AHMEDABAD UNIVERSITY

CSC102 Object Oriented Programming

Auto Attendance System

Say ‘NO’ to Proxy!

Team

Roll No.	Name	Programe
AU1841076	Rahul Chocha	BTECH ICT
AU1841109	Jeet Karia	BTECH ICT

Problem Definition

- This project mainly is designed to open new and creative and also more secured ways of taking attendance.
- It is also helpful in not only taking attendance of students but also of industrial employees or any other organization.
- World is getting revolutionized at an unimaginable pace using cutting edge AI technology and let us all be a part of it.

Objective and End Users

- It will take attendance in dynamic manner which may sound like science fiction and very efficiently.
- It is making attendance of students by recognizing their faces and it even has got attendance management thing in it.
- This software has got three End Users which includes student, faculty (one taking attendance), management(they, who store databases and training images of students so that they get recognized when used by faculty).

Module-wise functionality/features

- We have three modules in our software which are :
- **STUDENT:**
- This module has three features :
- 1)Show Sub. wise Attendance
- It will show attendance of logged in student subject wise using bar graph in very systematic manner.
- 2) Show Total Attendance
- It will show overall attendance of logged in student till date in percentage using pie chart.

- 3) Take Courses

- This feature will allow students to take courses according to their semester they are studying at that moment in very user friendly way.
- **FACULTY:**
- This module has got Two features:
- 1)Start Attendance:
- This feature will allow logged in faculty to select their respective course ID and lecture number and then attendance will start according to the faces recognized.

- 2) Manual Attendance:
  - This feature is there if any student has come and his face is not recognized properly then he can contact respective faculty member and he can put his/her attendance manually and that particular student will be cross verified if he is speaking truth by checking the database of images which stores every image it detects.

- **MANAGEMENT**

- This module has even got three features
- 1) Add Student:
- This feature allows logged in management member to add student details and store his face and hence training them in order so that they can be recognized by faculty during attendance.
- 2) Update Student:
- This feature will allow students details if they have got their address changed or any detail.

- 3) Remove Student:
  - This feature has got ability to delete any student's data from the university's database.

Algorithm Used For Face Training And Recognizing

- We have used LBPH Face Algorithm for face prediction.
- LBPH Algorithm:
- In earlier times it was known as just LBP (Local Binary Pattern) and is getting used since 1994 for just texture recognition. But afterwards people found that if histograms are connected to it then this can be more accurate.

Steps in LBPH Algorithm:

- **Parameters**: the LBPH uses 4 parameters:
  - **Radius**: the radius is used to build the circular local binary pattern and represents the radius around the central pixel. It is usually set to 1.
  - **Neighbors**: the number of sample points to build the circular local binary pattern. Keep in mind: the more sample points you include, the higher the computational cost. It is usually set to 8.
  - **Grid X**: the number of cells in the horizontal direction. The more cells, the finer the grid, the higher the dimensionality of the resulting feature vector. It is usually set to 8.
  - **Grid Y**: the number of cells in the vertical direction. The more cells, the finer the grid, the higher the dimensionality of the resulting feature vector. It is usually set to 8.

- **Step II:**

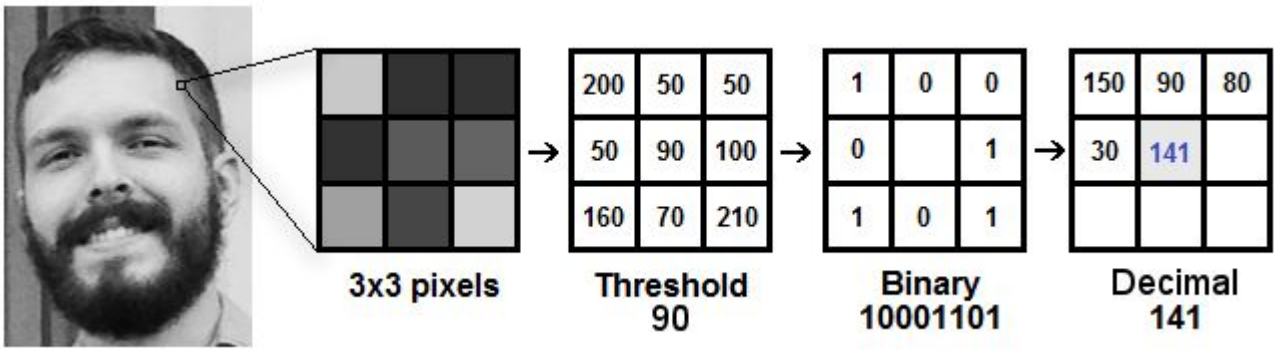
Training the Algorithm:

First, we need to train the algorithm. To do so, we need to use a dataset with the facial images of the people we want to recognize. We need to also set an ID (it may be a number or the name of the person) for each image, so the algorithm will use this information to recognize an input image and give you an output. Images of the same person must have the same ID. With the training set already constructed, let's see the LBPH computational steps.

• Step III:

Applying the LBP operation: The first computational step of the LBPH is to create an intermediate image that describes the original image in a better way, by highlighting the facial characteristics. To do so, the algorithm uses a concept of a sliding window, based on the parameters **radius** and **neighbors**.

The image below shows this procedure:



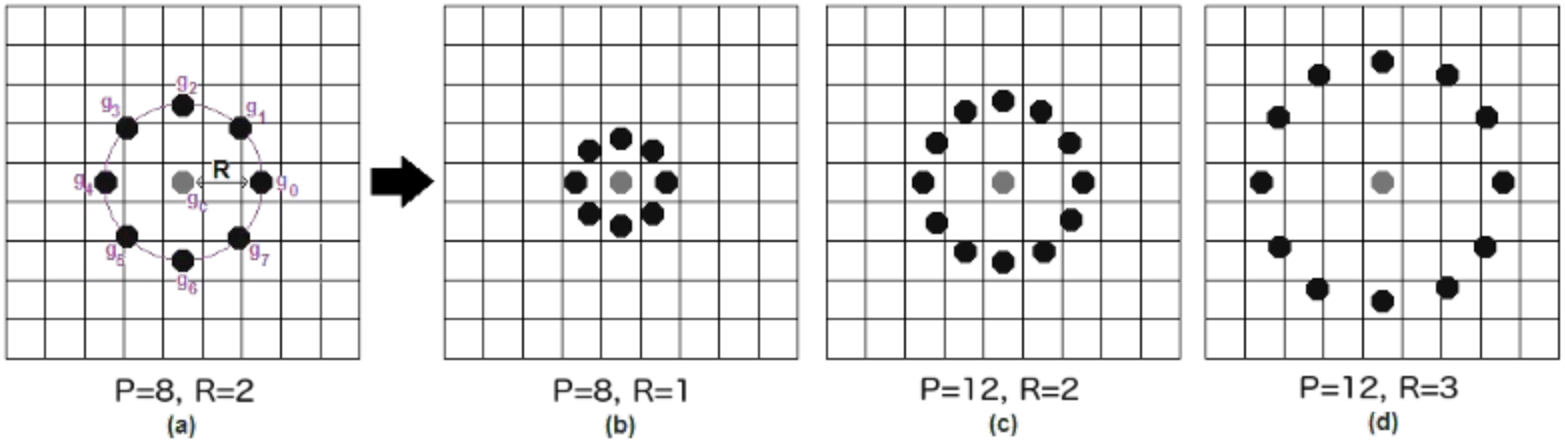
Based on the image in previous slide, let's break it into several small steps so we can understand it easily:

- Suppose we have a facial image in grayscale.
- We can get part of this image as a window of 3x3 pixels.
- It can also be represented as a 3x3 matrix containing the intensity of each pixel (0~255).
- Then, we need to take the central value of the matrix to be used as the threshold.
- This value will be used to define the new values from the 8 neighbors.

- For each neighbor of the central value (threshold), we set a new binary value. We set 1 for values equal or higher than the threshold and 0 for values lower than the threshold.
- Now, the matrix will contain only binary values (ignoring the central value). We need to concatenate each binary value from each position from the matrix line by line into a new binary value (e.g. 10001101). Note: some authors use other approaches to concatenate the binary values (e.g. clockwise direction), but the final result will be the same.

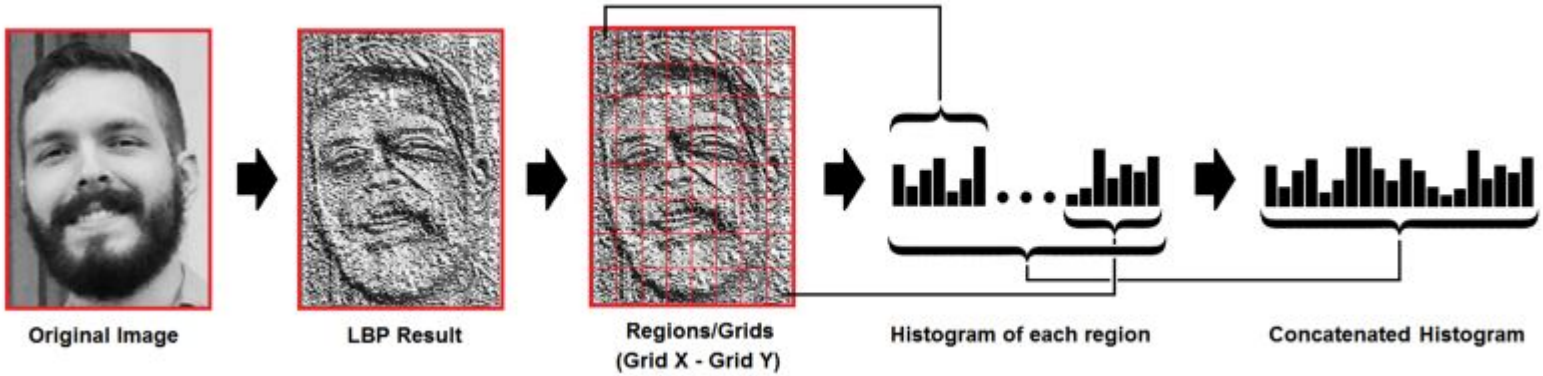
- Then, we convert this binary value to a decimal value and set it to the central value of the matrix, which is actually a pixel from the original image.

- At the end of this procedure (LBP procedure), we have a new image which represents better the characteristics of the original image.



• Step IV:

Extracting the Histograms: Now, using the image generated in the last step, we can use the **Grid X** and **Grid Y** parameters to divide the image into multiple grids, as can be seen in the following image:



Based on the image in the previous slide, we can extract the histogram of each region as follows:

As we have an image in grayscale, each histogram (from each grid) will contain only 256 positions (0~255) representing the occurrences of each pixel intensity.

Then, we need to concatenate each histogram to create a new and bigger histogram. Supposing we have 8x8 grids, we will have 8x8x256=16,384 positions in the final histogram. The final histogram represents the characteristics of the image original image.

Step V: (Performing Face Recognition)

In this step, the algorithm is already trained. Each histogram created is used to represent each image from the training dataset. So, given an input image, we perform the steps again for this new image and creates a histogram which represents the image.

- So to find the image that matches the input image we just need to compare two histograms and return the image with the closest histogram.

- We can use various approaches to compare the histograms (calculate the distance between two histograms), for example: **euclidean distance**, **chi-square**, **absolute value**, etc. In this example, we can use the Euclidean distance (which is quite known) based on the following formula:

$$D = \sqrt{\sum_{i=1}^{n}(hist1_i - hist2_i)^2}$$

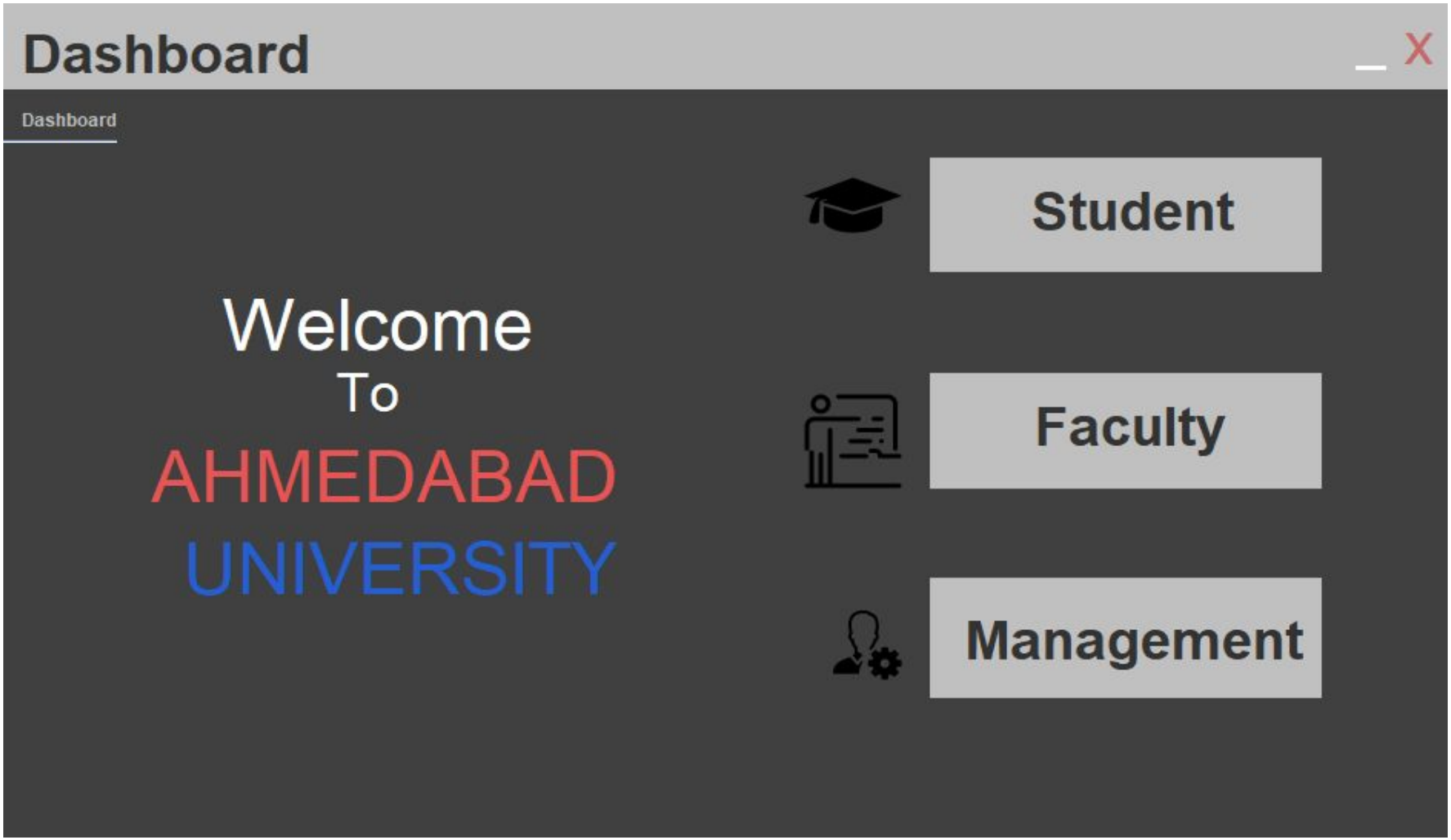
- So the algorithm output is the ID from the image with the closest histogram. The algorithm should also return the calculated distance, which can be used as a '**confidence**' measurement.

  Note: don't be fooled about the 'confidence' name, as lower confidences are better because it means the distance between the two histograms is closer.
- We can then use a threshold and the 'confidence' to automatically estimate if the algorithm has correctly recognized the image. We can assume that the algorithm has successfully recognized if the confidence is lower than the threshold defined.

User Interface Design/ Results /System Outputs

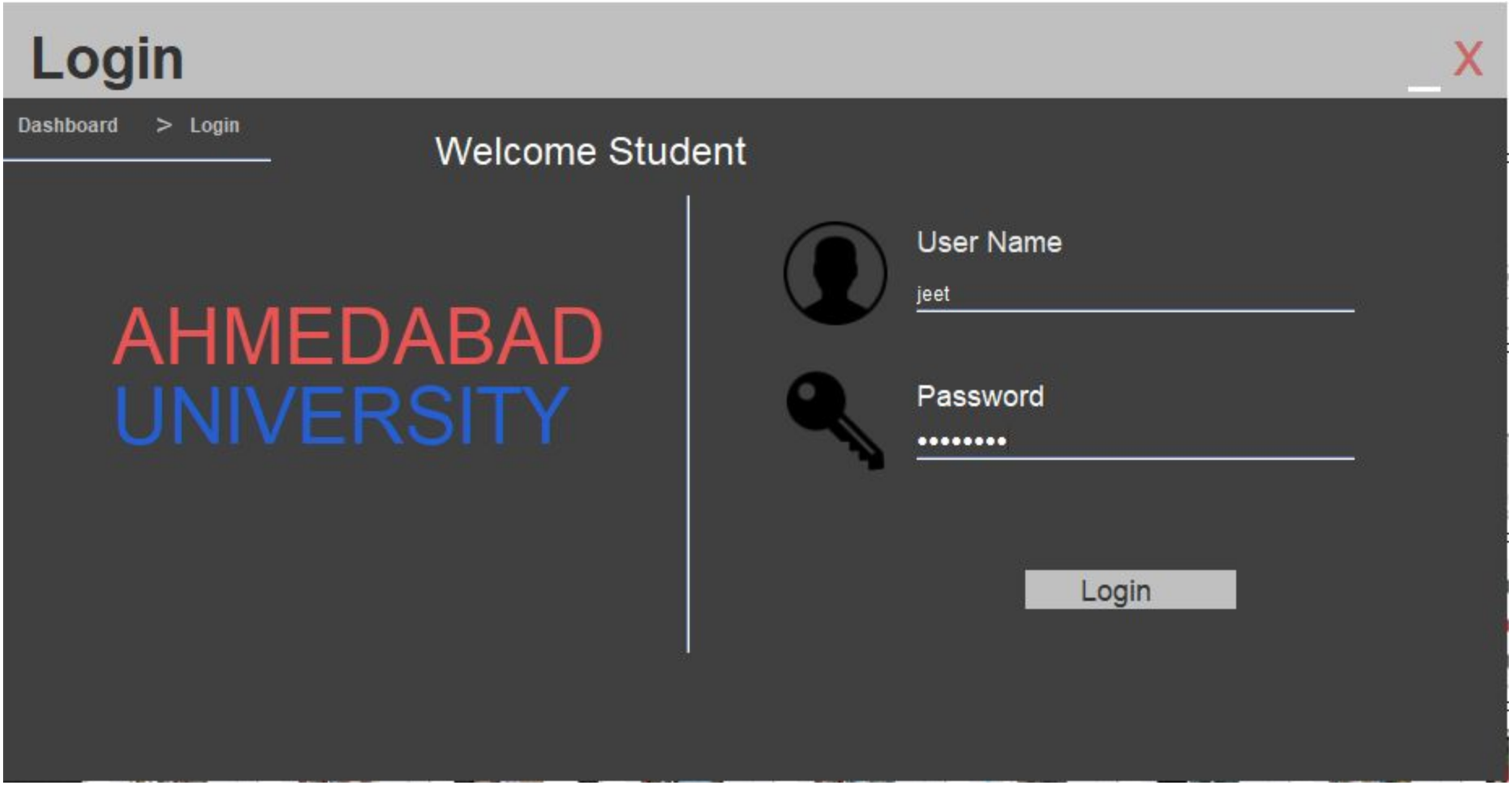
User Interface Design / Screenshots

Screen-2 - DashBoard



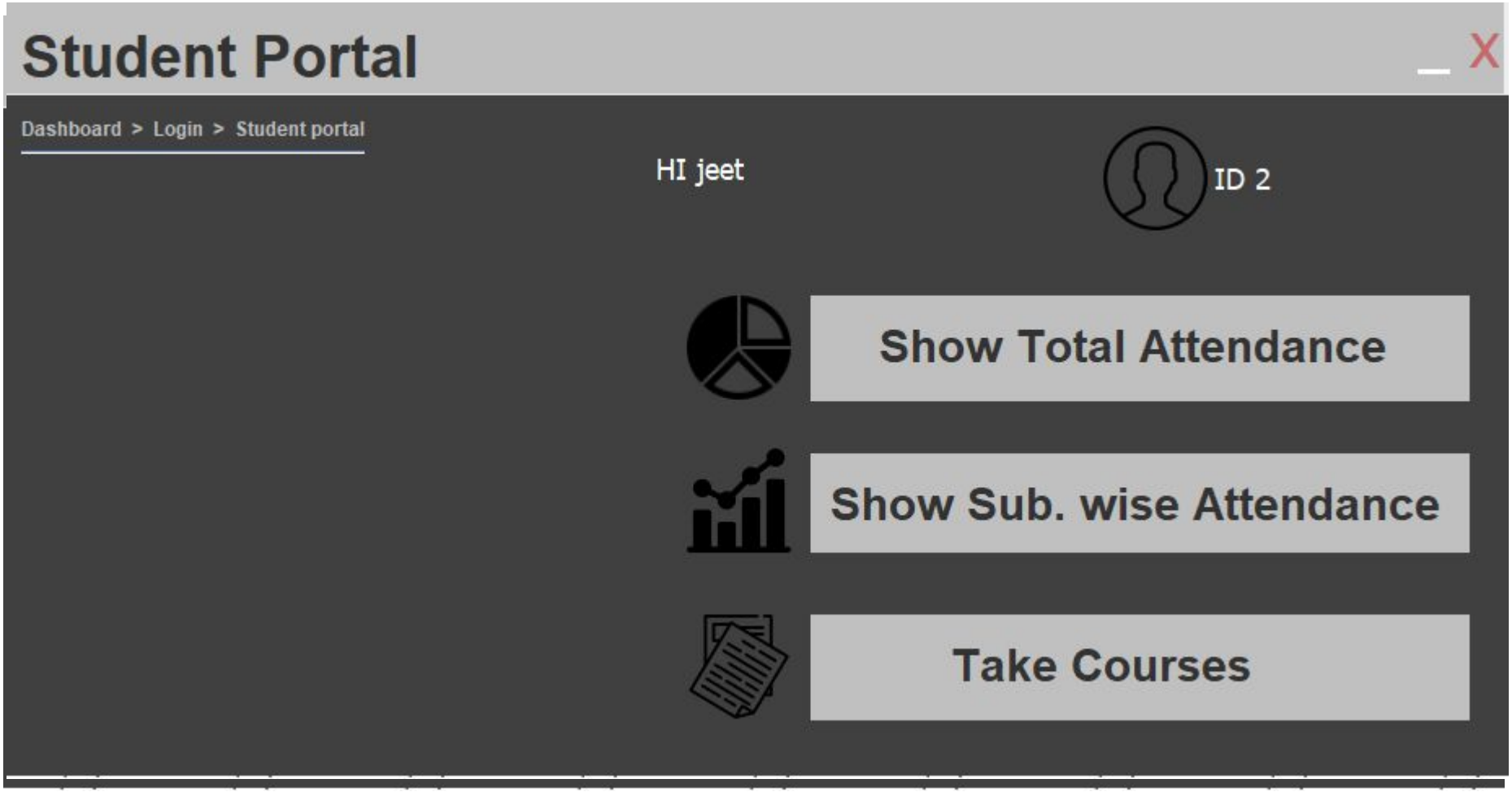
User Interface Design / Screenshots

Screen-3 – Student Login



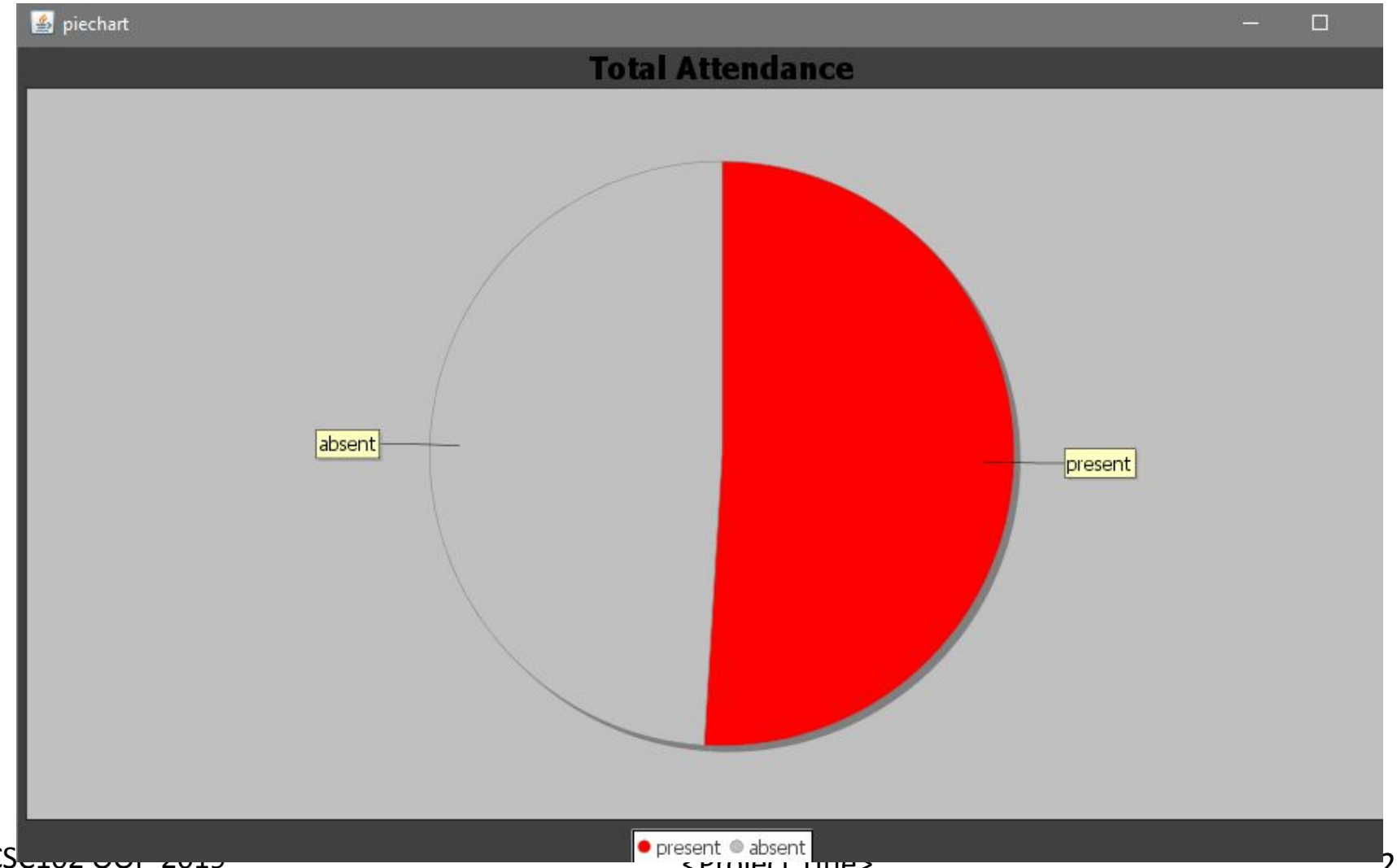
User Interface Design / Screenshots

Screen-4– Student Portal



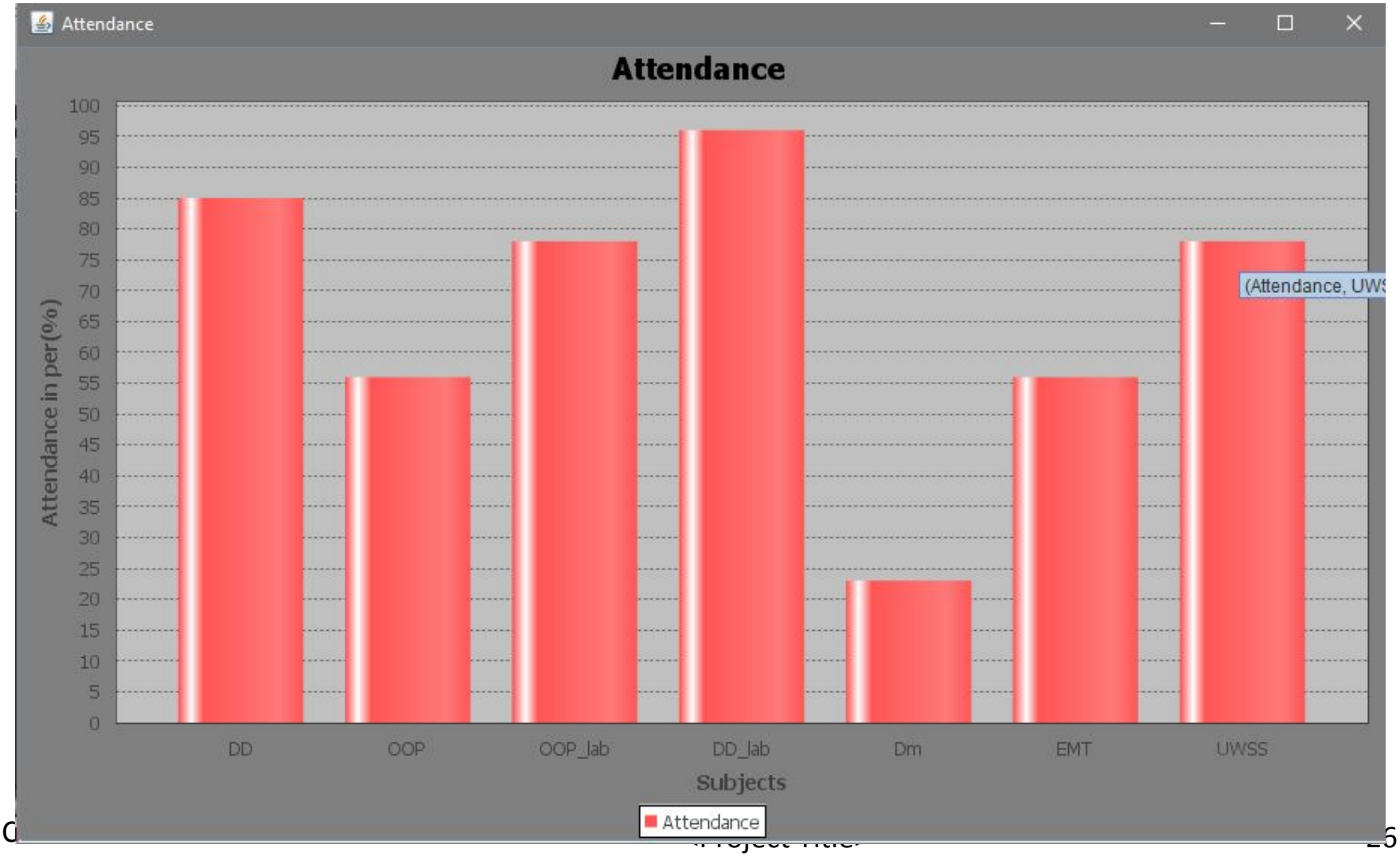
User Interface Design / Screenshots

Screen-5-Show Total Attendance



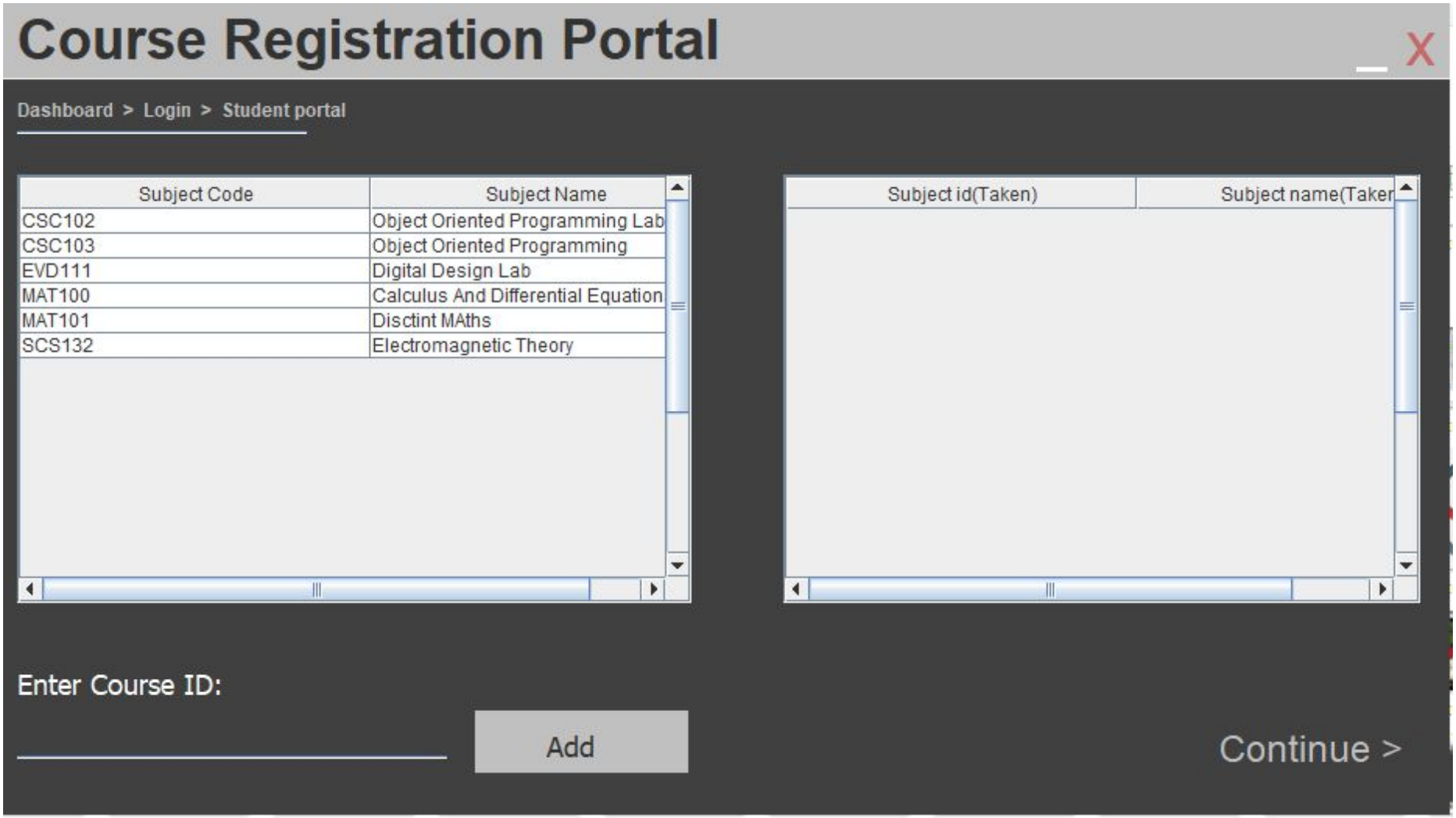
User Interface Design / Screenshots

Screen-6 – Subject Wise Attendance



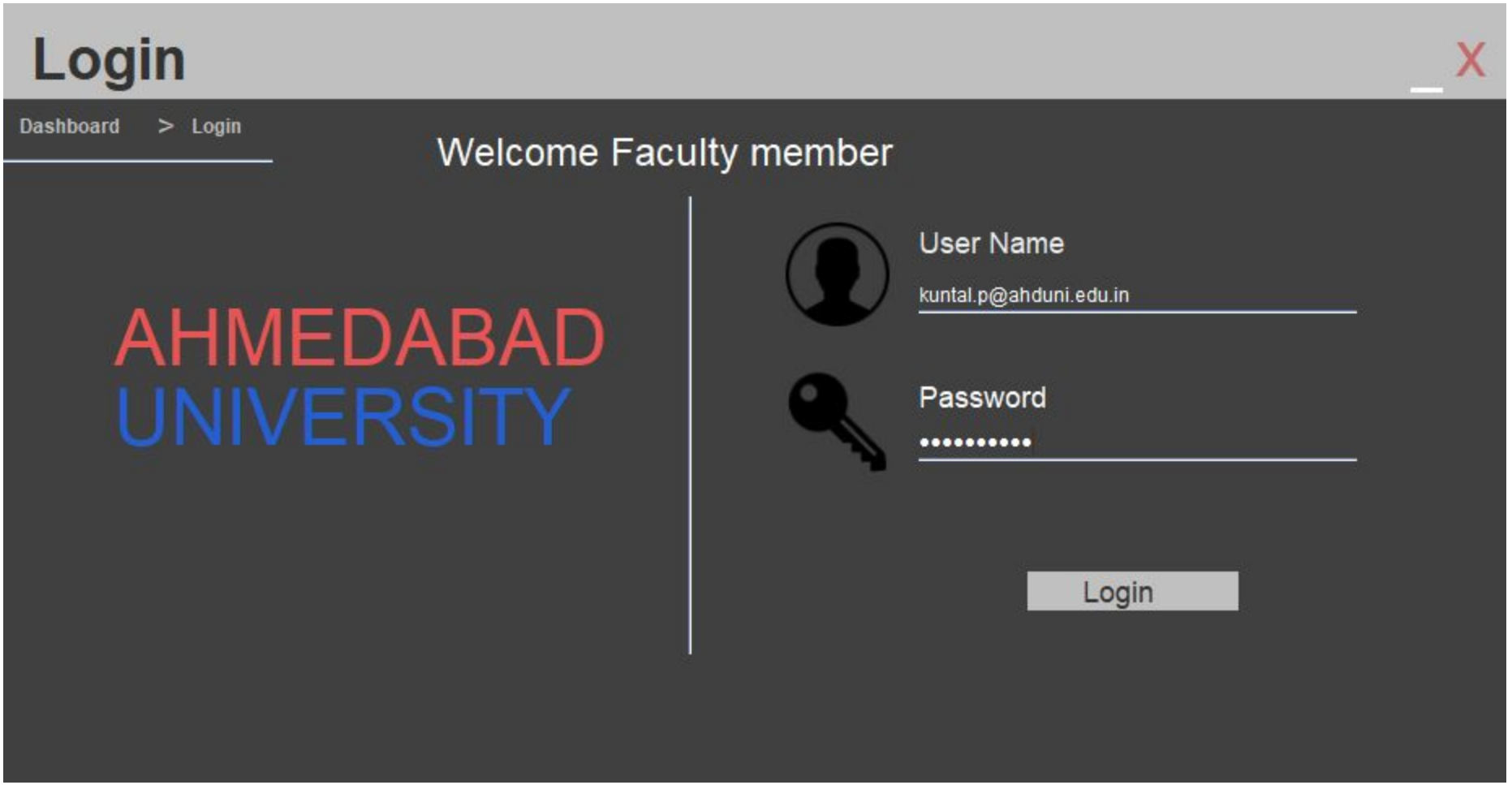
User Interface Design / Screenshots

Screen-7 – Take Courses



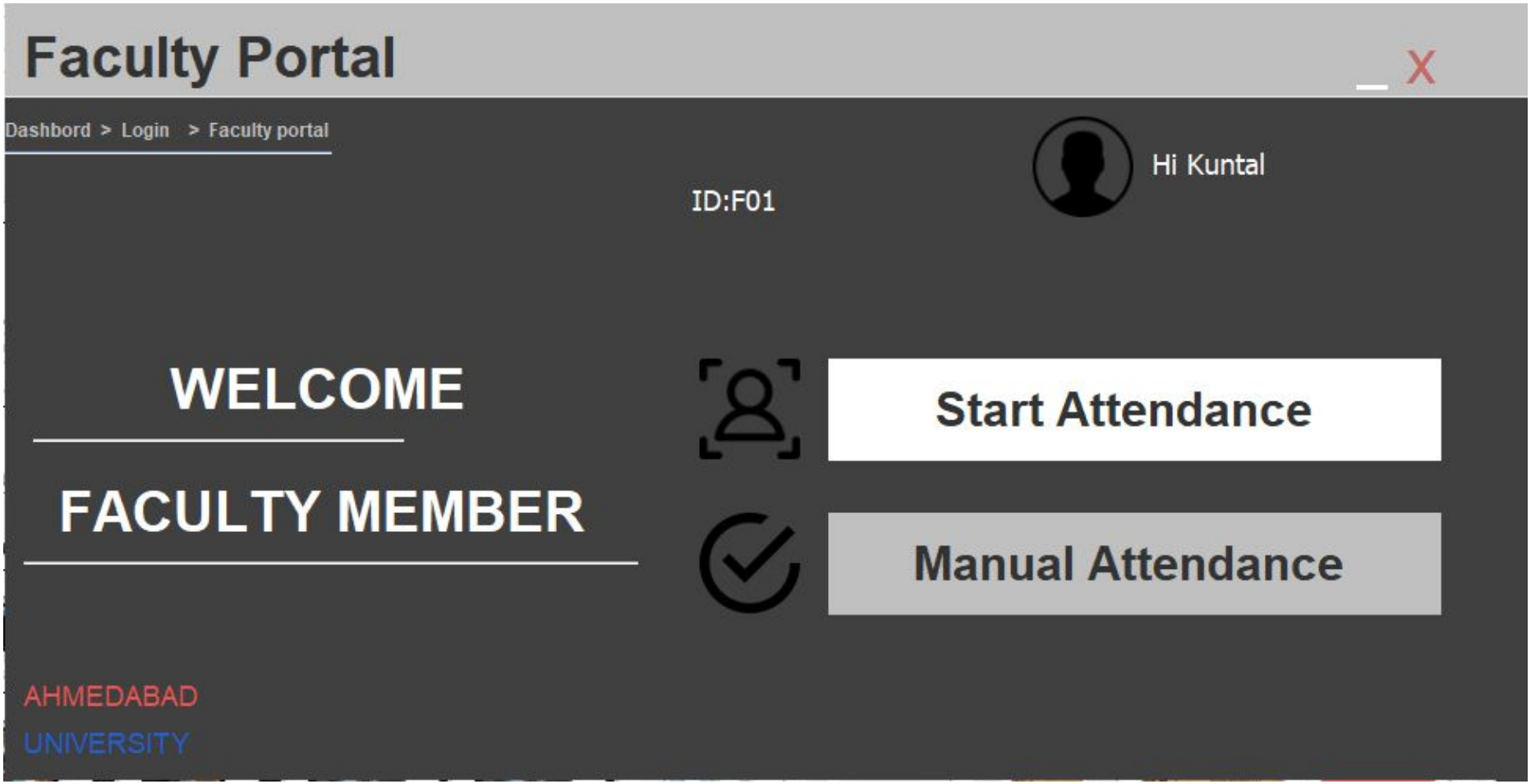
User Interface Design / Screenshots

Screen-8-Faculty Login



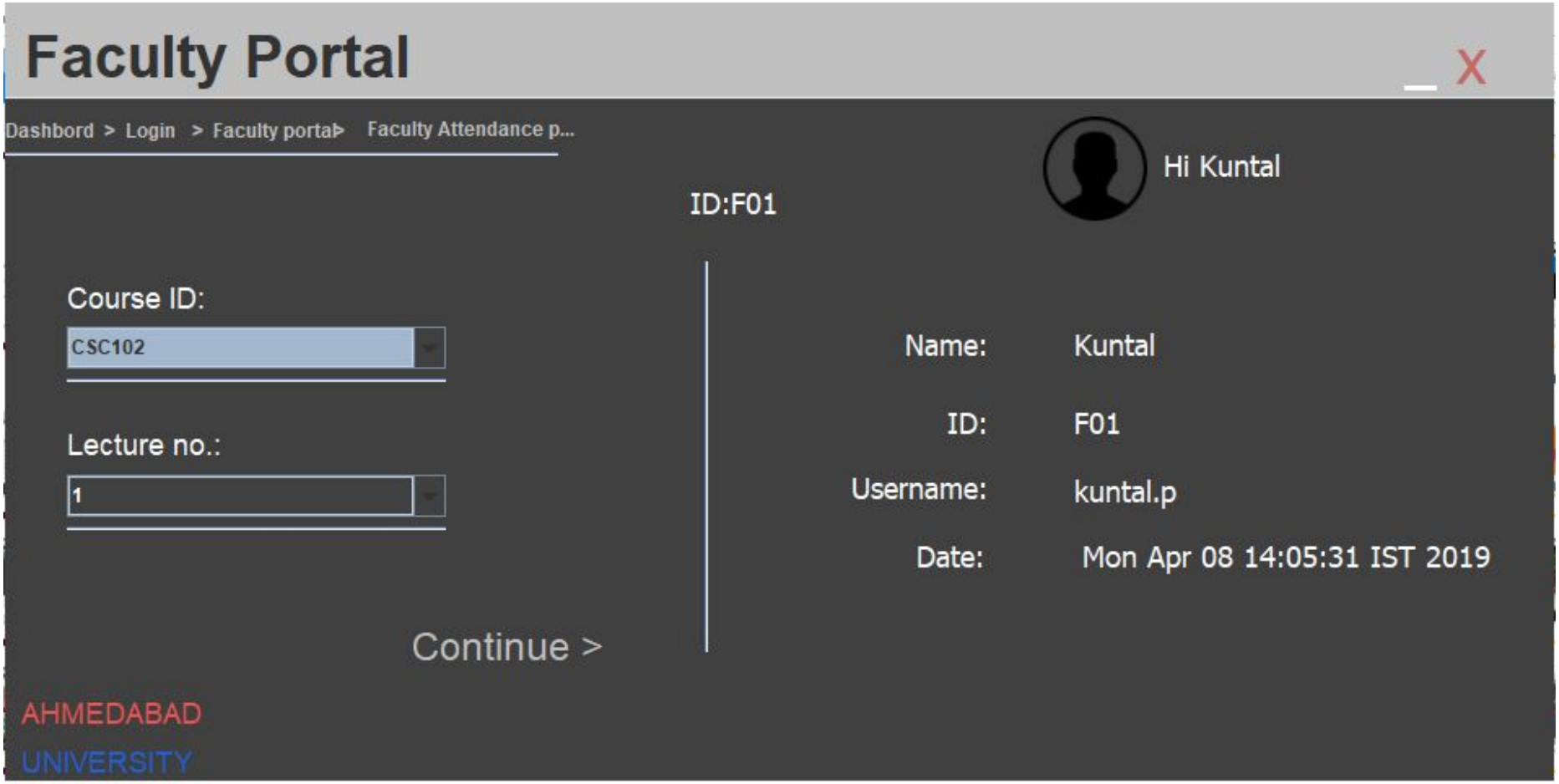
User Interface Design / Screenshots

Screen-9-Faculty Portal



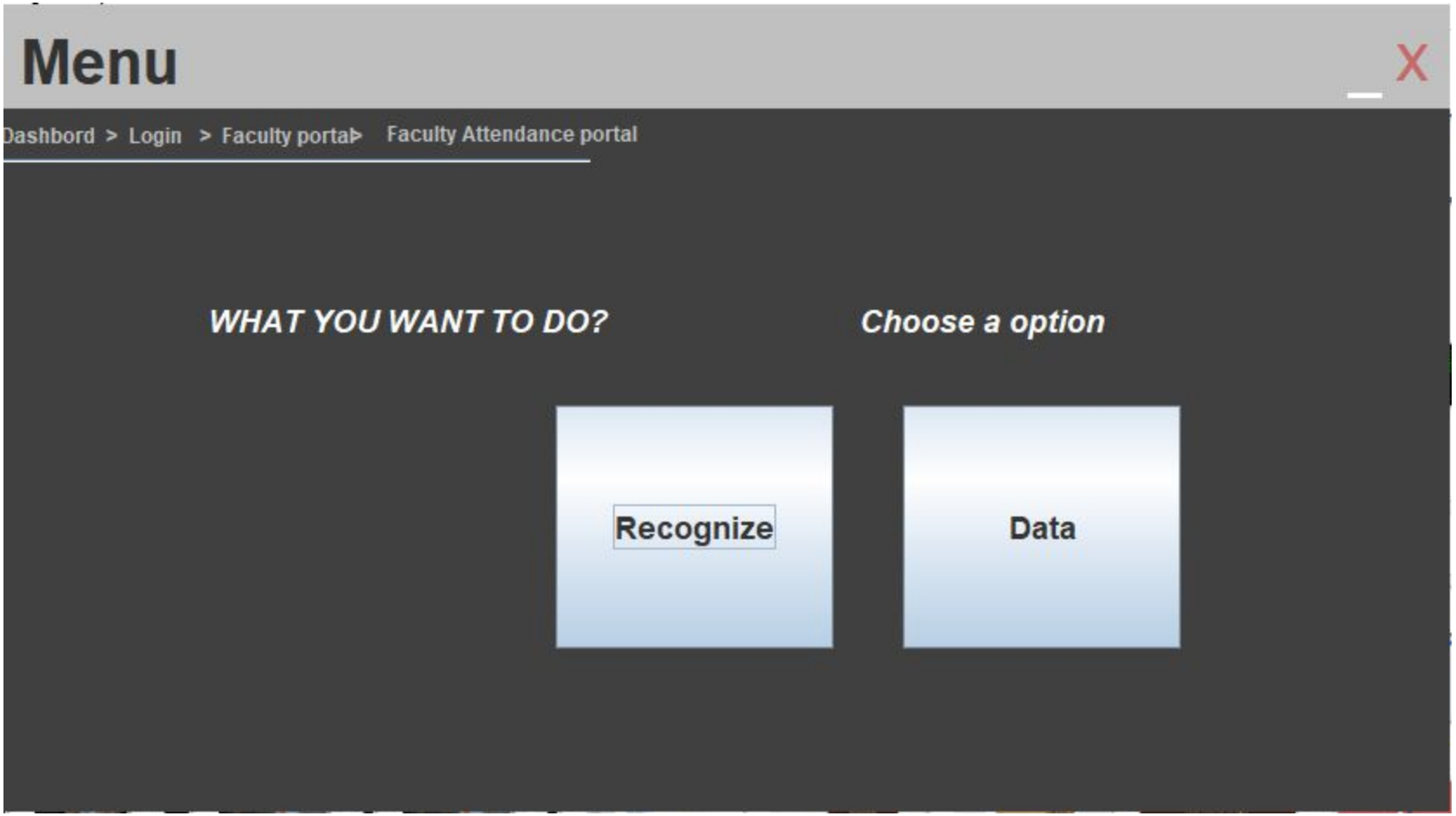
User Interface Design / Screenshots

Screen-9-Take Attendance



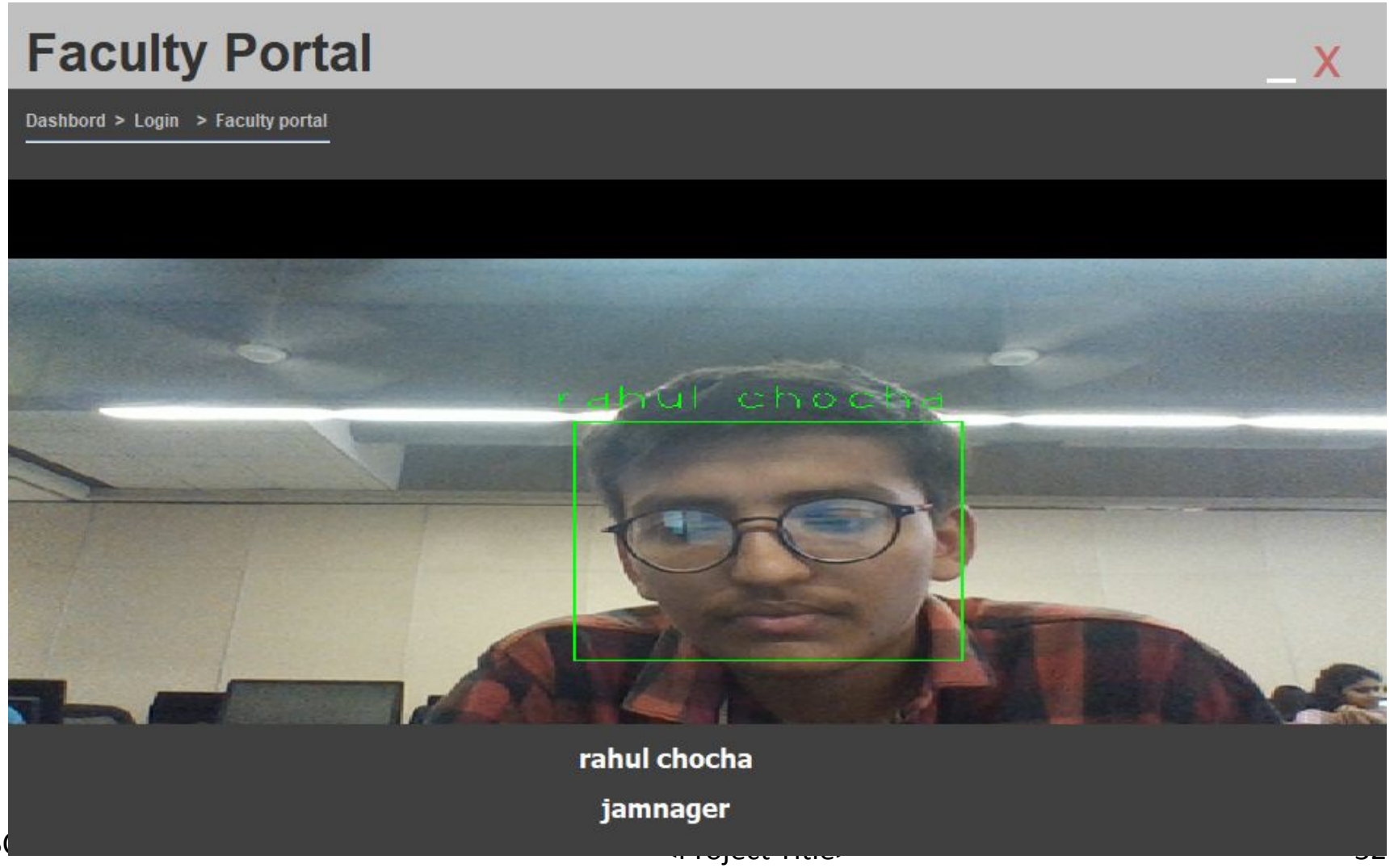
User Interface Design / Screenshots

Screen-9-Faculty Portal(Select option)



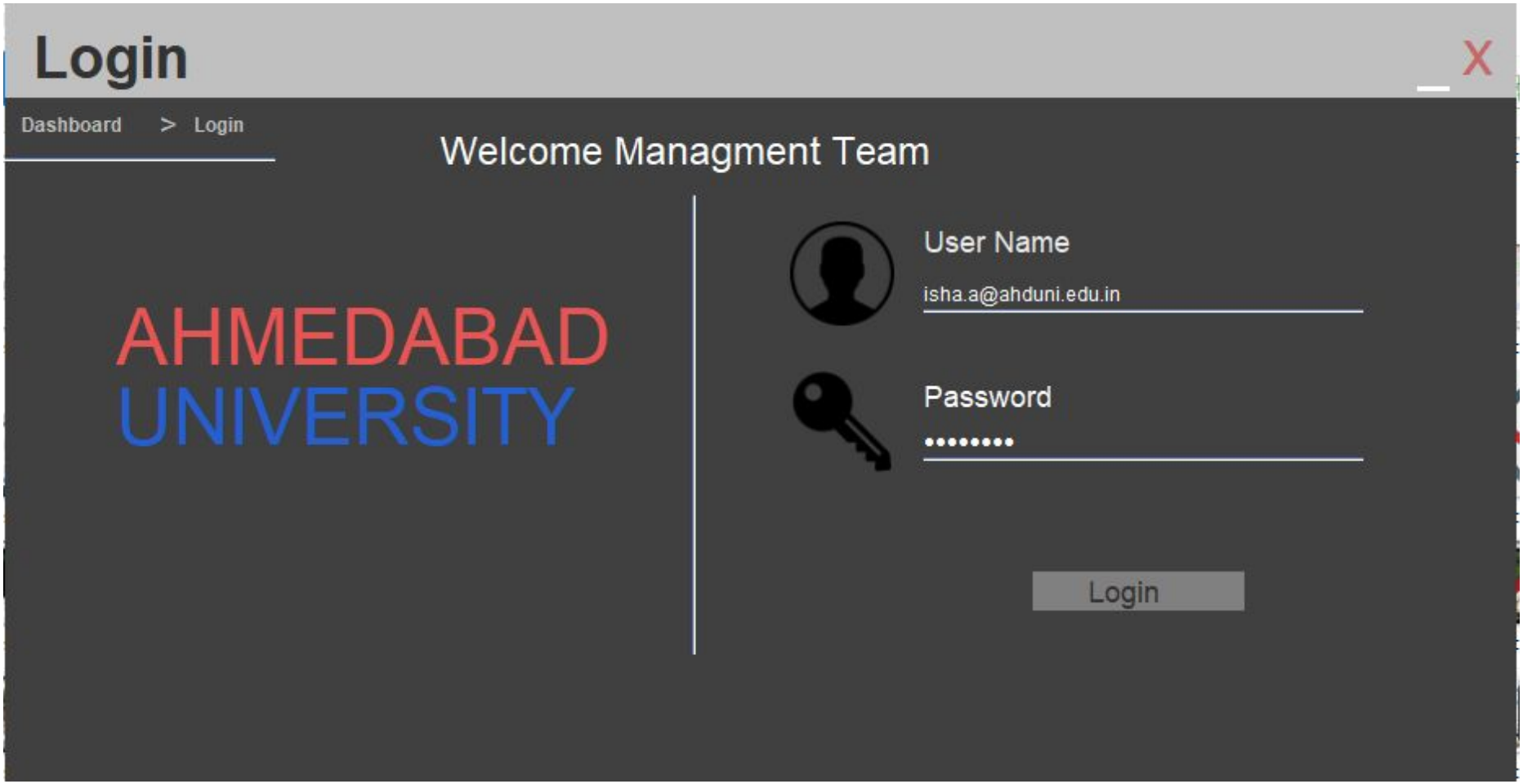
User Interface Design / Screenshots

Screen-10-Faculty Portal(Recognition)



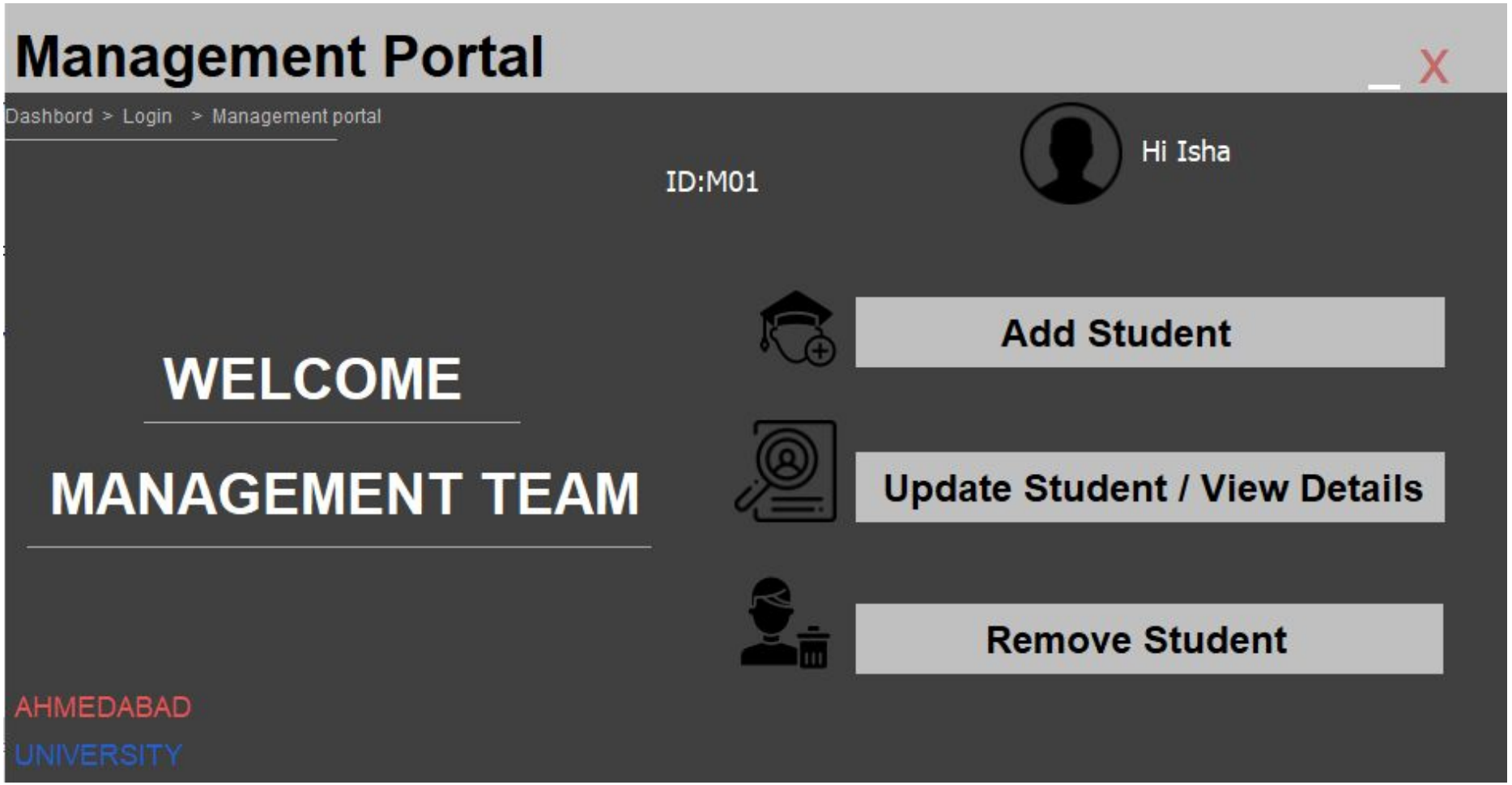
User Interface Design / Screenshots

Screen-11-Management Login



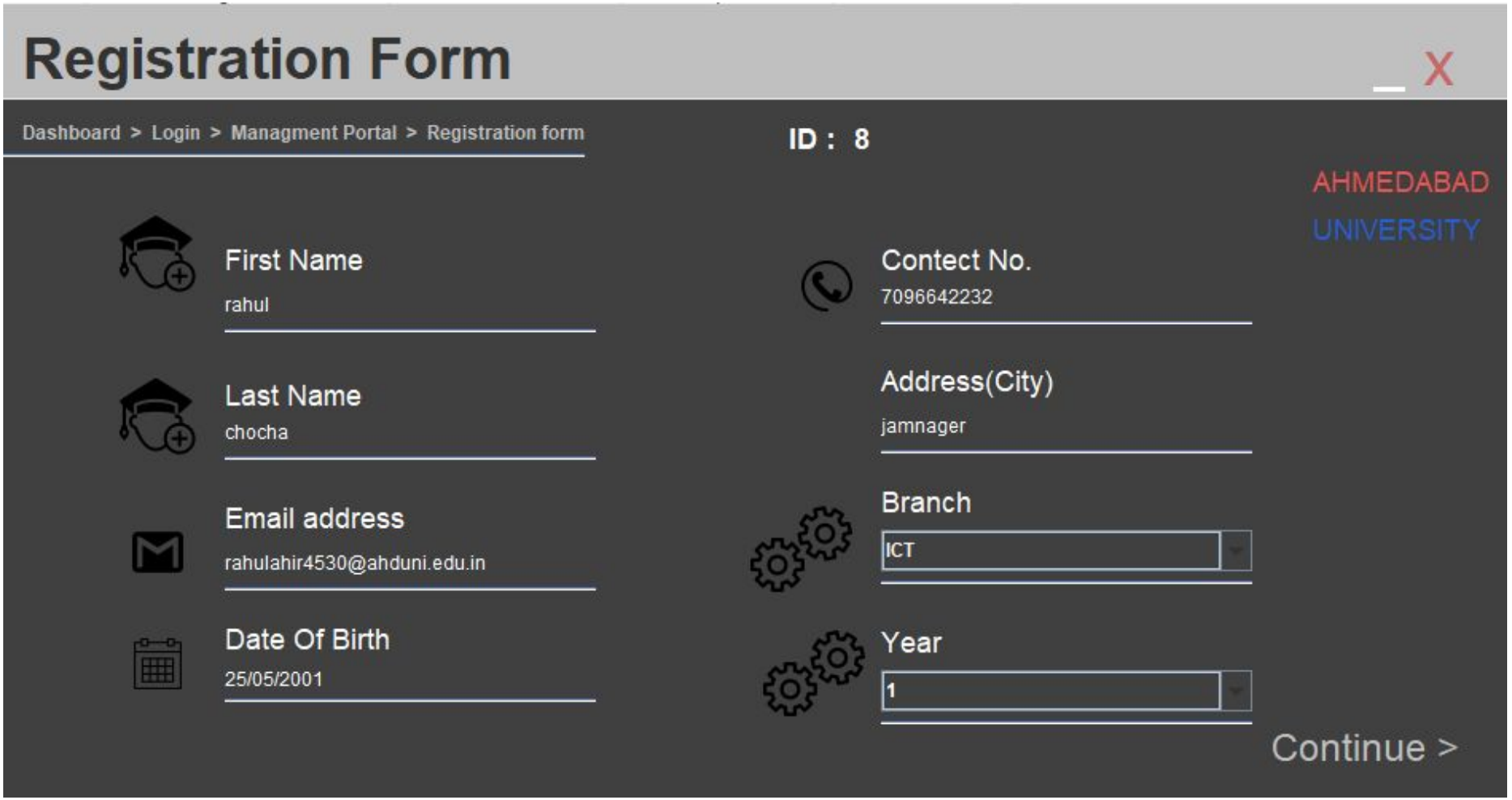
User Interface Design / Screenshots

Screen-12-Management Portal



User Interface Design / Screenshots

Screen-13-Management Portal(Add Student)

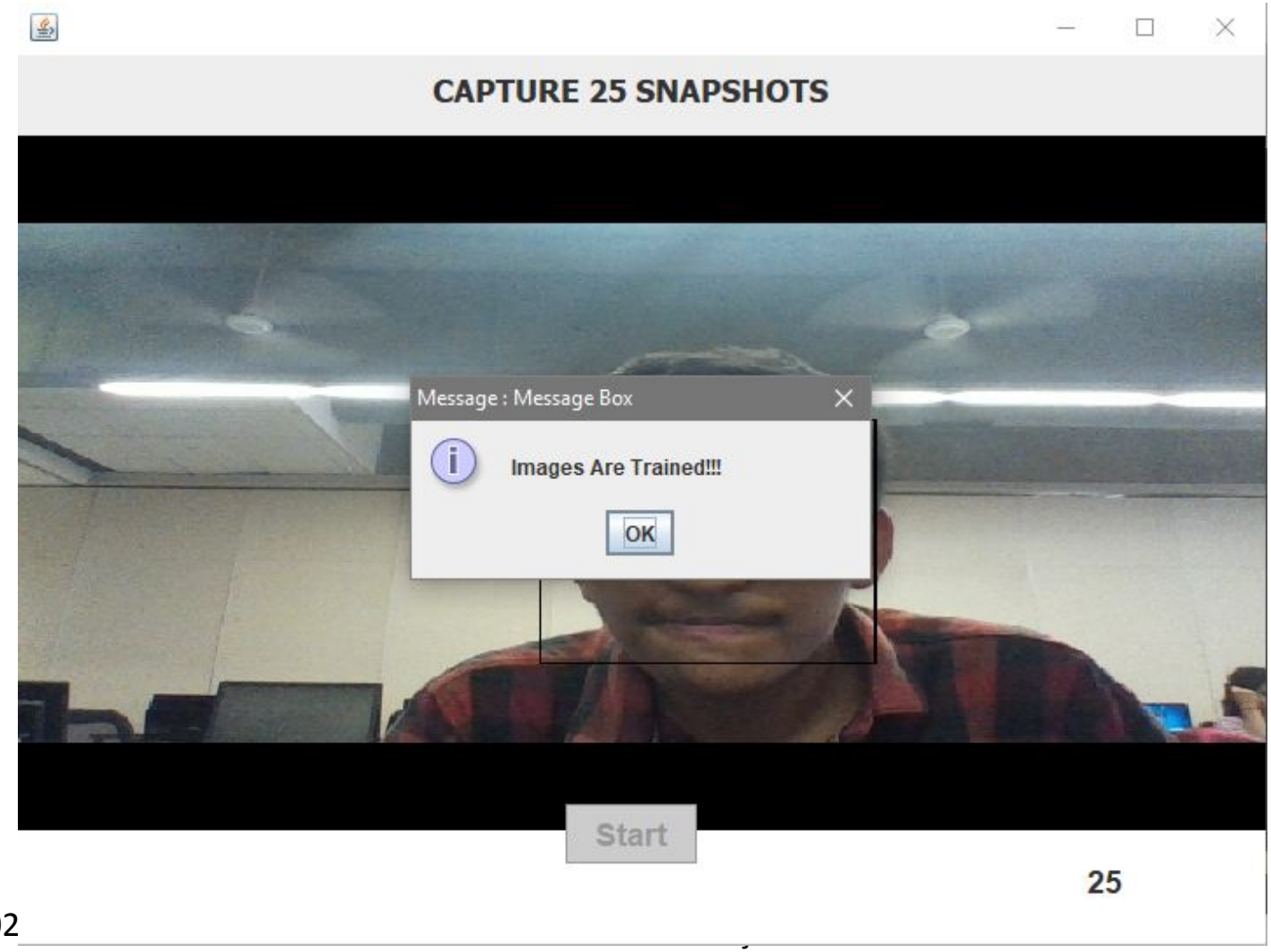


User Interface Design / Screenshots

Screen-14-Management Portal(Face Detection)

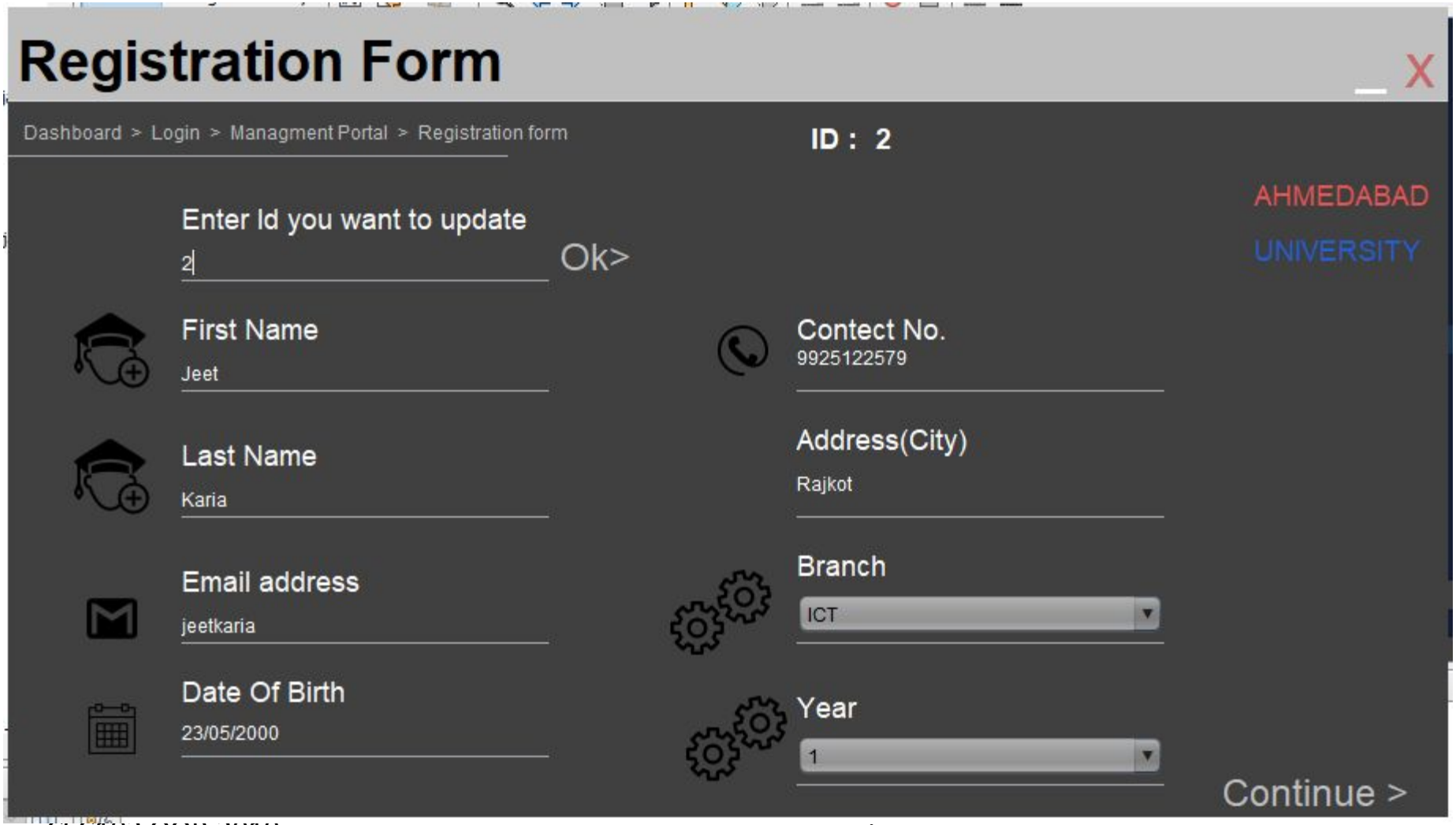
User Interface Design / Screenshots

Screen-15-Management Portal(Data Storing)



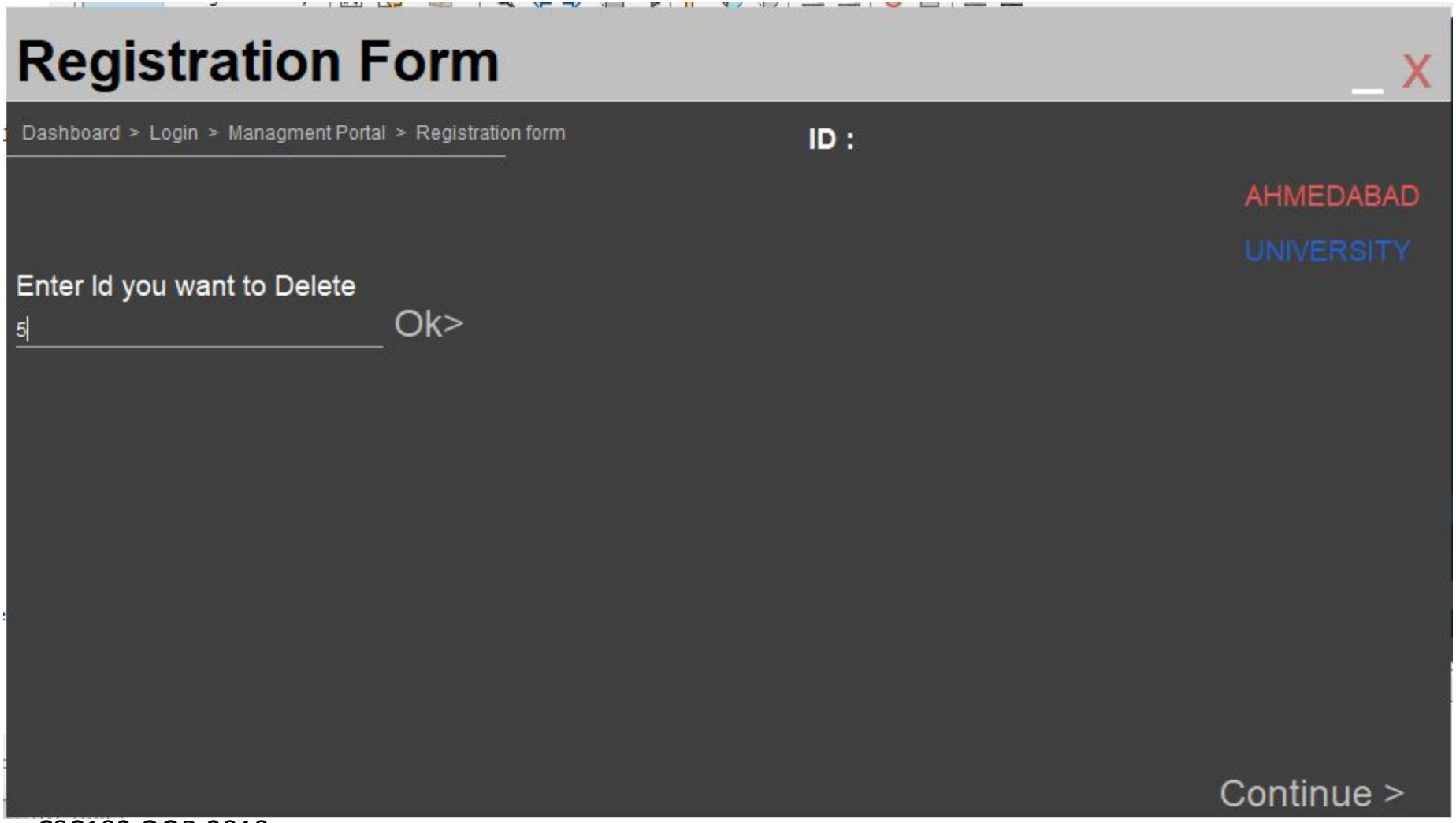
User Interface Design / Screenshots

Screen-9-Update Student Details

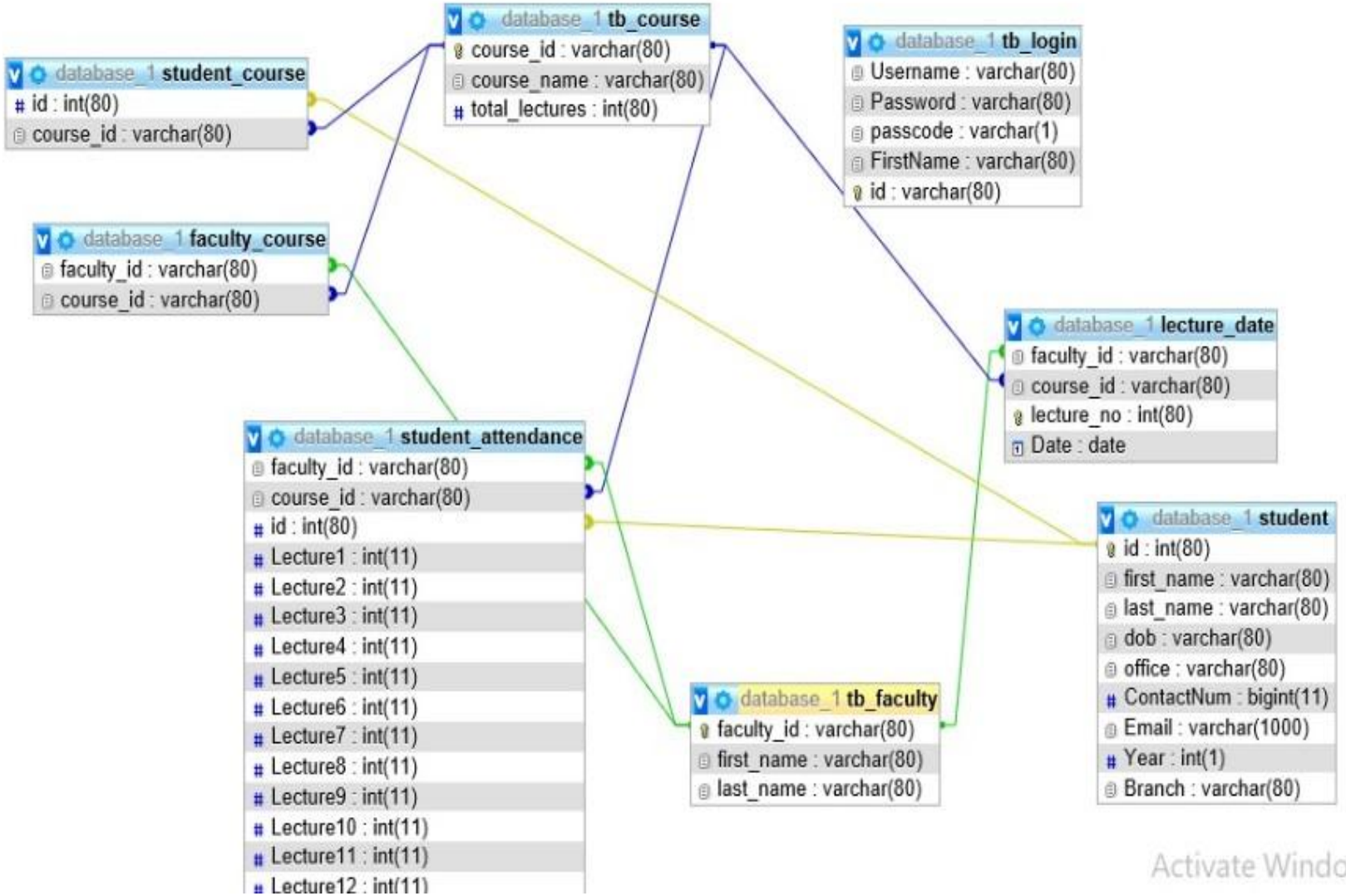


User Interface Design / Screenshots

Screen-15-Management Portal(Remove Student)



Backend structure of SQL tables managing database of people using the software



Something unique to share with the Visitors like - Project Based Learning Experience - Learnings from the course

- Analyze real-life problems and design object oriented solutions to address them
- Analysis, design and code a multi-user working system with database
- Project Based Learning – put ideas into real working system - Collaboration and team work experience
- Learning by doing / hands-on learning / self learning

References:

https://towardsdatascience.com/face-recognition-how-lbph-works-90ec258c3d6b

https://medium.freecodecamp.org/facial-recognition-using-opencv-in-java-92fa40c22f62

Thank You