given time.**