Smart Attendance System

By

17BIT034 (Kandarp Kakkad)
17BIT046 (Siddharth Marvania)
17BIT075 (Parth Patel)
17BIT085 (Himanshu Prajapati)

तमसो मा ज्योतिर्गमय

NIRMA
UNIVERSITY

INSTITUTE OF TECHNOLOGY

NAAC ACCREDITED 'A' GRADE

Department of Computer Science & Engineering
Ahmedabad 382481

Smart Attendance System

Minor Project

Submitted in partial fulfillment of the requirements
For the degree of
Bachelor of Technology in Information Technology

By
17BIT034 (Kandarp Kakkad)
17BIT046 (Siddharth Marvania)
17BIT075 (Parth Patel)
17BIT085 (Himanshu Prajapati)

Guided By
Prof. Dvijesh Bhatt
Department of Computer Science & Engineering

NIRMA UNIVERSITY
INSTITUTE OF TECHNOLOGY
NAAC ACCREDITED 'A' GRADE

Department of Computer Science & Engineering
Ahmedabad 382481

Certificate

This is to certify that the Minor Project entitled "Smart Attendance System" submitted by Kandarp Kakkad(17BIT034), Siddharth Marvania(17BIT046), Parth Patel(17BIT075) and Himanshu Prajapati(17BIT085), towards the partial fulfillment of the requirements for the degree of Bachelor of Technology in Information Technology/Computer Engineering of Nirma University is the record of work carried out by him/her under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination.

Prof. Dvijesh Bhatt
Assistant Professor
Department of CSE,
Institute of Technology,
Nirma University,
Ahmedabad

Dr. Madhuri Bhavsar(HOD)
Department of CSE,
Institute of Technology,
Nirma University,
Ahmedabad

Acknowledgement

We would like to humbly thank Nirma University for giving us the opportunity to work on this project. We would like to express special gratitude and thanks to our project guide Prof. Dvijesh Bhatt whose kind support and help has made us complete the project duly. We would also like to appreciate Prof. Parita Oza for her different perspective which led to new ideas.

We would also like to express our gratitude to our parents and friends for their valuable insights and encouragement.

Abstract

This report is a basic documentation of the product of ”Smart Attendance System”. This is a small-scale prototype for attendance system using face recognition. The backend of the web application uses DL based Face Recognition algorithm trained on photos from database. The database contains the model weights and model config to restore or update the model. This is a stand-alone and easy-to-use system.

Contents

List of Figures

Chapter 1

Introduction

1.1 General

Smart Attendance system is a difficult task and providing facilities in each and every field is a burdensome task. Problems like paperwork, updating attendance and modifying the attendance. Giving an effective method to communicate among teachers and students isn't a simple task.

The main objective of the project is to provide an efficient way of managing university processes. Records include student data, teacher data, attendance data etc. The goal is to provide an online system to perform tasks like inserting, fetching and manipulating the data, various online services. In this system we will be reducing the paperwork and also some offline evaluation will be done online.

Deep Learning has automated laborious tasks like security surveillance and medical image analysis. Similarly, Deep Learning can also automate attendance systems. So, the Smart Attendance System will be a substitute for the tedious manual attendance system which is currently in practice.

As many universities have adopted smart attendance systems, we are trying to make this change in Nirma University as well. This will ensure real-time attendance uploading thereby removing the teacher/professor interference. Smart Attendance System removes redundancy and saves time. It also reduces mistakes committed during attendance.

Our project will work as a standalone system along with automatically recording attendance, it will also provide the user interface for students, professors and administration. After completion, the project can be extended to provide functionalities like detecting prohibited activities, requesting leave applications from students with sub-par attendance and also reduce the manual efforts of creating the attendance review reports and other paperwork.

1.2 Actors

This project contains mainly 4 actors:

1. Professor
2. Faculty Advisor
3. Student
4. Admin

1.2.1 Professor

Right now the problems that professors are facing is the manual attendance taking and uploading it later. In this way they have to waste their time and the university wastes its resources like paper. Also there is a chance of mistake while taking attendance or uploading it. The uploading means to re-enter the manual attendance. Also there are chances of proxy.

1.2.2 Faculty Advisor

The problems that faculty advisors face is to generate the review report at the end of semester. To review all the students' attendance subject-wise and make a document of data is very tedious and time consuming.

1.2.3 Student

The problems that students face is untimely updates of attendance, mistakes in attendance taking. Student does not know how many bunks of the class is still left before safe zone.

1.2.4 Admin

The problems that admin face is manual update of each entry if the update is required. The database is not clear.

Chapter 2

Functionality

2.1 Professor

The following are the functionalities of professor:

- The professor can take attendance.
- The professor can modify attendance.
- The professor can update their details and inform other professor if they are not available for the lecture.
- The professor can update password, achievements and authority.
- The professor can fill out form to delete the attendance of a specific student and send it to admin for the procedure.

2.2 Faculty Advisor

Following are the functionalities of faculty advisor:

- The faculty advisor can take attendance.
- The faculty advisor can modify attendance.
- The faculty advisor can update their details and inform other professor if they are not available for the lecture.

- The faculty advisor can update password, achievements and authority.
- The faculty advisor can fill out form to delete the attendance of a specific student and send it to admin for the procedure.
- The faculty advisor can generate an Attendance lagging Report of its respective department.

2.3 Student

The following are the functionalities of student:

- The student can view attendance.
- The student can view attendance of a particular subject date wise.
- The student can also check if their attendance is above 85% or not.

2.4 Admin

The following are the functionalities of admin:

- The admin has access to all the databases of the software.
- The admin has the right to add the content to the databases.
- The Admin have the right to modify or edit the existing information in the database.
- The Admin has some special features of filter, which is used to filter out the data according to the need.
- Admin can view the recent actions of all the actors which are included in software.
- Admin can give the edit access to any other user.
- The admin can create a new database as and when required.
- Admin has right to delete the trivial content.

Chapter 3

Database

3.1 Professor

The following are the relational databases for professors:

- Professor Login
  - Professor Username (Primary Key)
  - Professor Password
  - Professor Name
- Professor Time Table
  - Professor Username (Primary Key)
  - Lecture Start Time
  - Lecture End Time
  - Day
  - Class Name
  - Lecture Subject

3.2 Faculty Advisor

The following are the relational databases for professors:

- Facult Advisor Login
  - FA Username (Primary Key)
  - FA Password
  - FA Name

3.3 Student

The following are the relational databases for students:

- Student Login
  - Student Roll Number (Primary Key)
  - Student Name
  - Student Branch
  - Student Lab Batch
  - Student Tutorial Batch
  - Student Semester
  - Student Username (Unique)
  - Student Password
  - Student Passport Format Photo
- Student Attendance
  - Attendance Date (Primary Key)
  - Student Roll Number (Primary Key)
  - Attendance Status
  - Professor Username
  - Lecture Subject (Primary Key)
  - Lecture Type

- Student Time Table
  - Student Batch
  - Time
  - Day
  - Lecture Subject
- Student Subjects
  - Student Branch
  - Student Semester
  - Lecture Subject
  - Lecture Type

3.4 Admin

The following is the relational database for admins:

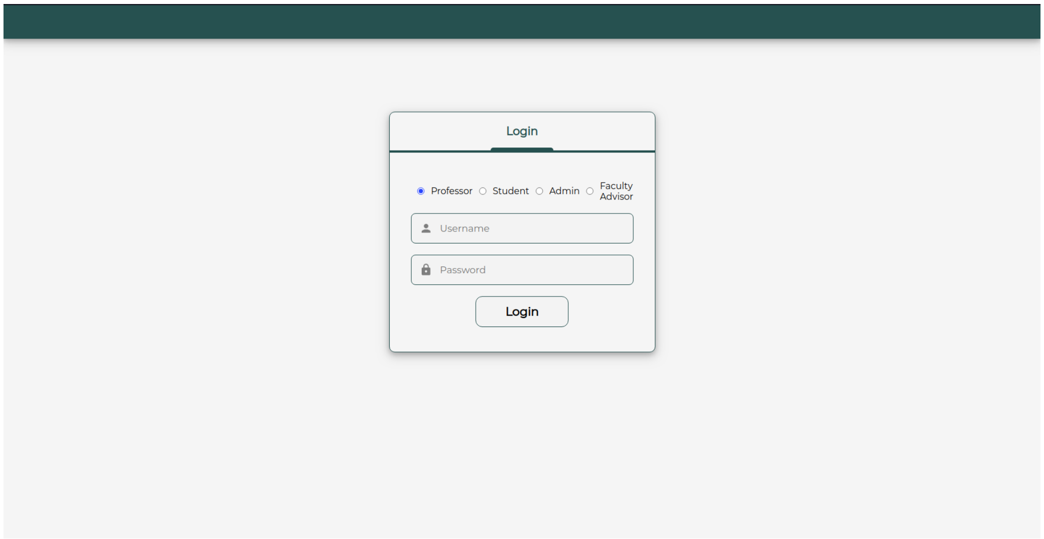
- Admin Login
  - Admin Username (Primary Key)
  - Admin Password
  - Admin Name

Chapter 4

GUI

4.1 Login and Sign Out

The login page has a form asking for username and password and the user's role in the university(Professor, Student or Admin) from the user. When the user is login, cookies are stored in the browser and so no other user can use the same browser to login. The life of cookies is 2 months
On signing out, the cookies deleted. The sign-out button is given at the top-right corner of any webpage of the website.



● Professor ○ Student ○ Admin ○ Faculty Advisor

Figure 4.1: Login Page

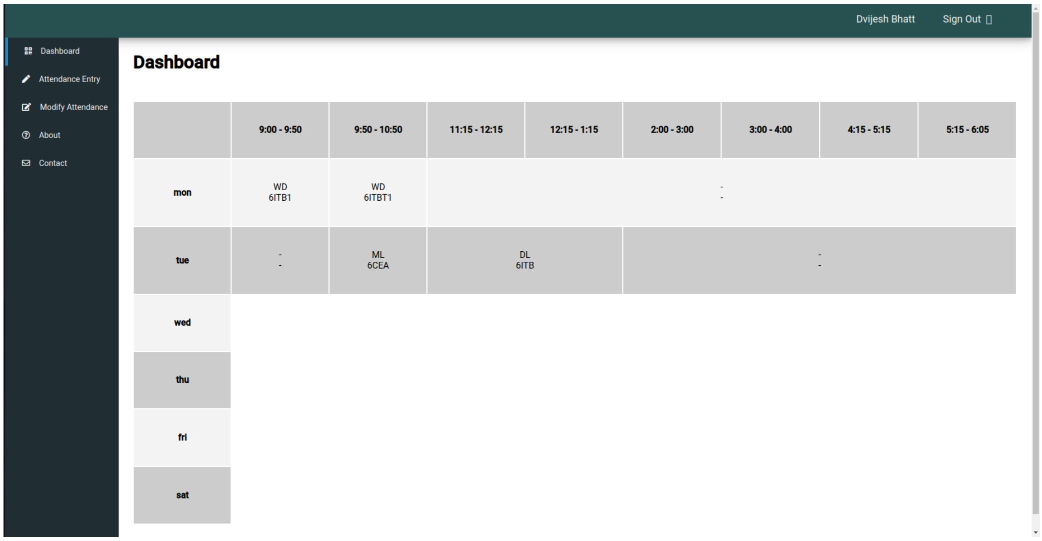
4.2 Professor

The home page contains the professor dashboard which has a professor's timetable. The timetable is interactive and responsive. By clicking on the subject of the timetable slot, the professor is taken to "Take Attendance" page. This page has details filled out as the details in the timetable slot that was selected. By clicking the "Take Attendance" button, the camera will start and take attendance for 10 seconds. After this, the attendance is posted and the professor is redirected to the dashboard. There is a side navigation bar which has options of:

- Dashboard
- Take Attendance
- Modify Attendance
- About
- Contact

By clicking on "Take Attendance" option in the side navigation bar, the professor is directed to the page that has subject and class options to be edited. After entering these requirements, professor can take attendance of proxy lecture.

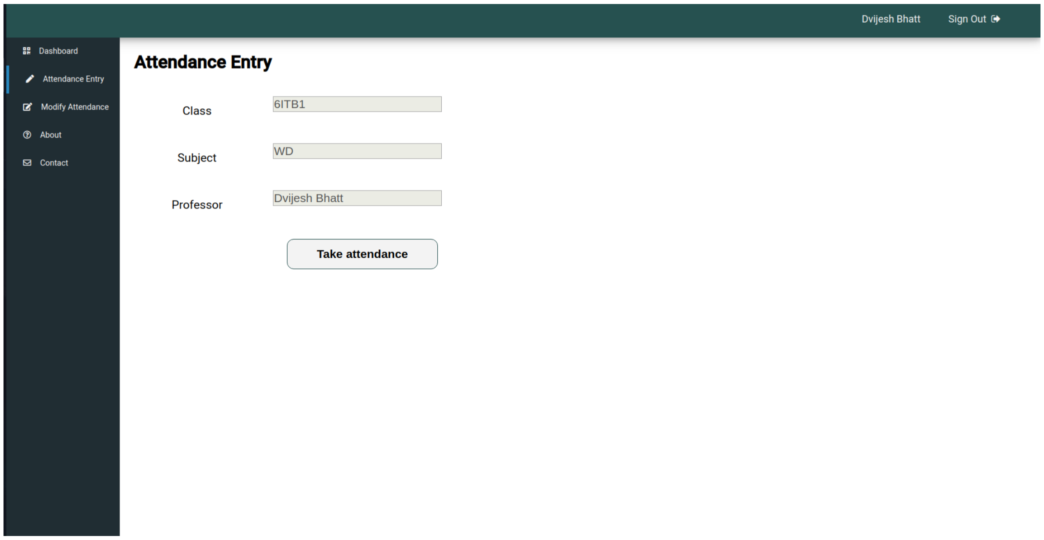
By clicking on "Modify Attendance" option in the professor is redirected to a page where the professor is asked to enter date, subject and class name to search the absentees on that date for that subject from that class. This will show the list below in that page in the form of checkbox and the attendance of desired student can be modified.



Dashboard

	9:00 - 9:50	9:50 - 10:50	11:15 - 12:15	12:15 - 1:15	2:00 - 3:00	3:00 - 4:00	4:15 - 5:15	5:15 - 6:05
mon	WD 6ITB1	WD 6ITBT1	- -					
tue	- -	ML 6CEA	DL 6ITB		- -			
wed								
thu								
fri								
sat								

Figure 4.2: Professor Home Page



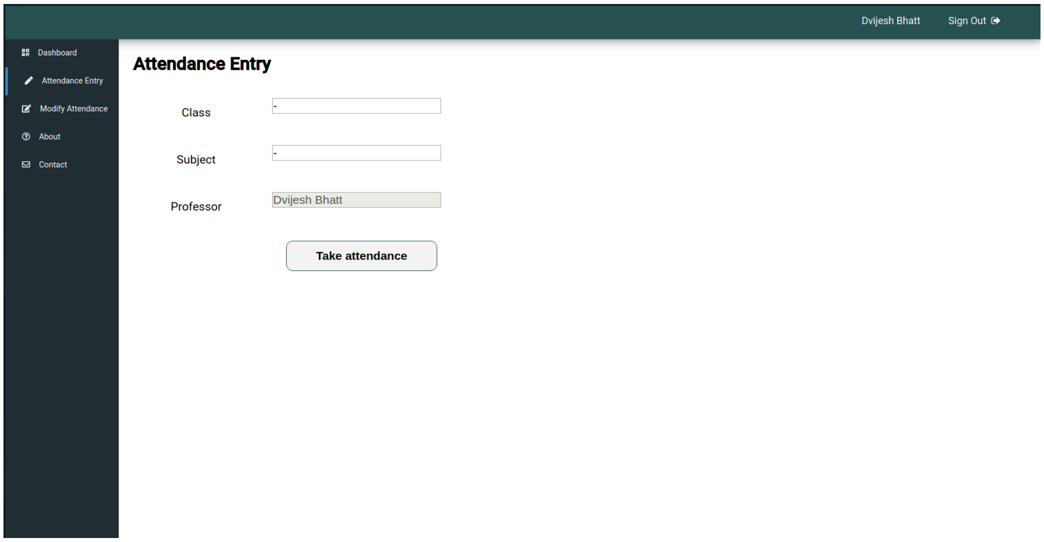
Attendance Entry

Class 6ITB1

Subject WD

Professor Dvijesh Bhatt

Figure 4.3: Take Attendance From Time Table



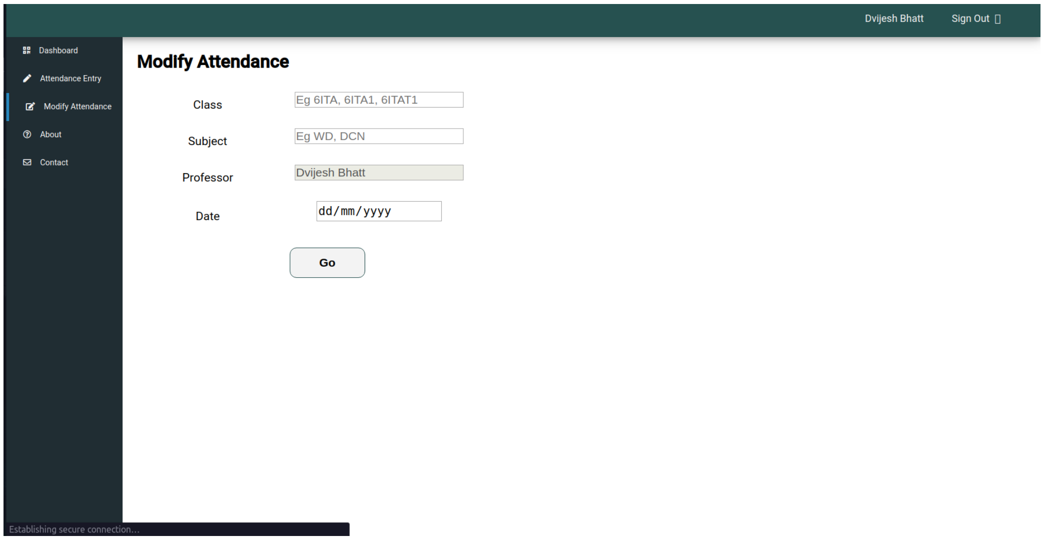
Attendance Entry

Class -

Subject -

Professor Dvijesh Bhatt

Figure 4.4: Take Attendance For Proxy



Modify Attendance

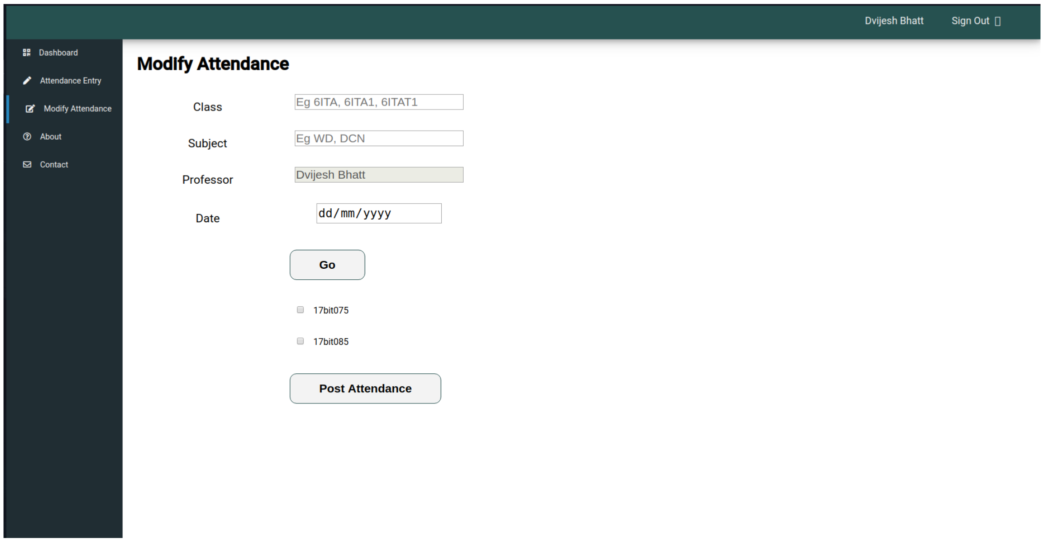
Class Eg 6ITA, 6ITA1, 6ITAT1

Subject Eg WD, DCN

Professor Dvijesh Bhatt

Date dd/mm/yyyy

Figure 4.5: Modify Attendance Part 1



Modify Attendance

Class: Eg 6ITA, 6ITA1, 6ITAT1

Subject: Eg WD, DCN

Professor: Dvijesh Bhatt

Date: dd/mm/yyyy

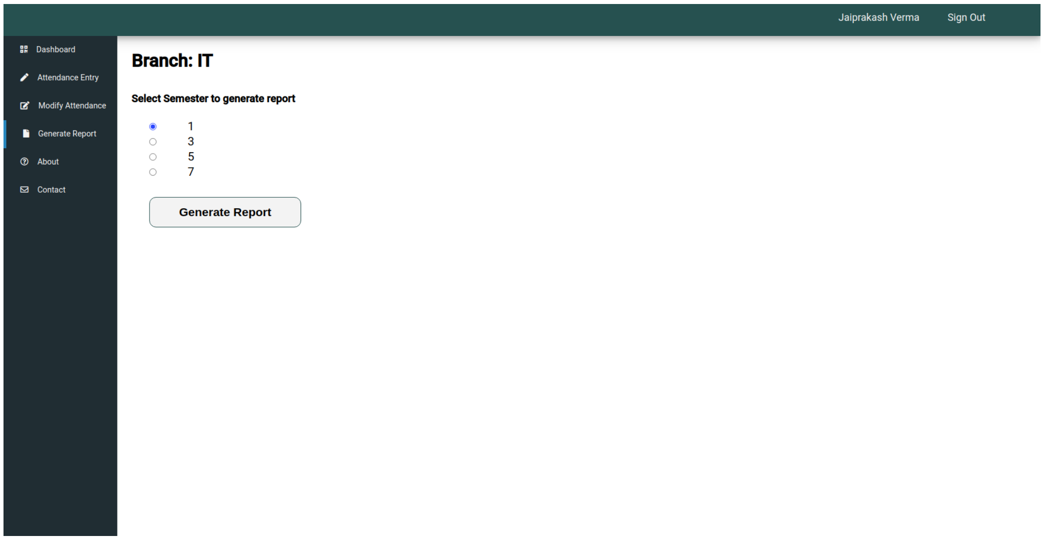
□ 17bit075

□ 17bit085

Figure 4.6: Modify Attendance Part 2

4.3 Faculty Advisor

The home page contains the dashboard just like that of the professor. As we know that the faculty advisor differs from professor in just one functionality i.e. review report generation. So all the functionalities of faculty advisor are same as professor and one more functionality is added i.e. report generation. The report generation page shows the semesters according to time and as the branch of faculty advisor is known, the report is generated as $branch - semester.xls$. The excel file has 3 sheets: Group B, Group C and Group D indicating 75% to 85%, 65% to 75% and below 65%. The file is then sent for the user to download.



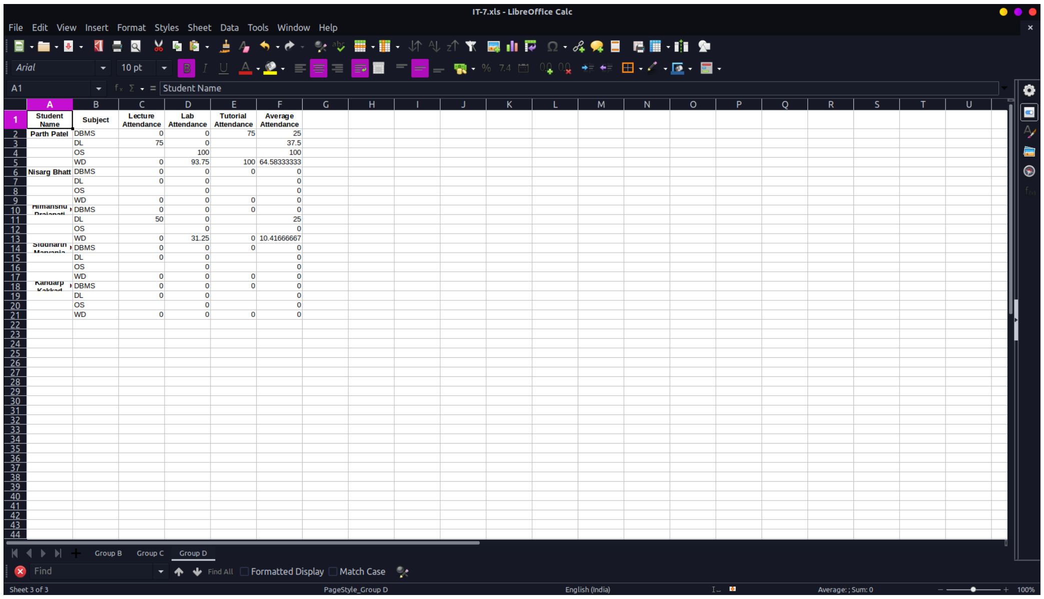
Branch: IT

Select Semester to generate report

- ● 1
- ○ 3
- ○ 5
- ○ 7

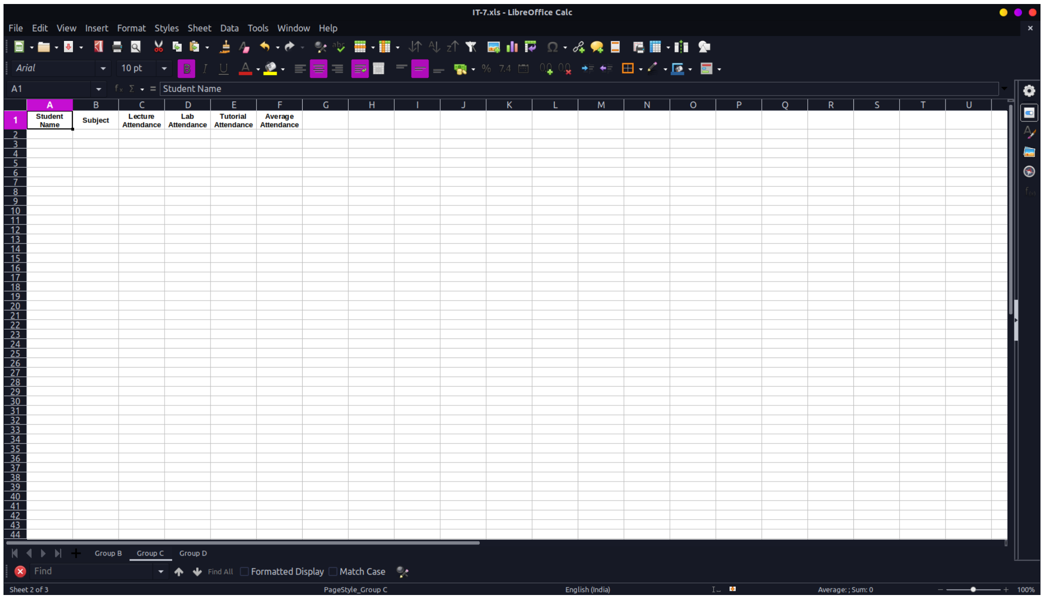
Generate Report

Figure 4.7: Generate Report Page



Student Name	Subject	Lecture Attendance	Lab Attendance	Tutorial Attendance	Average Attendance
Parth Patel	DBMS	0	0	75	25
	DL	75	0		37.5
	OS		100		100
	WD	0	93.75	100	64.58333333
Nisarg Bhatt	DBMS	0	0	0	0
	DL	0	0		0
	OS		0		0
	WD	0	0	0	0
Himanshu Prajapati	DBMS	0	0	0	0
	DL	50	0		25
	OS		0		0
	WD	0	31.25	0	10.41666667
Siddharth Marvania	DBMS	0	0	0	0
	DL	0	0		0
	OS		0		0
	WD	0	0	0	0
Kandarp Kakkad	DBMS	0	0	0	0
	DL	0	0		0
	OS		0		0
	WD	0	0	0	0

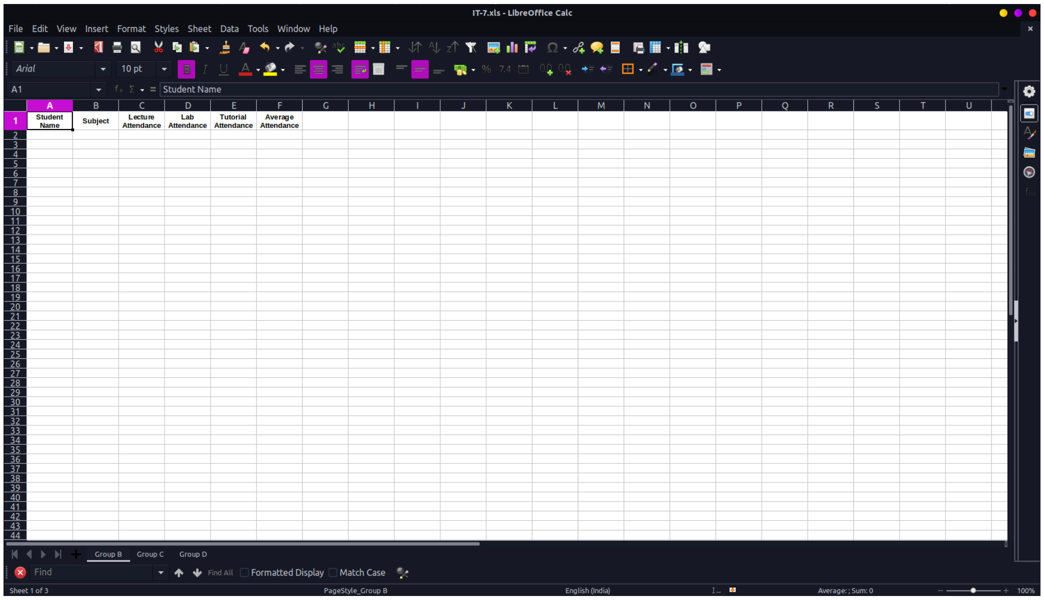
Figure 4.8: Group D



Student Name

Student Name	Subject	Lecture Attendance	Lab Attendance	Tutorial Attendance	Average Attendance

Figure 4.9: Group C



Student Name

Student Name	Subject	Lecture Attendance	Lab Attendance	Tutorial Attendance	Average Attendance

Figure 4.10: Group B

4.4 Student

The home page contains the student dashboard which has a student's attendance graph. The attendance graph is interactive and responsive. By keeping a mouse pointer on the graph, the student would be able to see the attendance of that particular subject. There are four graphs which show attendance of lecture, lab and tutorial and one graph is for the average attendance. There are four different colours which are used to plot the graph. Green colour states that attendance for that particular subject is above 85%. Yellow colour states that attendance is between 75% - 85% . Orange colour states that attendance is between 65% -75%. Red colour states that attendance is less than 65%. There is a side navigation bar which has options of:

- Dashboard
- View Attendance
- About
- Contact

By clicking on the "View Attendance" option in the side navigation bar, the student is directed to a page that has subject and lecture type options to be edited. After entering these requirements, the student can view the attendance of that particular subject.

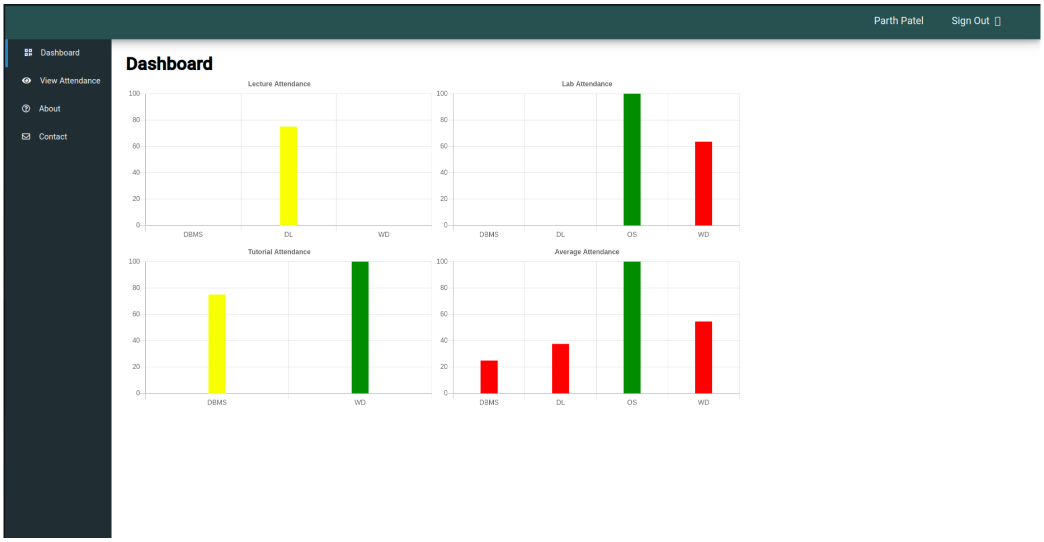
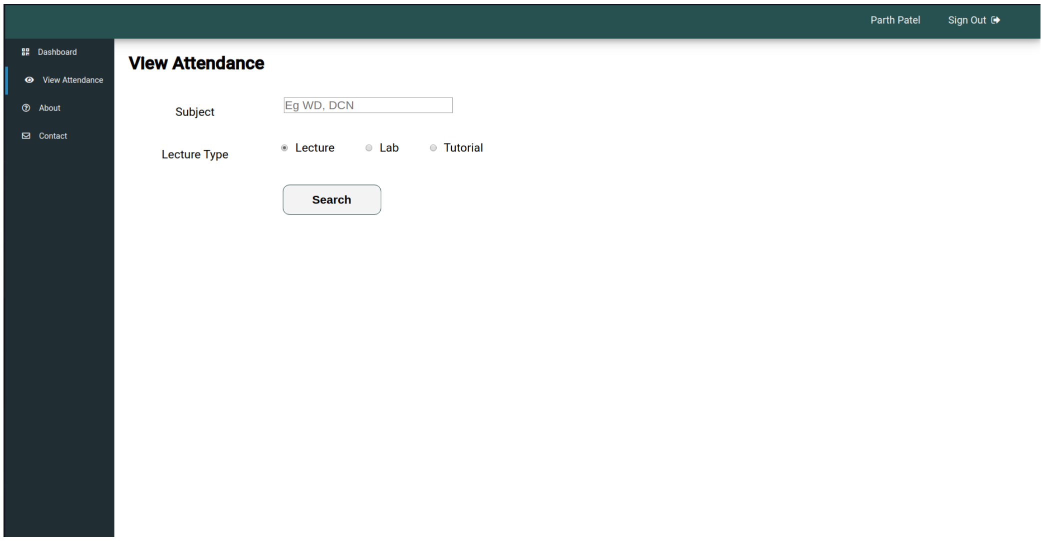


Figure 4.11: Student Dashboard



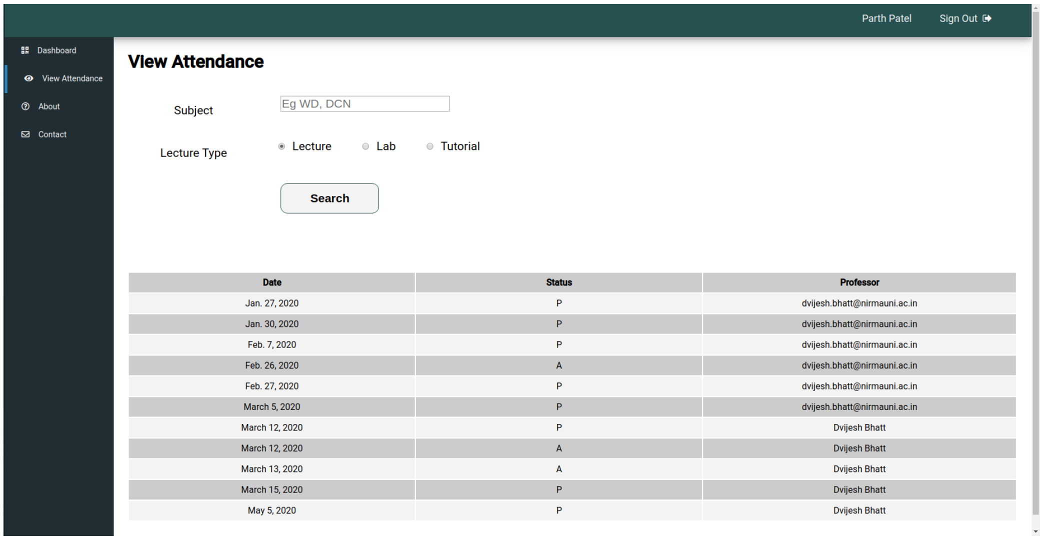
View Attendance

Subject Eg WD, DCN

Lecture Type ● Lecture ○ Lab ○ Tutorial

Search

Figure 4.12: View Attendance Part 1



View Attendance

Subject: Eg WD, DCN

Lecture Type: ● Lecture ○ Lab ○ Tutorial

Date	Status	Professor
Jan. 27, 2020	P	dvijesh.bhatt@nirmauni.ac.in
Jan. 30, 2020	P	dvijesh.bhatt@nirmauni.ac.in
Feb. 7, 2020	P	dvijesh.bhatt@nirmauni.ac.in
Feb. 26, 2020	A	dvijesh.bhatt@nirmauni.ac.in
Feb. 27, 2020	P	dvijesh.bhatt@nirmauni.ac.in
March 5, 2020	P	dvijesh.bhatt@nirmauni.ac.in
March 12, 2020	P	Dvijesh Bhatt
March 12, 2020	A	Dvijesh Bhatt
March 13, 2020	A	Dvijesh Bhatt
March 15, 2020	P	Dvijesh Bhatt
May 5, 2020	P	Dvijesh Bhatt

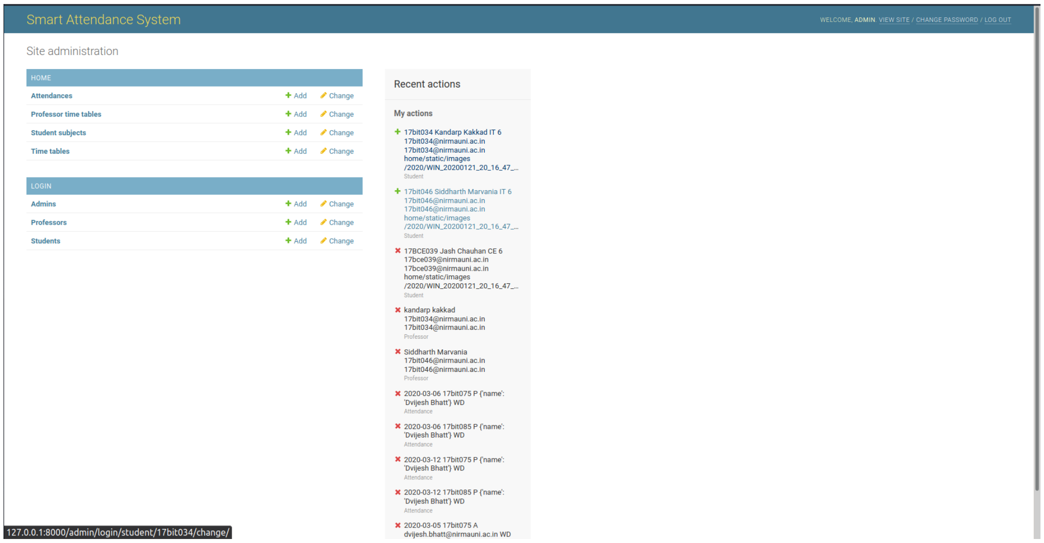
Figure 4.13: View Attendance Part 2

4.5 Admin

The home page contains the name of all the tables or database which are as follows:

- Home
  - Attendances
  - Professor Time Tables
  - Student subjects
  - Time Tables
- Login
  - Professor
  - Student
  - Admin

The home page also contains the recent activity tabs in which all the recent activity performed by the admin is listed along with its detail such as last modified time, about action etc. Admin can view the tables of the database by clicking on it. The tables are displayed in the tabular manner. The tables are supported with suitable filters to efficiently search for particular records. The admin has privilege to add, delete and modify the content. So for that also different pages are there. Edit page contains forms which ask for certain information. Modify page is almost same as edit page. For delete there is a small button against each record, to delete a particular record we just need to select that button.


Smart Attendance System
WELCOME, ADMIN. VIEW SITE / CHANGE PASSWORD / LOG OUT
127.0.0.1:8000/admin/login/student/17bit034/change/

Site administration

HOME		
Attendances	+ Add	Change
Professor time tables	+ Add	Change
Student subjects	+ Add	Change
Time tables	+ Add	Change

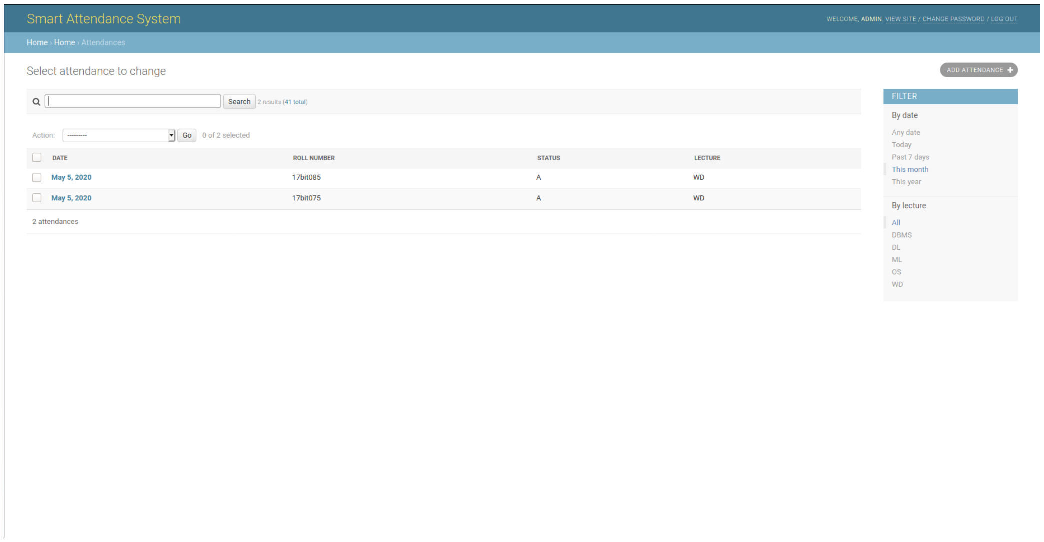
LOGIN		
Admins	+ Add	Change
Professors	+ Add	Change
Students	+ Add	Change

Recent actions

My actions

- + 17bit034 Kandarp Kakkad IT 6 17bit034@nirmauni.ac.in 17bit034@nirmauni.ac.in home/static/images /2020/WIN_20200121_20_16_47_...
  Student
- + 17bit046 Siddharth Marvania IT 6 17bit046@nirmauni.ac.in 17bit046@nirmauni.ac.in home/static/images /2020/WIN_20200121_20_16_47_...
  Student
- ✗ 17BCE039 Jash Chauhan CE 6 17bce039@nirmauni.ac.in 17bce039@nirmauni.ac.in home/static/images /2020/WIN_20200121_20_16_47_...
  Student
- ✗ kandarp kakkad 17bit034@nirmauni.ac.in 17bit034@nirmauni.ac.in
  Professor
- ✗ Siddharth Marvania 17bit046@nirmauni.ac.in 17bit046@nirmauni.ac.in
  Professor
- ✗ 2020-03-06 17bit075 P {'name': 'Dvijesh Bhatt'} WD
  Attendance
- ✗ 2020-03-06 17bit085 P {'name': 'Dvijesh Bhatt'} WD
  Attendance
- ✗ 2020-03-12 17bit075 P {'name': 'Dvijesh Bhatt'} WD
  Attendance
- ✗ 2020-03-12 17bit085 P {'name': 'Dvijesh Bhatt'} WD
  Attendance
- ✗ 2020-03-05 17bit075 A dvijesh.bhatt@nirmauni.ac.in WD

Figure 4.14: Admin Home Page



Select attendance to change

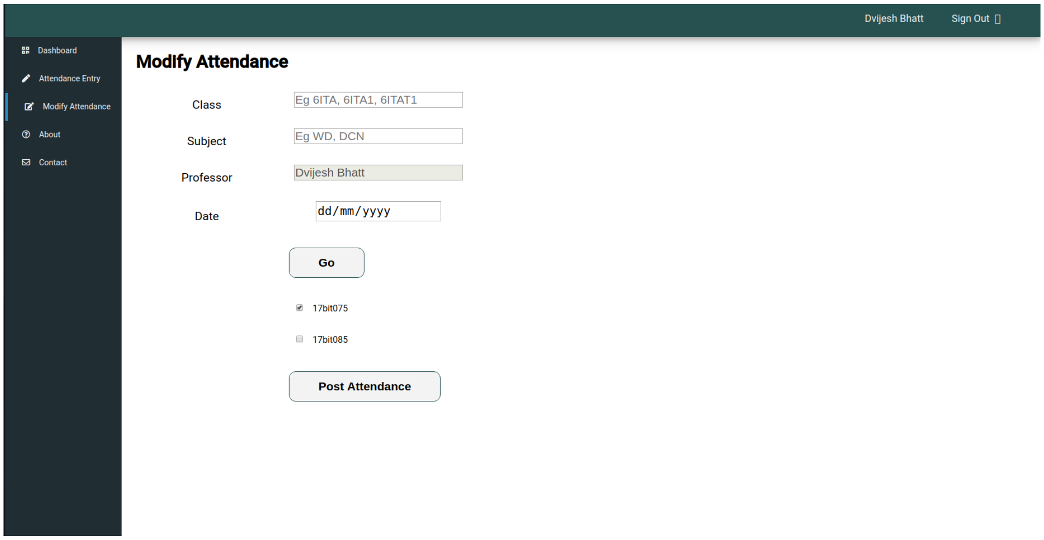
2 results (41 total)

Action: --------- 0 of 2 selected

□	DATE	ROLL NUMBER	STATUS	LECTURE
□	May 5, 2020	17bit085	A	WD
□	May 5, 2020	17bit075	A	WD

2 attendances

Figure 4.15: Previous To Modification



Modify Attendance

Class: Eg 6ITA, 6ITA1, 6ITAT1

Subject: Eg WD, DCN

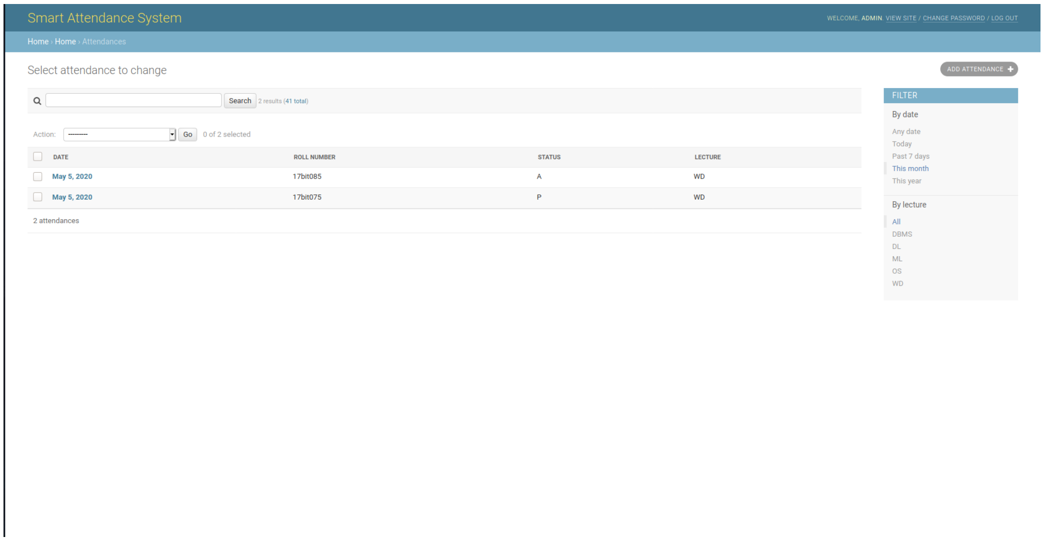
Professor: Dvijesh Bhatt

Date: dd/mm/yyyy

☑ 17bit075

□ 17bit085

Figure 4.16: Modification By Professor



Smart Attendance System

WELCOME, ADMIN. VIEW SITE / CHANGE PASSWORD / LOG OUT

Home › Home › Attendances

Select attendance to change

Search 2 results (41 total)

Action: --------- Go 0 of 2 selected

□	DATE	ROLL NUMBER	STATUS	LECTURE
□	May 5, 2020	17bit085	A	WD
□	May 5, 2020	17bit075	P	WD

2 attendances

Figure 4.17: After Modification

Chapter 5

Deep Learning Module

5.1 General

In this application deep learning module takes video recording of class room and provides list of the student present in the class. It majorly consists of 3 modules:

1. Preprocessing Module
2. Face detection Module
3. Face Verification Module

5.2 Preprocessing Module

These module takes video as input and divide it into frames of predefined size so that it is easy for the second module to do it's task otherwise different videos with different resolution are prone to errors or system failure. And it extracts 10 frames form the video recording of around 10 minute so that system overhead can be reduced. After selecting which frame will be processed that frames will be stored to specific folder where second module can find it and use it for detection.

5.3 Face Detection Module

This is basically deep learning model which takes image as input and provides faces which are presented in input image as separate image. The output images are of size (112 X 112) which is available in specified folder for the use of third module.
The deep learning model which use in this module is known as “MTCNN”[1]. “MTCNN” stands for “Multi-task Cascaded Convolutional Networks”. As the name suggests 3 Convolutional Network are Cascaded to form “MTCNN”. Input image is first scaled to different sizes to create image pyramid and then fed to the first Network.
The name of the networks are:

- Proposal Network (P-Net): This Network is used to obtain the candidates boxes vector and then non-maximum suppression (NMS) is used to merge the overlapped candidate boxes.

- Refine Network (R-Net): Output of the P-Net to fed as input to the R-Net which discard most of the false boxes and calibrates with bounding box regression [2] and then NMS.

- O-Net: This Net use output of R-Net as input and gives final output prediction with five face landmarks.

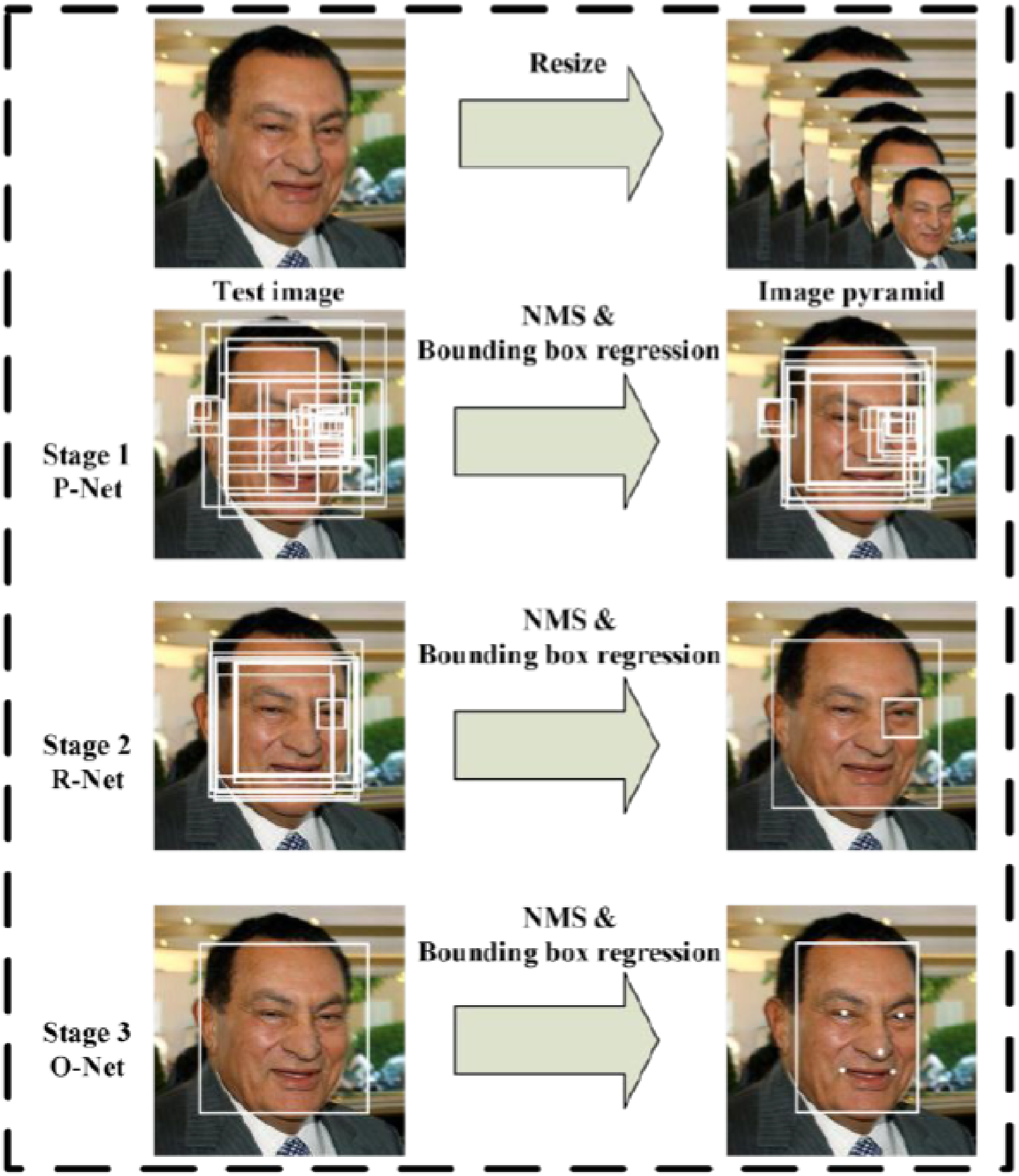


Figure 5.1: Data flow of MTCNN as passes through 3 subnets of MTCNN

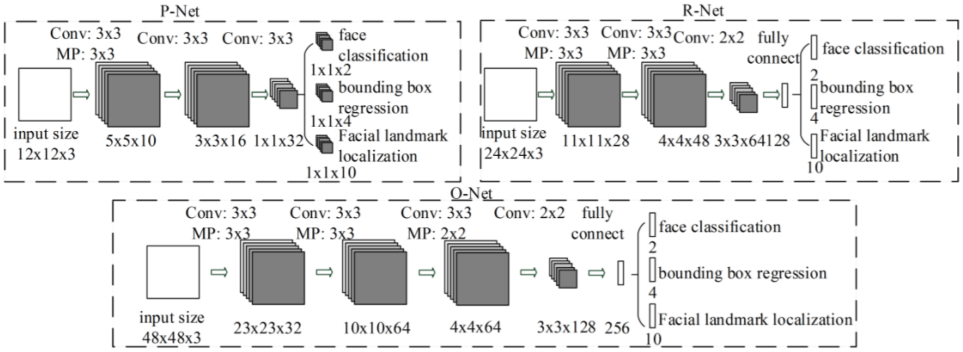


Figure 5.2: MTCNN Architecture (P-Net, R-Net and O-Net) [3]

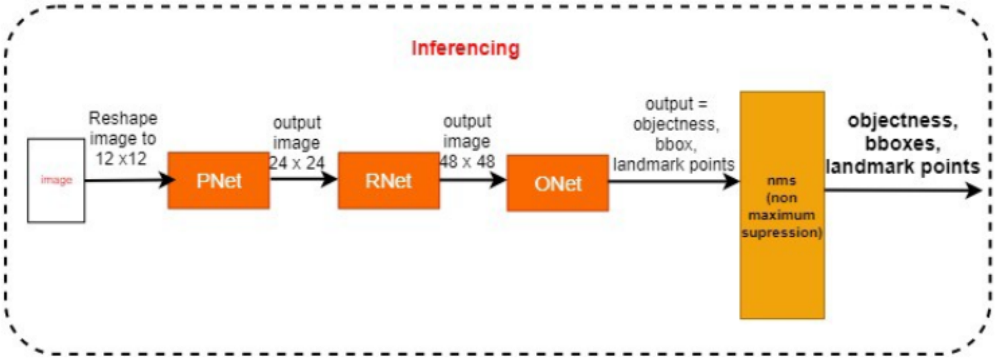

Inferencing
image
Reshape image to 12 x12
PNet
output image 24 x 24
RNet
output image 48 x 48
ONet
output = objectness, bbox, landmark points
nms (non maximum supression)
objectness, bboxes, landmark points

Figure 5.3: Input and output of different module of MTCNN with its internal structure [3]

5.4 Face Verification Module

To perform face verification task there is a well known algorithm named “FaceNet” which is known to be “State of Art” algorithm for face verification task. But in our application we need to verify around 80 – 150 students at a time for one class, But maybe there is a good chances that we have to do it for multiple classes simultaneously which impose very high load on server and it maybe leads to crash system.

So we need some other model which has same accuracy in Face verification with none to little degradation and lighter than “FaceNet” in terms of computation. Thus we used “SeesawFacenet” [4] which is designed to be used on

mobile platform and it is also robust in terms of face verification task.
There are two types of block structure which are used in this Network:

1. Seesaw-shuffle block
2. Seesaw-share block

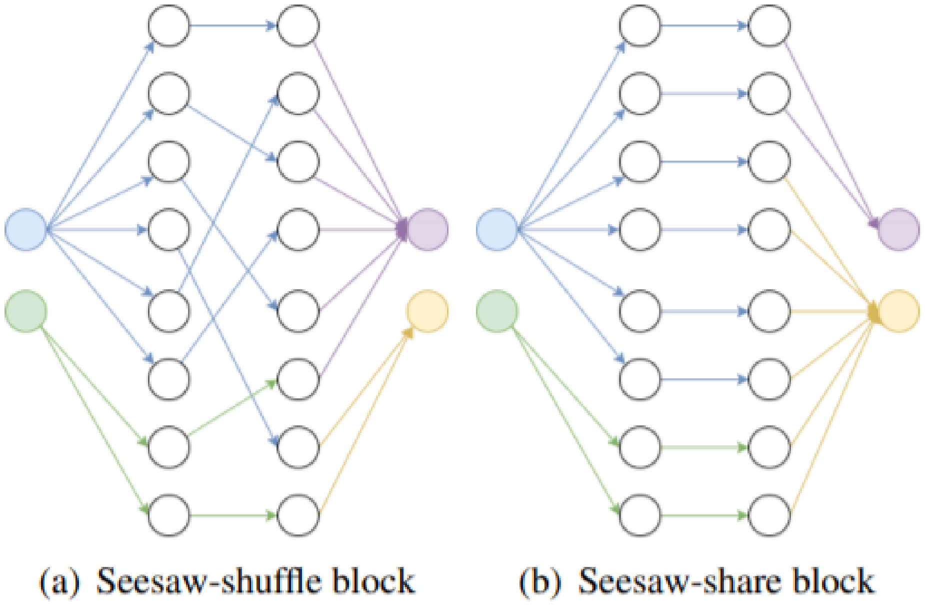


Figure 5.4: Structure of Seesaw-shuffle block and Seesaw-share block [4]

These blocks are responsible for Faster computation , high accuracy and light weight in terms of total parameters.
We trained this model on the dataset that we have created by collecting one image of each student in our class and one image from the video recording of CCTV camera. So that model can identify images when it is deployed. We get around 75 – 80% accuracy when we tested our model. This module takes images one by one form the folder where the previous module has saved it and compare it with the data set and return the list of the student whose images are stored in the folder by previous module.

Chapter 6

Deployment

6.1 Installing Dependencies

First we need to create an account on Heroku. Then we need to install Heroku ToolBelt. ToolBelt is actually a command line tool to manage Heroku apps. After installation is complete we need to login to our account using terminal. $*heroku login*

6.2 Preparing the Application

The process of Deployment is actually done through Git. The project would be stored in Git repository in Heroku cloud. Following things need to be added to the project:

- In root add Procfile
- In root add requirements.txt file which contains all the requirements.
- In requirements.txt add Gunicorn.
- A runtime.txt which would specify the correct python version.
- We need to configure whitenoise to server static files.

6.3 Procfile and runtime.txt

We need to create a file called Procfile in the root of the project with the following command:

$web : gunicornsas.wsgi$
Also we need to create runtime.txt file to specify the version of python. The content of the file is just 1 line:

$python3.8$

6.4 requirements.txt

If we are using pip and virtual environment we can run
$pip\ freeze\ >\ requirements.txt$

Requirements.txt contains following :

$Django == 1.9.8$
$dj - database - url == 0.3.0$
$dj - static == 0.0.6$
$gunicorn == 19.6.0$
$Unipath == 1.0$
$python - decouple == 3$
$Pillow == 3.3.0$
$Markdown == 2.6.6$
$bleach == 1.4.3$
$psycopg2 == 2.6.1$
$whitenoise == 3.2$

6.5 Setting Static Assets

We need to configure settings related parameter in settings.py. Then install Whitenoise and add it to wsgi.py file.

```
import os
from django.core.wsgi import get_wsgi_application
from whitenoise.django import DjangoWhiteNoise

os.environ.setdefault(”DJANGO_SETTINGS_MODULE”,”sas.settings”)

application = get_wsgi_application()
application = DjangoWhiteNoise(application)
```

Then update settings.py

```
STATICFILES_STORAGE = 'whitenoise.django.GzipManifestStaticFilesStorage'
```

6.6 Final Steps

First we need to clone the git repository of our project. Then we need to login to heroku through toolbelt. Then create Heroku app in project root. Then add postgre sql database to our app. Then we need to login to our Heroku. Go to Reveal config vars in settings menu. There we need to add all the environment variables. Then we can push to deploy.

Chapter 7

Future Prospects

7.1 General

After completion, project can be extended to provide functionalities like detecting prohibited activities and requesting leave applications from students with sub-par attendance.

Bibliography

[1] "*Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks*": Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, Senior Member, IEEE, and Yu Qiao, Senior Member, IEEE

[2] "*Multi-view face detection using deep convolutional neural networks*": S. S. Farfade, M. J. Saberian, and L. J. Li

[3] "*SeesawFaceNets: sparse and robust face verification model for mobile platform*": Jintao Zhang, Di Di Chuxing

Appendix

- https://medium.com/@satya15july_11937/facial-landmark-detection-using-mtcnn-f7afa4f3818c
- https://docs.djangoproject.com/en/dev/contents/
- https://www.youtube.com/watch?v=qgGIqRFvFFk&list=PL6gx4Cwl9DGBlmzzFcLgDhKTTfNLfX1IK
- https://www.youtube.com/watch?v=n-FTlQ7Djqc&list=PL4cUxeGkcC9ib4HsrXEYpQnTOTZE1x0uc