**Roll Call System by Face Recognition**

2020-08-06

**Nayana Pampapura Madali**

Team leader

**Module**: Select Course Management

ID: 11048498

**Samuel Kalunga**

Team member

**Module**: User Management

ID: 11375012

**Vaishnavi Garikipati**

Team member

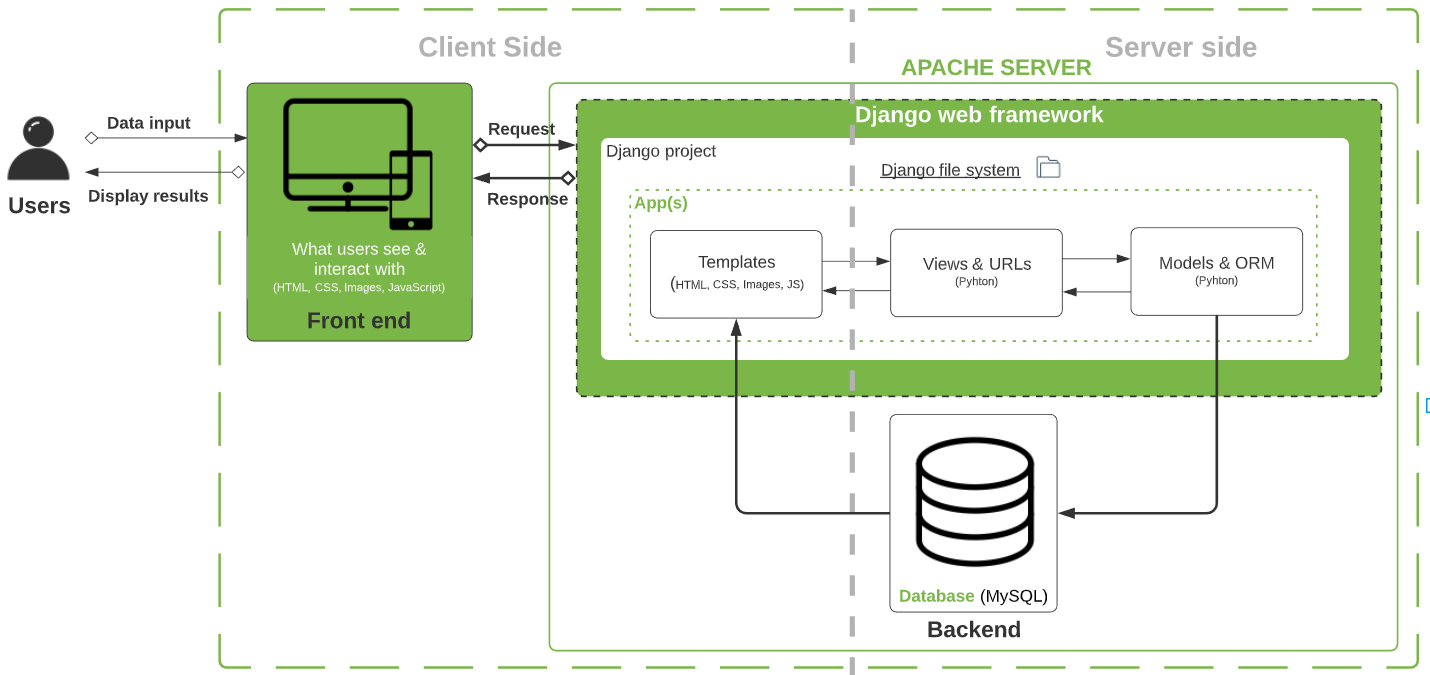
**Module**: Roll Call

ID: 11409444

**1. Objectives**

The system aims to take the roll call of a class by face recognition. The admin user/professor can upload a photo that contains all the students in the class. The system then extracts all the faces in the picture and compares them with the pre-stored photo in the system one by one. If the similarity exceeds over 60% or more, the face could be treated as the specific student's face. Then mark that the student attended the class. The admin user/professor will manage the term, course, and student data. The student user can use his/her student id to log into the system and only view his/her attendance records. ***(Nayana Pampapura Madali)***

**2. Overall Architecture**



*(Overall Architecture. Credit: Samuel Kalunga)*

Our web application architecture consists of both a front and back end. The frontend, also known as the client-side, is everything users see and interact with once they access the website on their browser. The client-side collects data from users and is written in variants of HTML, CSS, and JavaScript. ***(Samuel Kalunga)***

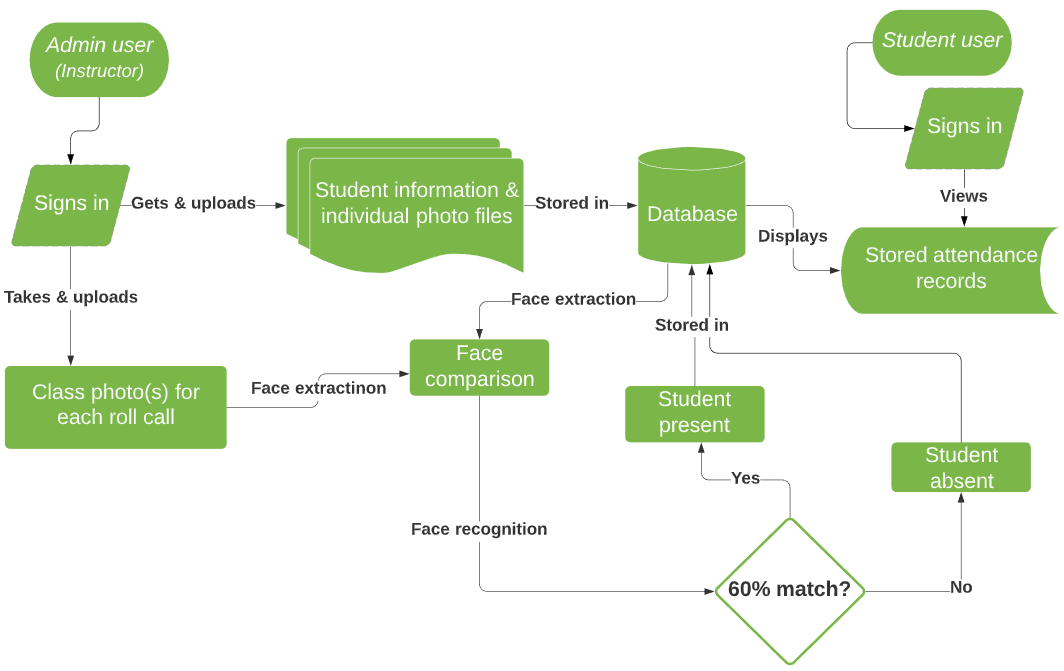
The backend, also known as the server-side of the app, is the part that is not accessible by users and where data is stored and manipulated. The backend processes HTTP requests, mostly fetching the data (text, images, files, etc.) called by the users. Unlike the frontend, the primary language on the backend is Python. ***(Samuel Kalunga)***

We used the Django framework powered by PyCharm and created a project to build the system web pages, connect to a MySQL database, and program it all in one place. We structured our Roll Call project in apps following the Models-Views-Templates (MVT) architecture: ***(Samuel Kalunga)***

* **Models** is a Python file that is a set of tools to design our data tables, which will be migrated to a relational database (MySQL) through ORM (Object Relational Mapping). ***(Samuel Kalunga)***
* **Views** is another Python file where functions and methods are implemented to determine the business logic for the data found in Model. Display logic is organized in URLs kept in a separate file. ***(Samuel Kalunga)***
* **Templates** are HTML files that users see as web pages on their screen based on the business and display logics set in *Views*. ***(Samuel Kalunga)***

**3. System design**

**3.1. System flow**

****

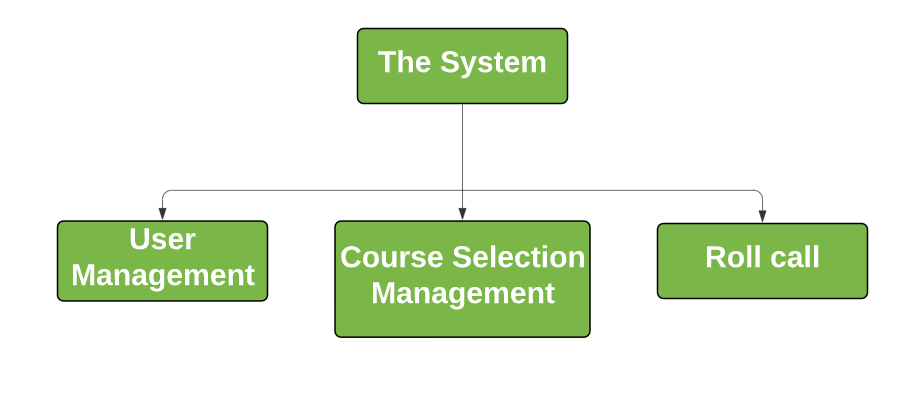
*(Roll Call System Flow. Credit: Samuel Kalunga)*

Admin users log in/sign into the roll call system and put in the details of term, course, and associate the term and the course using course selection. The admin user, who is also the professor, puts in the student's information into the database. The student information consists of student number, student name, and the select course id, which is used to identify the course and the term the student is enrolled in. Then the admin user puts in the student photos into the system. This is a one time process. And while taking the roll call, the admin user will select the term, course, and the date for which the roll call needs to be taken. Later the admin uploads the class photo/photos into the system. As part of face recognition, the images uploaded during roll call and the student photos initially uploaded will be encoded and compared. Based on the comparison or the match rate with 60% as the threshold, the student is marked as present or absent. ***(Nayana Pampapura Madali)***

The student is another user our rollcall system has, and the student can log into the system to only view his/her attendance records in all the courses the student is enrolled in. The admin and the student login credentials are pre-defined in our system. ***(Nayana Pampapura Madali)***

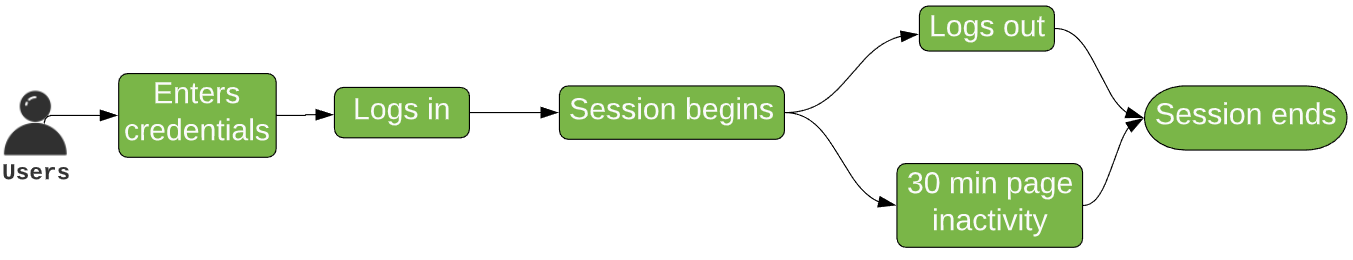
Face recognition is an integral part of the project. In our project, face recognition is carried out using different models like OpenCV, dlib, NumPy, face recognition, etc. The photos uploaded as a part of student information are first encoded and stored in the system. The images uploaded as a part of roll calls are stored and encoded in the system. Later the face locations of these roll call images are determined to identify the number of faces in an image. Then the roll call image encodings are compared with the encodings of the initially uploaded pictures of the student to determine if there is a match or not. If the match rate is higher than 60% than the match is marked as true. ***(Nayana Pampapura Madali)***

**3.2. System functions**

****

*(Roll Call System Functions. Credit: Samuel Kalunga)*

1. **User Management**

****

*(User Management process figure. Credit: Samuel Kalunga)*

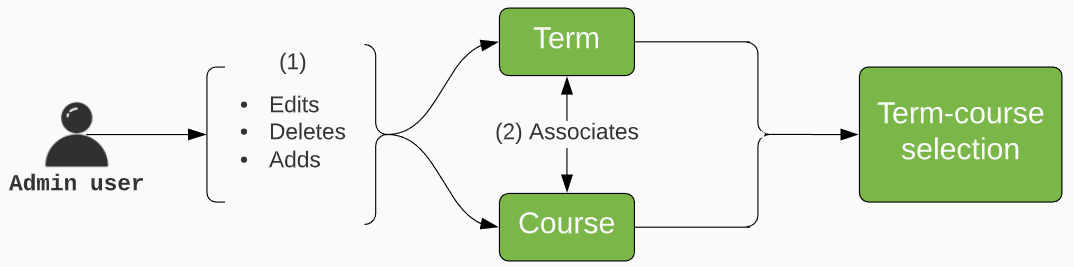
This module manages users’ login credentials and sessions. Those credentials are pairs of username (student ID) & password predefined in the system. To log in, users enter a predefined username and password provided by the system developers. ***(Samuel Kalunga)***

There are two types of users:

1. System administrator (instructor)
2. Students

Since users need to be authenticated to access the website, the system requires them to log in, and sessions are mechanisms that enable monitoring login information for specific users. The session framework stores data on the server-side and retrieves them on a per-site-visitor basis. Once the session is generated by a user’s login action, it lasts for a default time period of 30 minutes, and session variables are accessible to that user only. A session begins once users log in and ends when they log out, or it times out. ***(Samuel Kalunga)***

**b. Select Course Management**

****

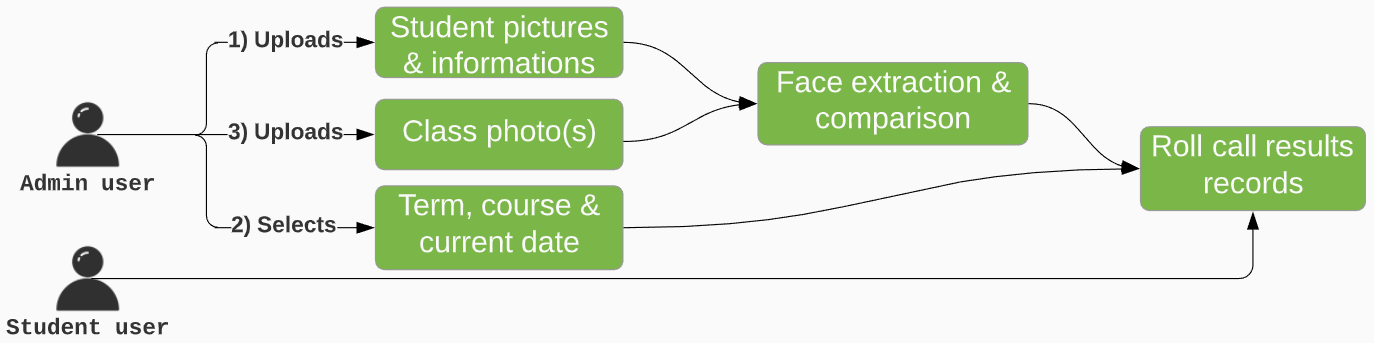
*(Select Course Management process figure. Credit: Samuel Kalunga)*

The module contains three kinds of important information which are term, course and course selection. It is used to determine who selects the course, and in which term the course opens. When the course selection is done, the basic information with their photo of students in this course will be imported into the system. ***(Nayana Pampapura Madali)***

There are three pages created for this module.

1. **Term Page:** On the term page, the admin user/professor can add, edit or delete a term. Term page is associated with the term table in the database. ***(Nayana Pampapura Madali)***
2. **Course Page:** On the course page, the admin user/professor can add, edit or delete a course. Course page is associated with the course table in the database. ***(Nayana Pampapura Madali)***
3. **Course Selection Page:** On the course selection page, the admin user can make the association between the term and course. Also, the user can edit and delete the records present in the course selection page. The course selection page also has the course start date and course end date. Course selection page is associated with the course selection table in the database. ***(Nayana Pampapura Madali)***

**c. Roll Call**

****

*(Roll Call process figure. Credit: Samuel Kalunga)*

The roll call module is the heart of this system where the face recognition takes place to mark attendance of the students. Main functions are applicable only to the admin user who is also the instructor/lecturer. There are 3 tasks for the admin to perform as soon as he logs in and the session starts. ***(Vaishnavi Garikipati)***

1. **Upload Student Information:** The admin has the knowledge of terms, courses and students enrolled in each of the courses, which he needs to feed to the system. He can do this uploading an excel file which contains details like student name, number, and select course id (scid). This will be one-time upload as the data will be stored in the database after successful upload. Also, it is used to import all the student details for a particular class in a certain course and term. ***(Vaishnavi Garikipati)***
2. **Select Term, Course & Date:** Now that there is data about students in the system, the admin has to decide for which class he intends to take roll call. For this purpose, a course and term along with respective roll call date must be selected. This link between students and course is enabled by filtering among available terms and courses. Similarly selecting the date will allow for that particular day’s attendance to be linked to the respective students from selected curse. ***(Vaishnavi Garikipati)***
3. **Upload Photos:** The admin has all the required details by now. He needs to take attendance for the class he intends, and this is served by uploading photos of the students. The images labeled with student number for encoding and image processing will be uploaded to the system using the “Start Roll Call” page. The admin can also have a preview of the photos before submitting and a reset button to change photos before upload. Once the images are saved to the system, the latest photo will be compared against the existing photo from initial upload. Match rate is calculated, and the student is marked present if match rate exceeds 60%, else absent. If the match rate is less and a student got marked absent, the admin can edit the attendance column and change the status as present. All the attendance records will be stored and can be viewed in the “History” page. ***(Vaishnavi Garikipati)***

Students also can login to the system but can only view their attendance records. Once a student logs in and the session starts, he can view his entire attendance records. He/She can filter courses or check attendance by date. No other action is enabled for them. ***(Vaishnavi Garikipati)***

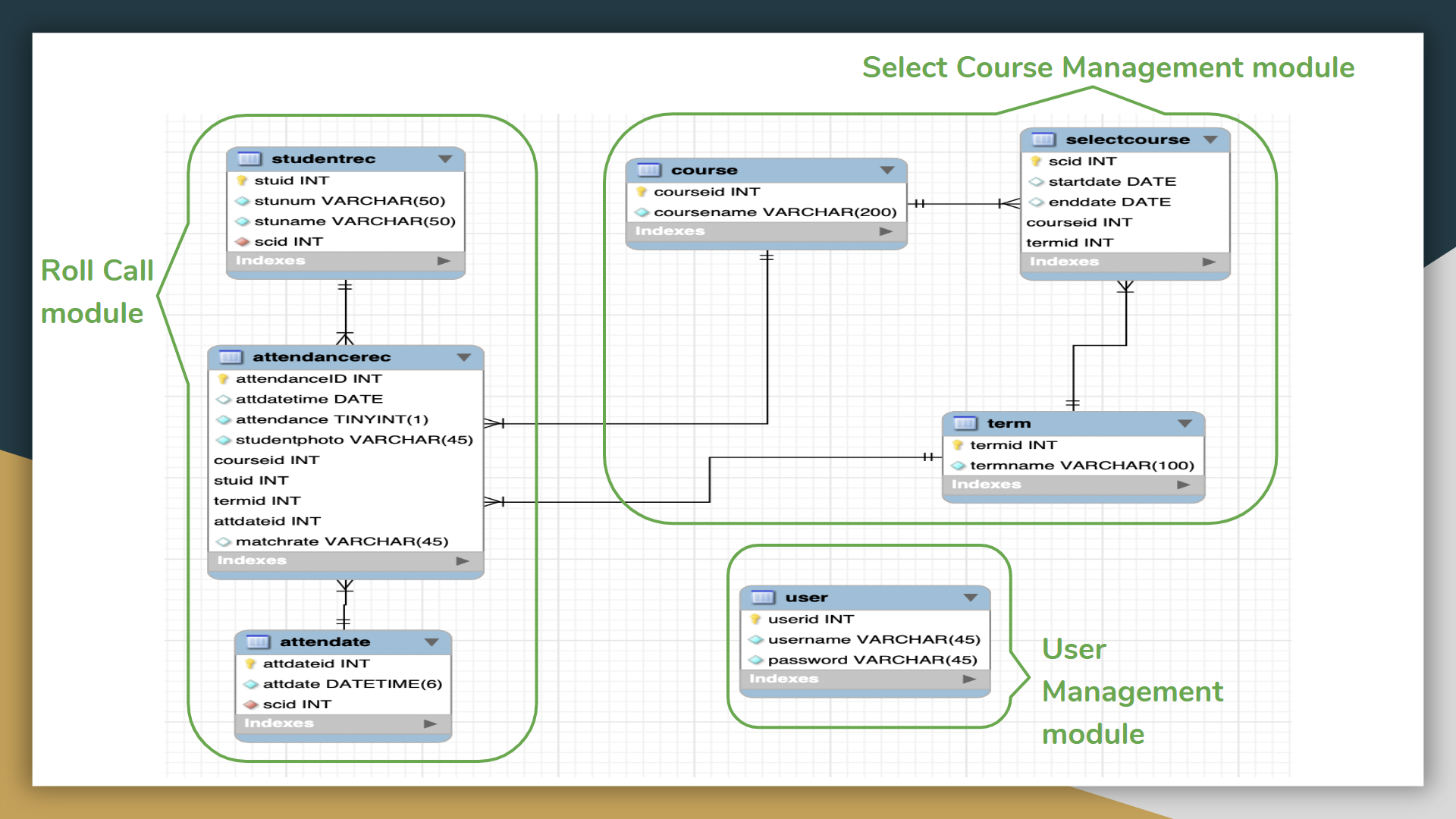
**3.3. Database Design**

The user table consists of username and password for both admin and students.

The course and term tables provide information on all the courses that run in a semester and the times at which it runs. However, select course table establishes the relation between term and course.

The studentrec table contains information of all the students. The attendate table is used to keep a record of the date whenever the attendance has been called. While the attedancerec table holds information about roll call results, it is linked with all other details like respective courses in particular terms and student information as well to maintain the entire record of attendance history.

The figure below shows the database structure for the entire Roll Call System.***(Vaishnavi Garikipati)***



*(ER Diagram for Roll Call System. Credit: Samuel Kalunga)*

Schema of tables in database with self-explanatory description are listed below

**a. User Management table(s)**

1. **User**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FieldName** | **DataType** | **Primary Key** | **Foreign Key** | **Description** |
| userid | Int | Yes |  | User ID |
| username | Varchar |  |  | User name |
| password | Varchar |  |  | Password |

***(Samuel Kalunga)***

**b. Select Course Management table(s)**

1. **Course**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FieldName** | **DataType** | **Primary Key** | **Foreign Key** | **Description** |
| courseid | AI | Yes |  | Course ID |
| coursename | Varchar |  |  | Name of the course |

***(Nayana Pampapura Madali)***

1. **Term**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FieldName** | **DataType** | **Primary Key** | **Foreign Key** | **Description** |
| termid | AI | Yes |  | Term ID |
| termname | Varchar |  |  | Name of the term |

***(Nayana Pampapura Madali)***

1. **Select Course**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FieldName** | **DataType** | **Primary Key** | **Foreign Key** | **Description** |
| scid | AI | Yes |  | Select Course ID |
| termid | int |  | Yes | Term ID |
| courseid | int |  | Yes | Course ID |
| startdate | date |  |  | Course start date |
| enddate | date |  |  | Course end date |

***(Nayana Pampapura Madali)***

**c. Roll Call table(s)**

1. **Studentrec**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FieldName** | **DataType** | **Primary Key** | **Foreign Key** | **Description** |
| stuid | Int | Yes |  | Studentrec Id |
| stunum | Varchar |  |  | Student Number |
| stuname | Varchar |  |  | Student Name |
| scid | Int |  | Yes | Select Course ID |

***(Vaishnavi Garikipati)***

1. **Attendate**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FieldName** | **DataType** | **Primary Key** | **Foreign Key** | **Description** |
| attdateid | Int | Yes |  | Attendate Id |
| attdate | Datetime |  |  | Attendance Date |
| scid | Int |  | Yes | Select Course ID |

***(Vaishnavi Garikipati)***

1. **Attendancerec**

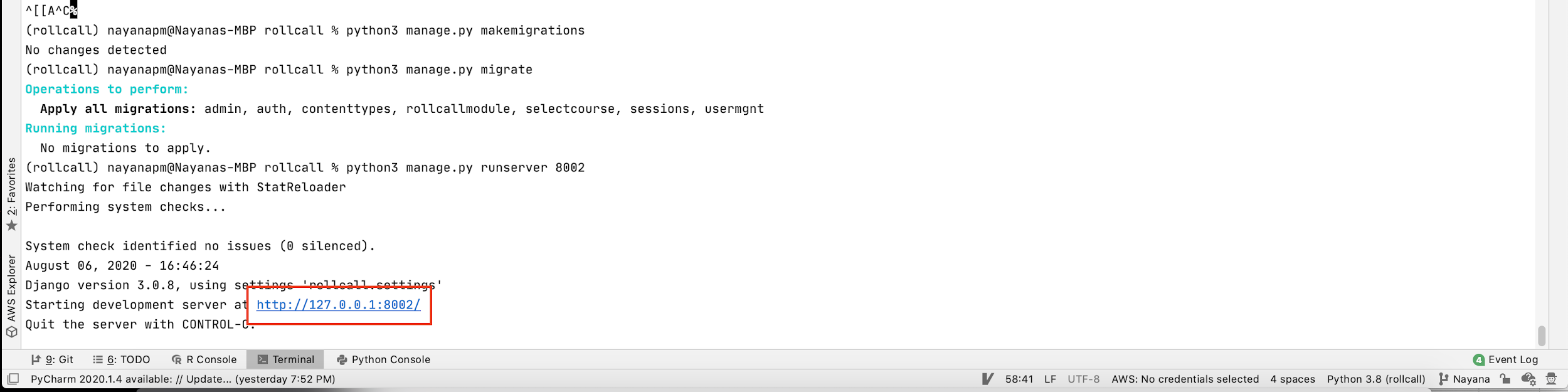
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FieldName** | **DataType** | **Primary Key** | **Foreign Key** | **Description** |
| attendanceID | Int | Yes |  | attendancerec id |
| attdatetime | Date |  |  | Attendance Date |
| attendance | Tinyint |  |  | Attendance(present/absent) |
| studentphoto | Varchar |  |  | Student Photo |
| courseid | Int |  | Yes | Course id |
| stuid | Int |  | Yes | studentrec id |
| termid | Int |  | Yes | term id |
| attdateid | Int |  | Yes | attdate id |
| matchrate | Varchar |  |  | Match Rate |

***(Vaishnavi Garikipati)***

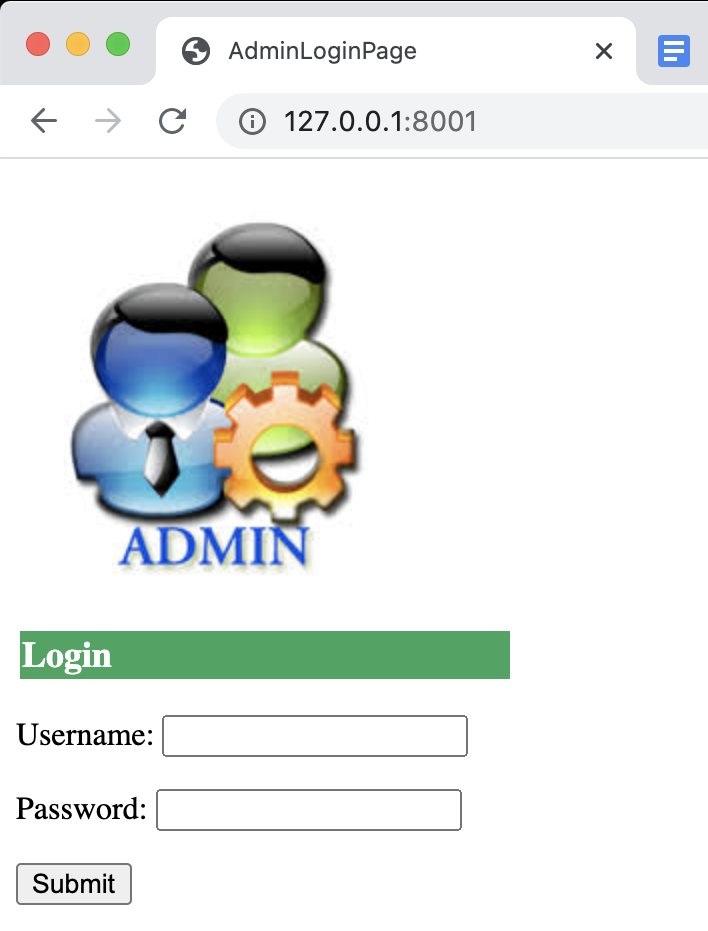
**4. System Testing**

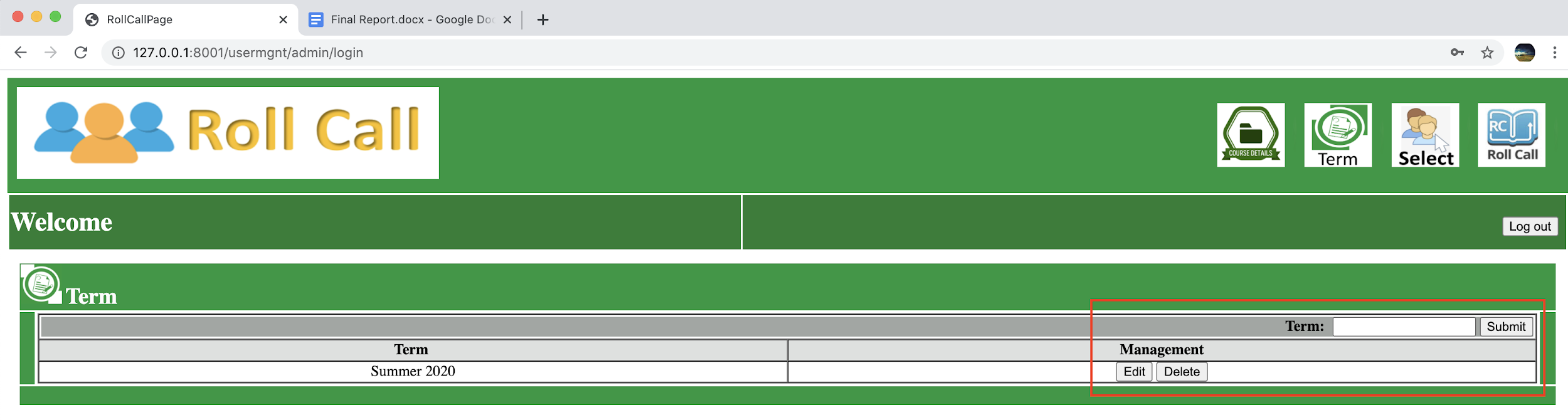
System testing was carried out for the entire system to test the functionality of the system. Below are the images captured while system testing. ***(Entire system testing carried out and written by Nayana Pampapura Madali)***

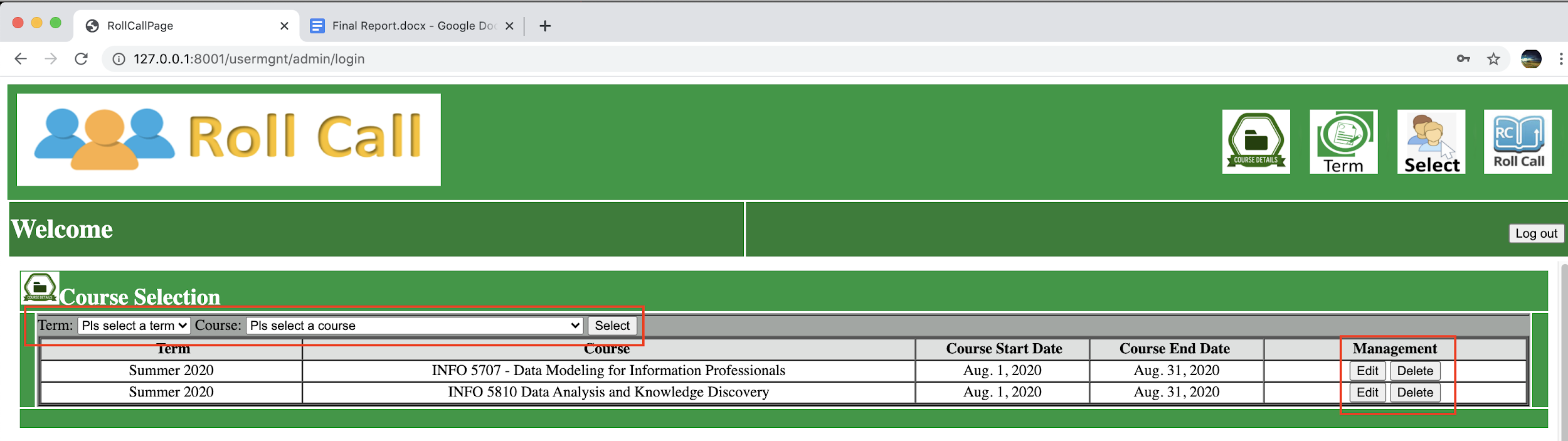
1. When the server is run on Pycharm, the user lands on the admin login page. The admin user logs in to the system using a predefined username and password. Below screenshot shows Pycharm terminal where the server is run.



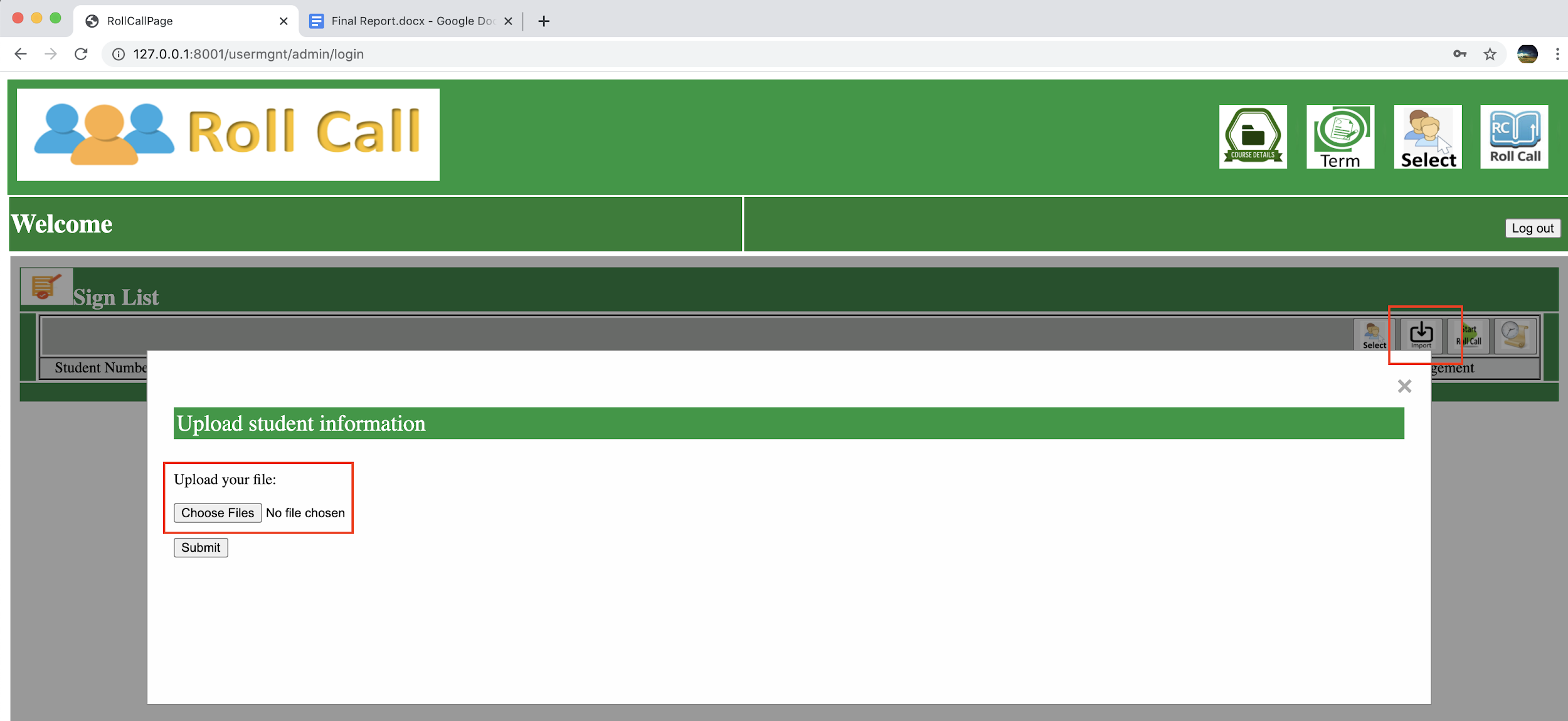
Below is the screenshot of the admin page.



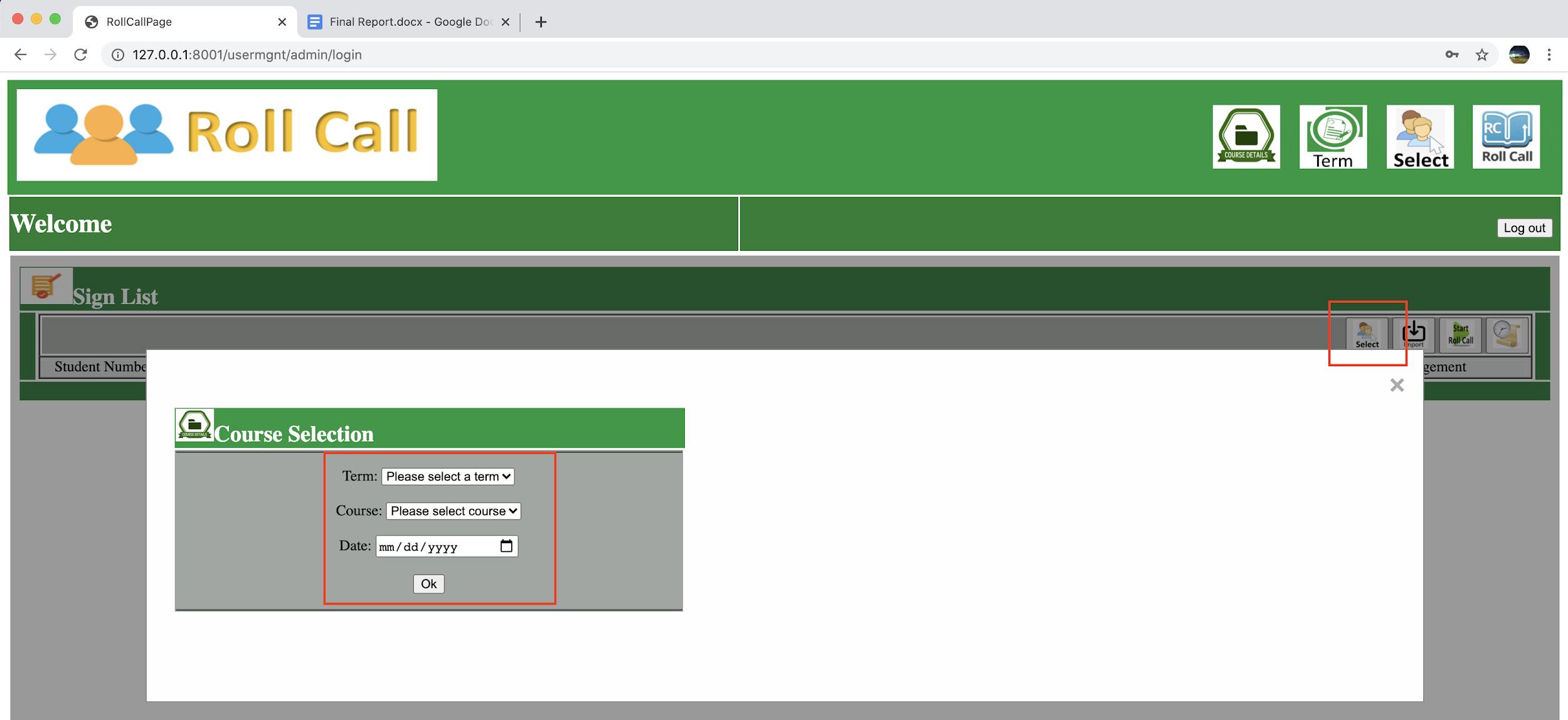
1. Once the admin/professor logs in, he/she will land on the course page, which is used to list the courses. The admin can either add, edit, or delete a course record. 
2. Next admin navigates to the term page. The term page is used to list the terms. The admin can either add, edit, or delete a term record. 
3. Terms and courses data defined in their respective pages are not related to each other. The relation between term and course is established using the course selection page, which is next to the term page. On this page, the admin user/professor will match a course to a term. The course selection table has a course start date and course end date details as well. The admin user can edit or delete a record here.



1. The next module is the roll call module. Before taking the roll call, the student information needs to be put into the database. This is done using the “Import” widget. The student details like the student number, student name, and the select course id are put into the database using an excel file. The student photos can be imported using the same page as well.



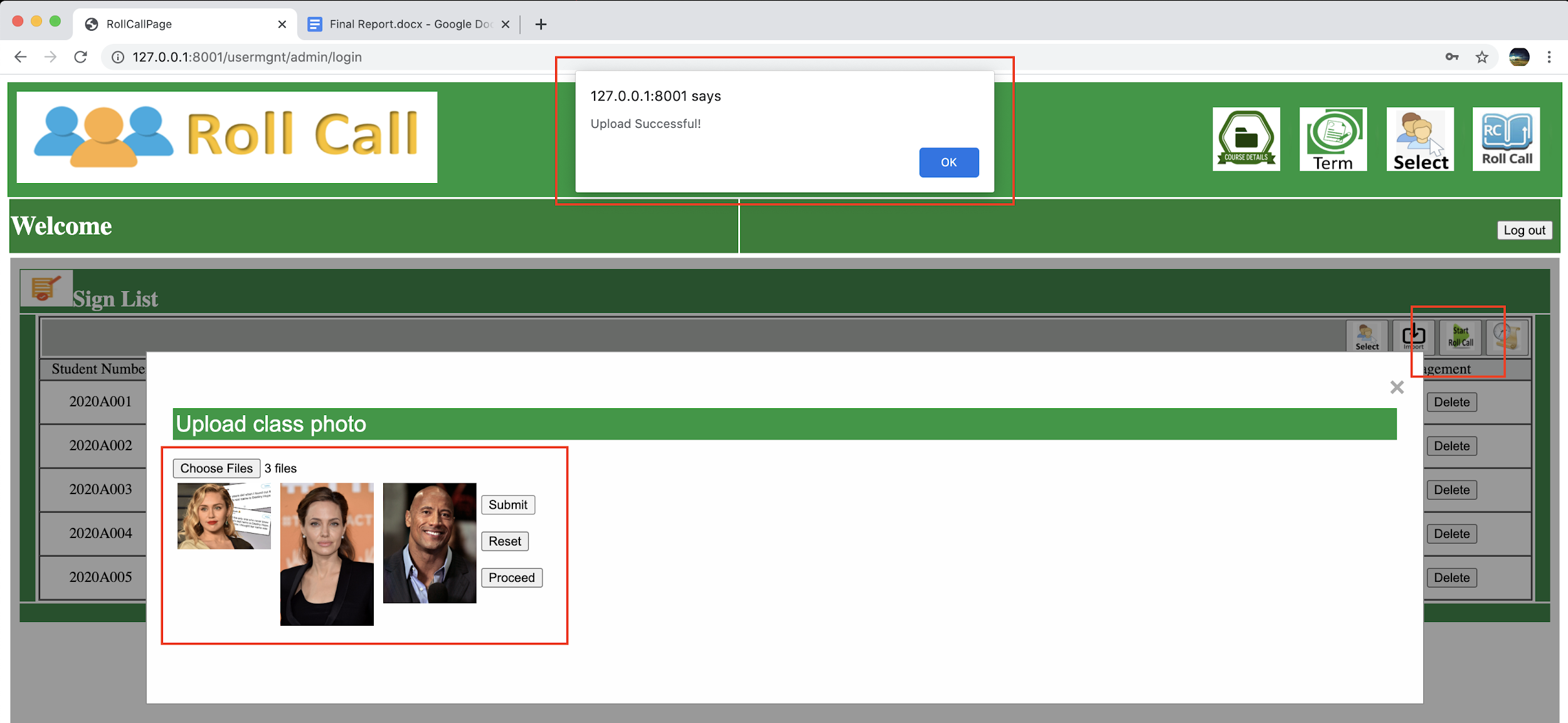
1. The admin or the professor will now select the term, course, and the date for which the attendance needs to be taken.



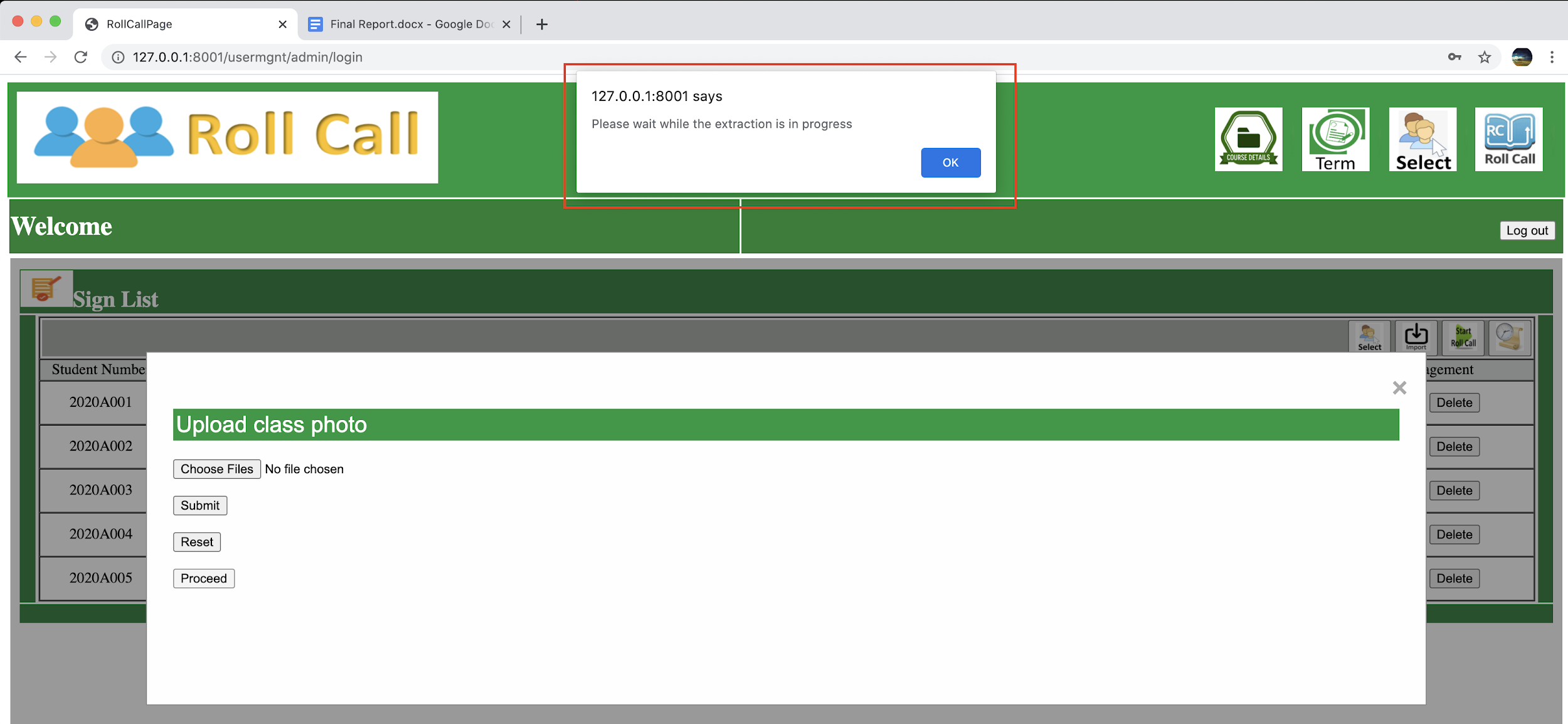
1. Now, if the admin goes to the sign list page(roll call page) and refreshes the page, he/she will be able to see the list of students present in that particular term and course. The admin can edit or delete a student record. The attendance % column refers to the overall student attendance percentage in a specific course.



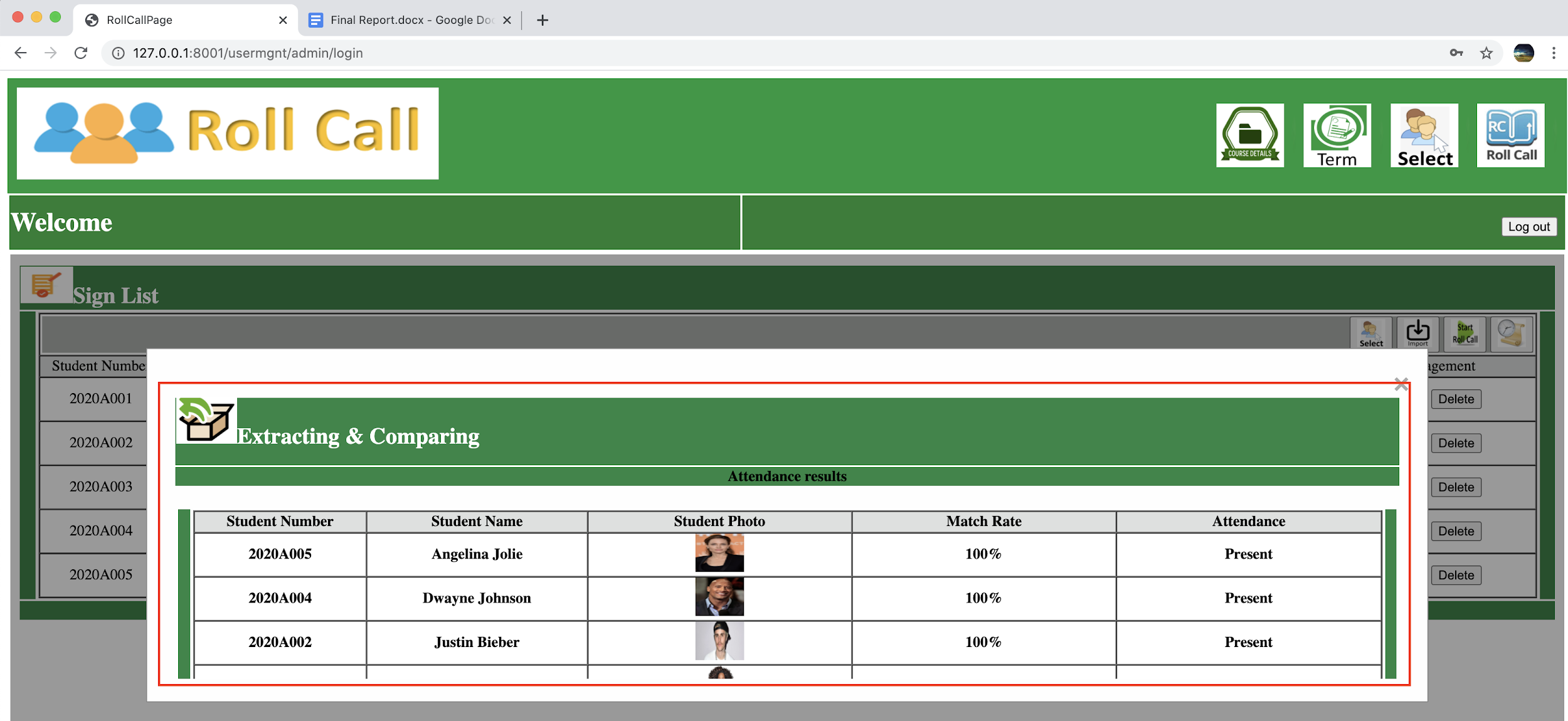
1. The attendance is taken by clicking on “Start Roll Call.” Here, the class photos should be uploaded into the system using the “Choose Files” button and then clicking on the submit button. The reset button is used to reset the page.



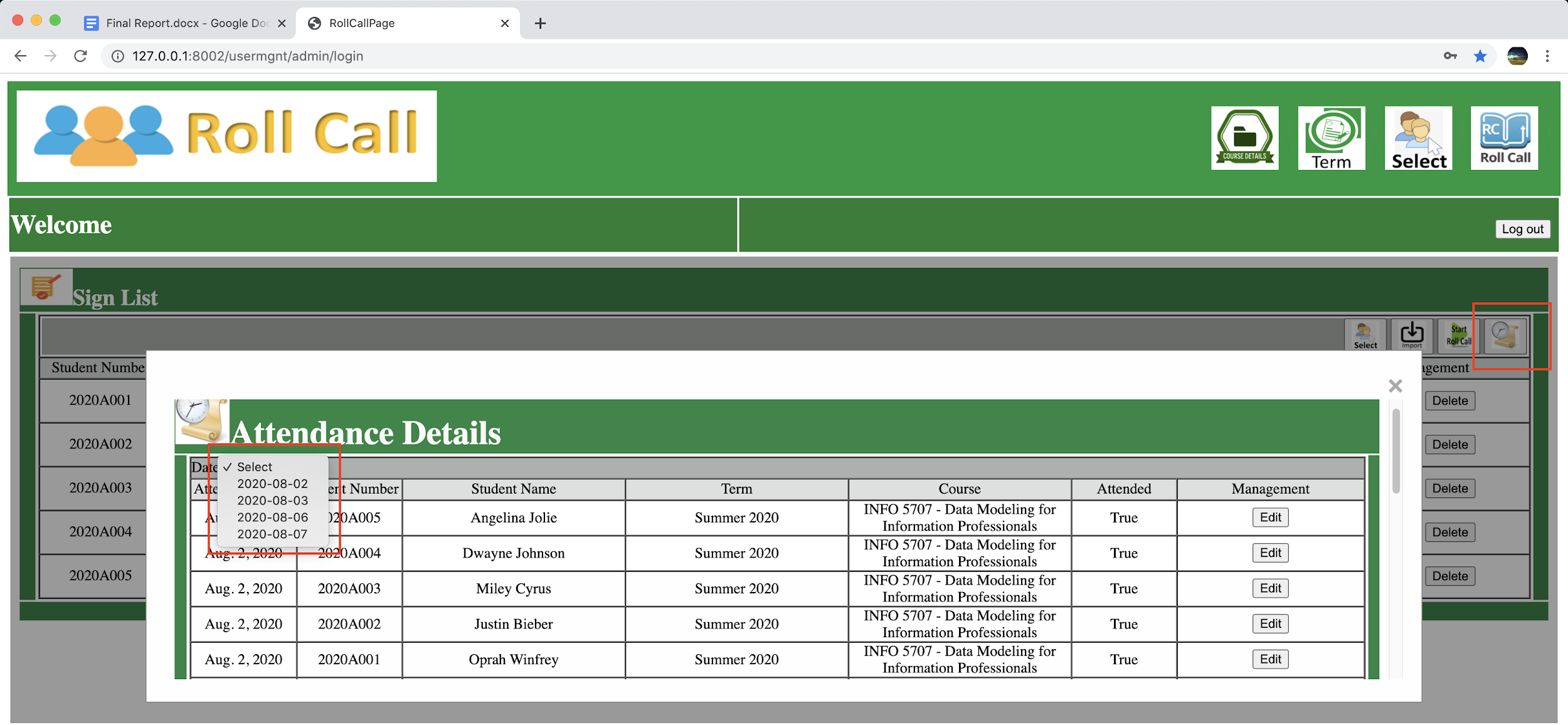
1. Once the roll call photos are uploaded successfully, and when the proceed button is clicked, the face extractions take place, and the results are displayed. The match rate between the photos uploaded during roll call and the images uploaded as a part of the initial student data load will be compared, and their match rate is determined. If the match rate is higher than 60%, then the student is marked as present; if not, the student is marked as absent. The below screenshot refers to that face extraction and recognition process while in progress.



Below screenshot refers to the face extraction result page.



1. We have a history page that is specific to the term and course selected and which lists all the attendance details taken for a particular course. The edit button is used to change the attendance of a student. The attendance records can be filtered based on the roll call dates.

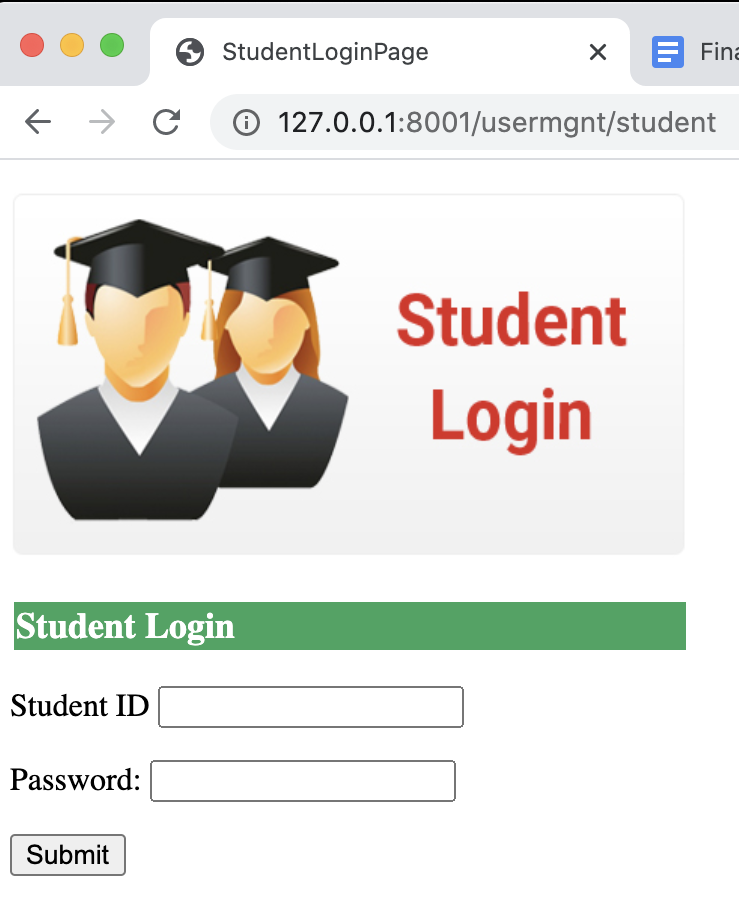


1. The admin can log out of the system by clicking on the logout button.

A screenshot of a video game

Description automatically generated

1. The student login page is used by the student to login to view their attendance records. The student will use their student id to login. The student login credentials are predefined in the system.



1. After the student logs into the system successfully, he/she will land into the "My Attendance" page, which will list all the attendance related to the student. If the student is enrolled in multiple courses, all the course's student records will be displayed. The student can filter the attendance record either by using "Course" dropdown alone or filter by "Date" solely or use both "Course" & "Date" to filter for the specific attendance record of the student. Once done with checking the attendance, the student will use the "Logout" button to logout of the system.

