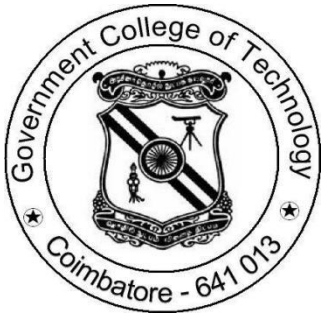
ATTENDANCE FOR GMEET USING SCREEN AND FACE TIME



PROJECT SUBMITTED IN PARTIAL FULFILLMENT OF THE

REQUIREMENTS FOR THE AWARD OF THE DEGREE OF

BACHELOR OF ENGINEERING IN **COMPUTER SCIENCE**

AND ENGINEERING OF THE ANNA UNIVERSITY

MINI PROJECT WORK

Submitted by

LALITH KUMAR M
1817129

MAHESHWARAN P
1817130

SANJAYKUMAR V
1817142

SANKARKUMAR K
1817L06

2022

Under the Guidance of

Dr. S.RATHI M.E., Ph.D.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
GOVERNMENT COLLEGE OF TECHNOLOGY
(An Autonomous Institution Affiliated to Anna University)

COIMBATORE-641013

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GOVERNMENT COLLEGE OF TECHNOLOGY

(An Autonomous Institution Affiliated to Anna University)

COIMBATORE - 641 013

MINI PROJECT WORK

JAN-2022

This is to certify that this project work entitled

ATTENDANCE FOR GMEET USING SCREEN AND FACE TIME

is the bonafide record of project work done by

LALITH KUMAR M
1817129

MAHESHWARAN P
1817130

SANJAYKUMAR V
1817142

SANKARKUMAR K
1817L06

Of B.E./B.Tech. (COMPUTER SCIENCE AND ENGINEERING) during the academic year 2021 - 2022

Project Guide	Head of the department
Dr.S.RATHI M.E., Ph.D.	**Dr.J.C.MIRACLIN JOYCE PAMILA M.E., Ph.D.**

Submitted for the project viva-voice examination held on _____

Internal Examiner	External Examiner

ACKNOWLEDGEMENTS

We express our sincere gratitude to **Dr.P.THAMARAI, M.E., Ph.D.,** Principal, Government College of Technology,Coimbatore for providing us all facilities that we needed for the completion of this project.

We whole-heartedly express our thankfulness and gratitude to **Dr.J.C.MIRACLIN JOYCE PAMILA, M.E., Ph.D.,** Professor and Head of the Department of Computer Science and Engineering, Government College of Technology, for helping us to successfully carry out this project.

Our thankfulness and gratitude to our respectable project guide **Dr.S.RATHI, M.E., Ph.D.,** Professor of the Department of Computer Science and Engineering and the panel members **Dr.R.MUTHURAM, M.E., Ph.D.,** Assistant Professor and **Prof.N.RAMYA, M.E.,** Assistant Professor who has been an immense help through the various phases of the project. With her potent ideas and excellent guidance, we were able to comprehend the essential aspects involved.

We extend our sincere thanks **to Dr.K.KUMAR, M.E., Ph.D.,** Associate Professor , **Prof.T.RAJA SENBAGAM, M.E**., Assistant Professor, **Dr.A.MEENA KOWSHALYA, M.E., Ph.D.,** Assistant Professor, **Dr.R.BHAVANI, M.E., Ph.D.,** Assistant Professor **and Prof.L.SUMATHI, M.E.,** Assistant Professor for all their valuable suggestions to the completion of the project.

We thank all the non-teaching staff and our friends for their cooperation towards the successful completion of the project. We would like to dedicate the work to our parents for their constant encouragement throughout the project

SYNOPSIS

If the Google Workspace (or Google Workspace Essentials) is used, Google Meet could be used to organize the video meetings and live streaming to compliment the new normal.

If the meeting had more than a few attendees, meeting host might want to know who attended the Google Meeting (or live stream) and how long.

Though Google Meet records attendance but this information is provided in Google Admin Console, and one needs Google Workspace Admin access to see it, which is not scalable and don’t want to give Google Workspace admin access to all users.

So thought of investing sometime and create a self-service solution where the Google Workspace users (without admin access) can simply fill up a Google Form, and instantly get their Google Meet attendance report.

Every solution comes with a certain set of features and limitations, and this meet attendance solution is no different, so before demonstration users should know how the solution works, and how easily set it up for the Google Workspace users.

TABLE OF CONTENTS

LIST OF ABBREVATIONS

API	- Application Program Interfaces
AWS EC2	- Amazon Web Services Elastic Compute Cloud
WINSCP	- Windows Secure Copy
HTTP	- Hypertext Transfer Protocol
DRF	- Django Rest Framework
REST	- Representational State Transfer
SSH	- Secure Shell
TCP	-Transmission Control Protocol
LBP	- Local Binary Patterns
JWT	- Json Web Token

CHAPTER 1

INTRODUCTION

1.1 DESCRIPTION

The purpose of the model is to provide attendance for students using face and screen time. The model is proposed in both web and app. The model is provided for both staff and students. The staff should register as staff in the website/app and a profile page will be created as staff consisting details of courses and no of students enrolled in them

The students should register as students in the website/app and a profile page will be created as student consisting details of courses and staff in charge. Before the class starts, the staff will provide a link to the students and ask them to join. The students join the class through the link provided. During the class the video of students are switched on in order to calculate the face time of the students.

To avoid students from skipping classes by switching other tabs, the screen time is also calculated. If the student tries to switch tab a warning message will be displayed by the server.

The calculated face time and screen time data will be sent to the server where the server in turn return the data to the staff in charge of the particular course.

The staff will provide attendance based on calculated face and screen time of the students in graphical representation. The students can also watch their attendance in graphical representation in their profile page. The main motto of the model is to make students to focus fully on studies instead of other unnecessary activities in online.

1.2 EXISTING SYSTEM

There are extensions that can record attendance for Gmeet, but they use web scraping to identify the leaving and joining time. Using those user can mark the attendance based on the joining time of the persons (students).

The issue in that is user cannot detect whether the meet is foreground or background. And cannot detect whether the student is in front of screen or not And a Single person can login with different account on Gmeet so that even a single student can get attendance for all his friends by just joining multiple accounts.

1.3 PROBLEM DEFINITION

Attendance for Gmeet usually does not consider whether the Gmeet is in foreground and whether the users is in front of the Gmeet, so that the users can use other applications while attending the Gmeet.

The problem arises because Gmeet Attendence Trackers does not take into account that whether the users is fully engaged with the Gmeet (not using other applications and user is in front of the Gmeet by accessing the camera).

The Gmeet Attendance Trackers will provide the attendance without noticing whether the users is in front of screen by simply tracking the joining time of the users.

1.4 PROPOSED SYSTEM

Aims to Enhance the listening time of the student during the online class by Restricting tab switch, screen-time, face-time

Tabs Switch: To keep the students in the class meeting tab,

Screen-Time: To restrict a user from using other applications,

Face-Time: To engage the student with a screen.

1.5 ORGANIZATION OF THE PROJECT

Literature reviews of already existing proposals are discussed in chapter2.

Chapter3 has system specification which tells about the software requirements.

Chapter 4 discusses the overall project and design which tells the brief description of each of the modules in this project.

Chapter 5 has the implementation and experimental result of the project.

Chapter 6 deals with the conclusion and future work.

Finally chapter 7 deals with the references

CHAPTER 2

LITERATURE REVIEW

2.1 FACE DETECTION USING CASCADING ALGORITHM:

Vision-based automated systems applied to face recognition can be divided into 4 steps: face detection, image pre-processing, feature extraction and matching.

Face detection is a hard task, once faces form a similar class of objects and their features, such as eyes, mouth, nose and chin, have, in general, the same geometrical configuration.

The captured image of the face may be pre-processed to overcome illumination variations.

Advantages:

The key advantage of a Haar-like feature over most other features is its calculation speed. Due to the use of integral images, a Haar-like feature of any size can be calculated in constant time.

Haarcascade has more accuracy than the LBP classifier

Haarcascade classifier detects more number of faces than the LBP classifier in an image.

Disadvantages:

Haarcascades are notoriously prone to false-positives.

2.2 GOOGLE MEET ATTENDANCE TRACKER:

This extension was made specifically for teachers to track the attendance of their students in online class.

It can also be used to have a general overview after the end of the meeting since it can give details like how many attended and how many attended more than 65% duration of the meeting etc.

Advantages:

Total number of persons attended the meeting can be detected. Count of people who attended more than 65% duration of the class can be detected.

Disadvantages:

It does not use the screen time so users can switch the tabs and use other applications.

It uses web scraping so just joining the meeting gives attendance.

2.3 GOOGLE MEET ATTENDANCE COLLECTOR EXTENSION:

The simplest way to take attendance on Google Meet.

This extension is considering the users joining time and leaving time and mark the attendance.

Advantages:

Able to capture participant snapshot. Capture a pic of the in-call attendees. Export the list of active participants in a file. Can mark the attendance quickly.

Track a participant's presence in Google Meet session after every minute, 5 minutes, or 60 minutes.

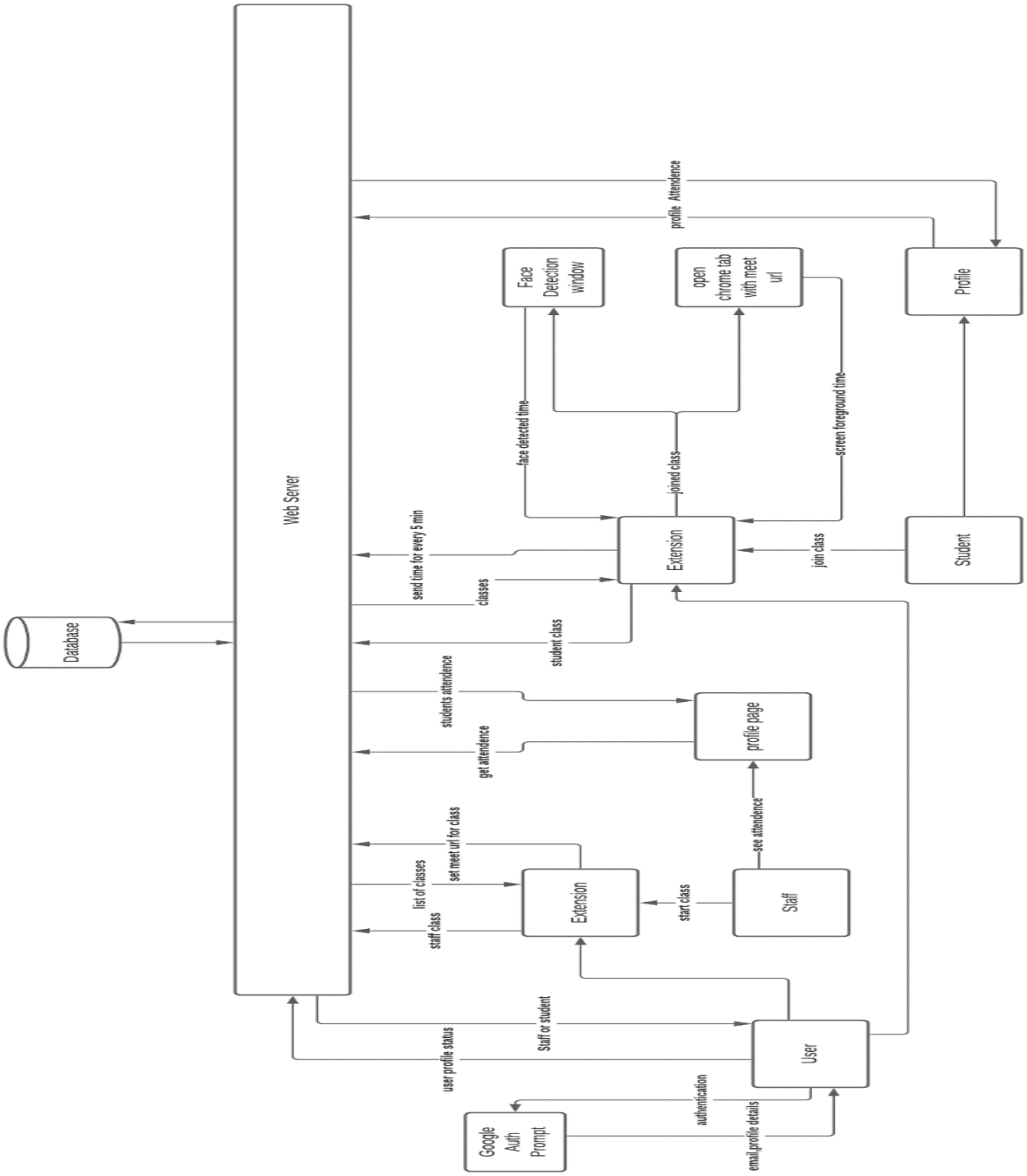
Disadvantage:

It does not consider whether the user is in-front of the meeting and it does not consider whether the meet is foreground or background.

CHAPTER 3

PROJECT DESIGN

3.1 PROJECT ARCHITECTURE



Prompt the google authentication page to the user and encode the user credentials by JWT.
If the email id is not matched, alert the user to create an account on the provided website.

After the successful login, route the user to the staff profile page if the email id registered as staff else navigate the user to the student profile. Display the list of registered courses so that staff can start the class from extension.

Staff needs to choose the class from the list to start the class while picking the class name the extension takes the list of opened tabs in the browser.
If there is any meet link detected send it to the server .If more than one meet link detected alert the staff to close the unwanted meet link.

If no meet link is detected at the browser, alert the staff to open at least one meet link before start the class. Students can join the class from extension itself.

Students need to pick the class name from the list of enrolled classes. If class has not started, notify the student to wait until class starts.

Else fetch the class url from the server and launch it on a new tab. And start the timer to calculate both screen time and face listening time and send it to the server.

3.2 PROJECT WORKFLOW

The proposed system is an ensemble of attendance for gmeet using screen and face time. The extension detecting the face and the user screen time to provide an attendance. The extension declines the one user login.

That is more than one user account in same system and the proposed system would send a user data from client side to server every 5minutes in regular interval and also the user can change tab during class time.

The proposed system will detects this kind of activities and notice to the server and timer won't calculate it for class time.

Advantages

To avoid the cheating in attendance during online class

To maintain genuine attendance.

3.3 MODULES

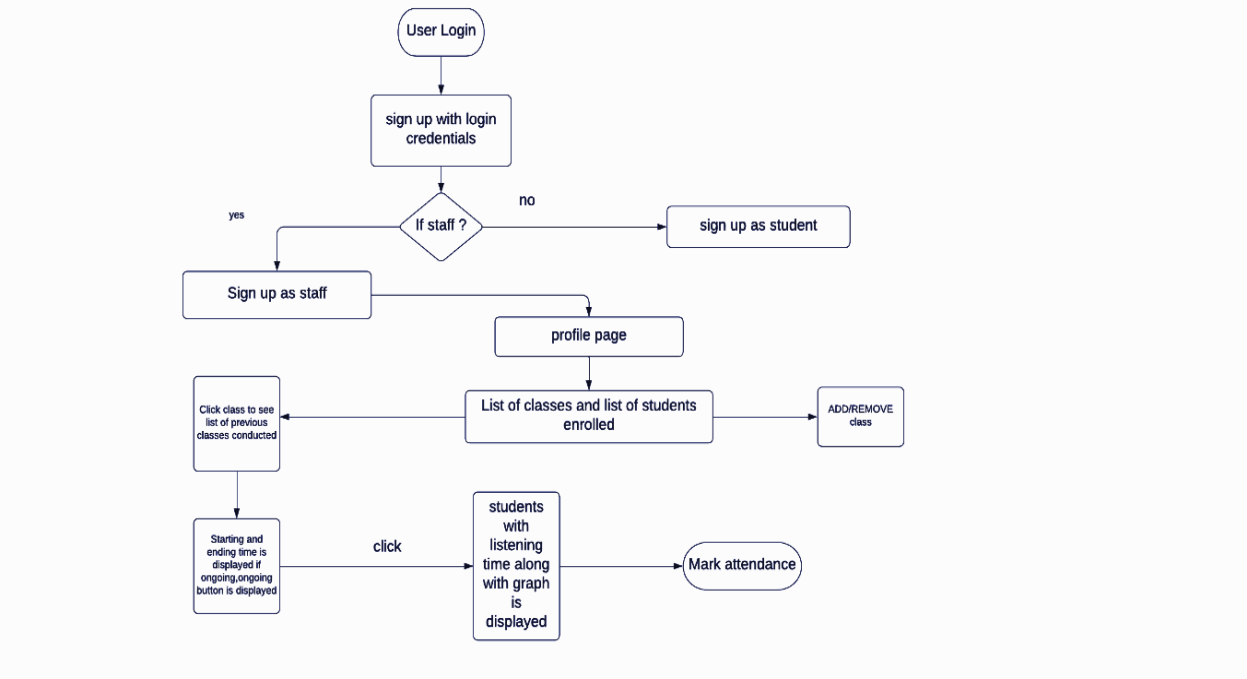
Staff registration in website

Student registration in website

Staff registration in extension

Student registration in extension

3.3.1 STAFF REGISTRATION IN WEBSITE



Staff will sign-up in website using username, email and password.

Staff should register as staff in the website with other login credentials.

Profile page will be created for staff containing list of class and list of students enrolled in that classes.

The Staff can create or delete classes in the profile page.

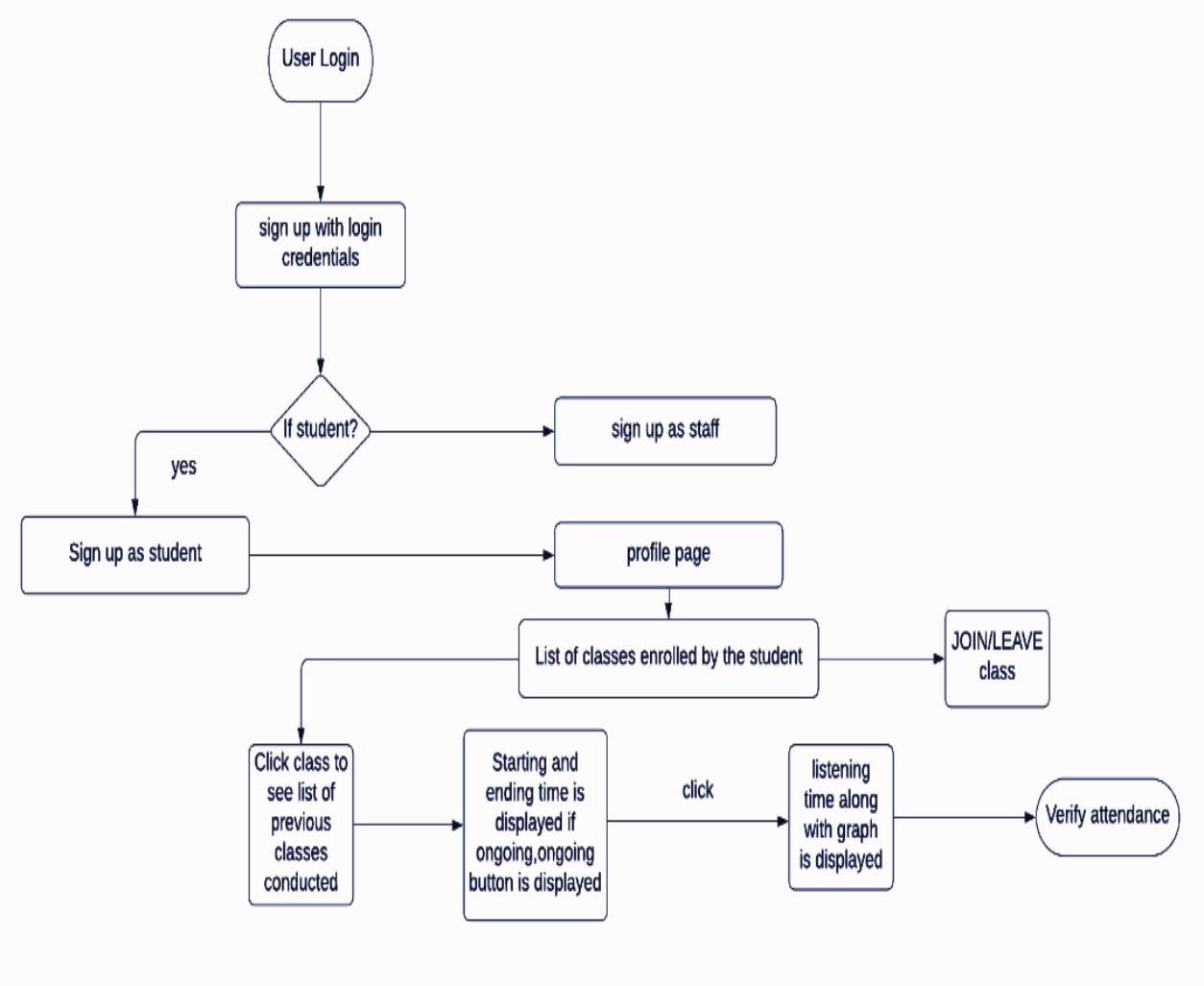
Staff can get the information of previous classes in profile page with starting and ending time of class.

Also, Staff can create URL for meeting where ongoing meets are represented using green light and finished classes are represented using red light.

The profile page consist of attendance time of students with time and graphical representation.

With attendance time along with graphical representation, attendance is marked.

3.3.2 STUDENT REGISTRATION IN WEBSITE



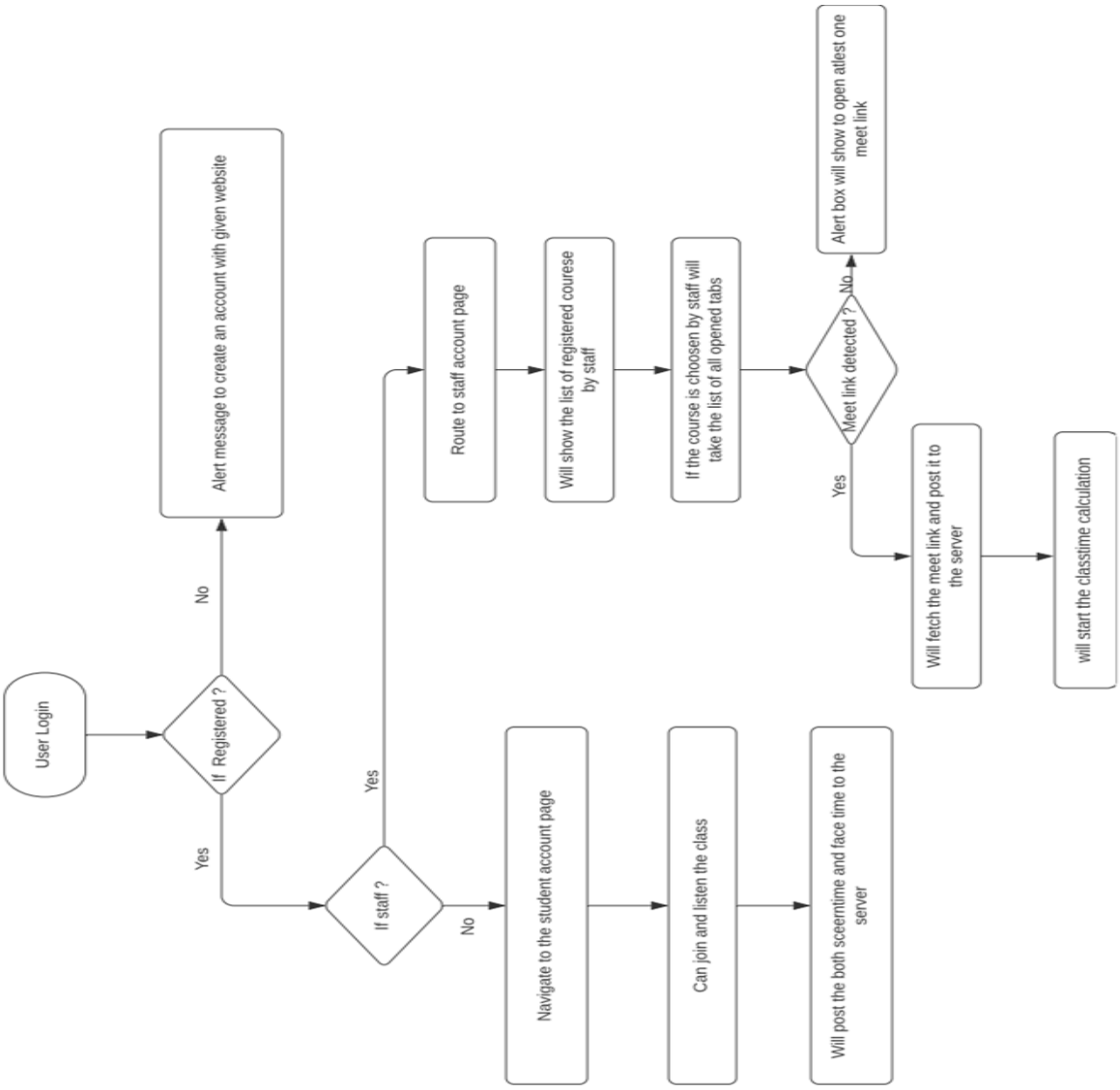
Student will also sign-up in website using username, email and password.

Student should register as student in the website.

Profile page will be created for students which contains the list of classes enrolled by them.

They can see attendance time in their profile.

3.3.3 STAFF SIGN-IN CHROME EXTENSION



Sign in using registered mail id provided to the website.

If mail id is not matched, notify the user to create an account in website.

Sign in, after mail id is matched. The server will response with staff or student profile. If staff, list of classes with buttons containing class names will appear.

For example, if currently a python class, all they need to do is click the button.

The server will check if the meet URL is opened, if there are more than one URL it will alert the staff.

If there is no link then alert staff to open a link. Correct meet link is sent to the server and the timer will start only after the staff enters the meeting.

Class timing is recorded and if the class ended then the end time is provided to the server.

Based on class time, attendance is provided.

3.3.4 STUDENT SIGN-IN CHROME EXTENSION

Sign in using registered mail id provided to the website.

If mail id is not matched, notify need to create account in website.

Sign in, after mail id is matched. The server will response with staff or student profile.

If student list of classes with buttons containing class names will appear. For example, if currently a python class, all they need to do is click the button.

The server will check if the meet URL is opened, if there are more than one tab is opened it will alert the student.

If there is no link then alert student to open a link. If the student did not join the meeting then it will alert

Class timing is recorded and if the class ended then the end time is provided to the server. Based on class time, attendance is provided.

CHAPTER 4

SYSTEM SPECIFICATION

4.1 SYSTEM REQUIREMENTS

4.1.1 SOFTWARE REQUIREMENTS

Operating system : Windows 10

Coding Language : PYTHON, JS

Tool : AWS EC2, WinScp, POSTMAN

Frameworks : Django

4.1.2 AWS EC2 SPECIFICATION

System : Ubuntu

Hard Disk : 30 GB

RAM : 16 MB

4.2 SOFTWARE DESCRIPTION

4.2.1 AWS EC2:

An Amazon EC2 instance is a virtual server in Amazon's Elastic Compute Cloud (EC2) for running applications on the Amazon Web Services (AWS) infrastructure.

AWS EC2 is used for hosting the server so that it can be accessed globally around the world.

4.2.2 WinSCP:

WinScp Its main function is file transfer between a local and a remote computer and used to integrate the Cloud tools with PuttyGen.

PuttyGen is an key generator tool for creating SSH keys for putty and which is a terminal emulator application which can act as a client for the SSH, Telnet, rlogin, and raw TCP computing protocols .

4.2.3 POSTMAN:

Postman is an application used for API testing. It is an HTTP client that tests HTTP requests, utilizing a graphical user interface, through which we obtain different types of responses that need to be subsequently validated.

Postman provides mock server, which is used for testing for front end before designing the back end setup.

4.2.4 DJANGO FRAMEWORK:

Django REST framework is a powerful and flexible toolkit for building Web APIs.

Django framework is used for developing web page for both staff and students with an inbuilt SQLite database for storing the student and staff information.

REST API is used for providing the interface between the chrome extension and application server.

Django Rest Framework (DRF), which is an application used for rapidly building RESTful APIs based on Django models.

CHAPTER 5

IMPLEMENTATION AND RESULTS

5.1 IMPLEMENTATION

Implementation Is Done As Two Modules

Chrome Extension

Web Server

5.1.1 CHROME EXTENSION:

Extensions are software programs, built on web technologies (such as HTML, CSS, and JavaScript) that enable to customize the Chrome using experience.

Google Login to the extension requires the “identity” permission from the browser.

Google Login prompt is requested from the Google Cloud Api Open Id.

```
Const CLIENT_ID = encodeURIComponent('575384002811-
v991bj9iovbalb307njkhpcba9a0t6st.apps.googleusercontent.com');
const RESPONSE_TYPE = encodeURIComponent('id_token');
```

Profile and email of the user is acquired by setting the SCOPE to “profile” and “email”.

```
const SCOPE = encodeURIComponent('openid email profile');
```

After the succesful login, Web server response the profile status , if the account is not registered , popup the alert message to create an account on Website first.

```
if (data.status === 404) {
    alert('Create An Account On Website First');
 }
```

If registered, the email and profile status is stored in local storage of the browser.

```
chrome.storage.local.set({ 'profile': 'staff', ‘email’:email_id}, function () {
    });
```

STAFF:

The list of classes is for the staff is fetched from the server and displayed on the extension

On clicking a particular class the meet url is fetched among the tabs and it posted to the web server.

```
If (More than one Meet Link Detected)
      Alert(‘Close unwanted meet links’)
Else if ( No Meet Link Detected)
Alert(‘Open Atleast one Meet Link Before Start’)
Else
Post(meet url to the server)
```

STUDENT:

The list of enrolled classes for the student is fetched from the server and displayed on the extension.

On clicking a particular class the meet URL is fetched from the server and url is launched on new tab.

```
chrome.tabs.create(  {url : meet_url} )
   chrome.windows.create({ url: face_detection_window_url , type : 'panel' })
```

Time calculation is done with the help of following two modules

Screen time

Face detection time

5.1.1.1 SCREEN TIME:

Screen time is the time for which the user is using that particular tab without switching tabs, changing focus, minimizing tabs and closing tabs.

If the user switch tab changing focus or minimize tab.
It stops calculating the time.

```
const timeDiff = parseInt((Date.now() - active.time) / 1000);
 total += timeDiff;
   active = {};
```

Screen time detection requires the following permissions from the browser

```
"activeTab",
"tabs",
"<all_urls>",
"scripting",
"storage".
```

5.1.1.2 FACE DETECTION TIME:

The face is detected using the pre-defined haarcascade models .

From the Face API the faces in the video are detected .

```
const detections = await faceapi.detectAllFaces(video, new
faceapi.TinyFaceDetectorOptions())
```

The maximum of the face detection and screen time is calculated and sent to the server for every 5 minutes.

```
fetch(url, {method: "POST",
headers: {
'Content-Type': 'application/json',
'Accept': '/'
},
body: JSON.stringify(data)
}).then(response => response.json())
.then(data => {
console.log("sent to server")total = 0});
```

5.1.2 WEB SERVER:

The models (tables) are created for the database using the inbuilt sqlite database.

```
class Class(models.Model):
    classname = models.CharField(max_length=15,unique=True)
    description = models.TextField()
    owner = models.ForeignKey(User,related_name="owner",on_delete =
models.CASCADE)
user = models.ManyToManyField(User,null=True,blank=True)

class MeetUrl(models.Model):
    classname = models.ForeignKey(Class,on_delete = models.CASCADE)
    url = models.URLField(max_length=115,null=True,blank=True)
    starttime = models.DateTimeField(null=True,blank=True)
    endtime = models.DateTimeField(null=True,blank=True)
    created = models.DateTimeField(auto_now_add=True)
```

Login page for the staff and student is done by using the django's built in authentication system.

If user is not registered, user can register with email ,username and is Staff status.

After successful Login, the profile page is displayed.

STAFF:

The profile page contains the list of classes and students associated with the classes.

```
def Profile(request):
    user = User.objects.get(email=request.user.email)
    return render(request,'main/profile.html',{ 'data' : user })
```

Creation of new class involves the addition of entry in the Class Model(Class Table) and updating class is the update of this particular entry.

```
class ClassCreateView(LoginRequiredMixin,CreateView):
    model = Class
    login_url = '/accounts/login/'
    fields = ['classname','description']
```

```
    template_name = 'main/create.html'
    success_url = '/profile_staff'
    def form_valid(self, form):
        form.instance.owner = self.request.user
        self.object = form.save(commit=False)
        self.object.save()
        return HttpResponseRedirect(self.get_success_url())
```

On clicking the particular class, the attendance page is displayed by fetching the students associated with the class in the Timings Table.

```
classe = Timings.objects.filter(pkformeeturl = pk).order_by('-timeListened')
```

STUDENT:

Profile page contains the list of enrolled classes.

Student can join the classes by using the class name.

```
def AddClass(request):
    if request.method == 'POST':
        form = JoinClassForm(data=request.POST,user=request.user)
        if form.is_valid():
            print("valid")
            return HttpResponseRedirect('/profile')
        return render(request, 'main/create.html',{'form':form})
    form = JoinClassForm(user=request.user)
    return render(request, 'main/create.html',{'form':form})
```

5.2 RESULT:

This project has been implemented as a web application and chrome extension with all the proper modules.

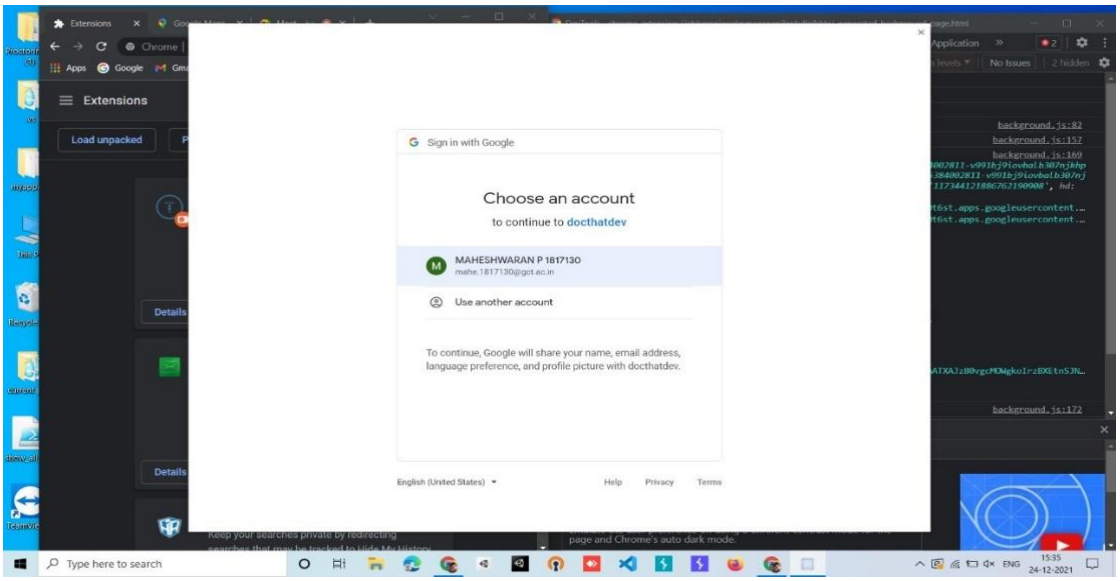


Figure No. 1 Google Signin(Extension)

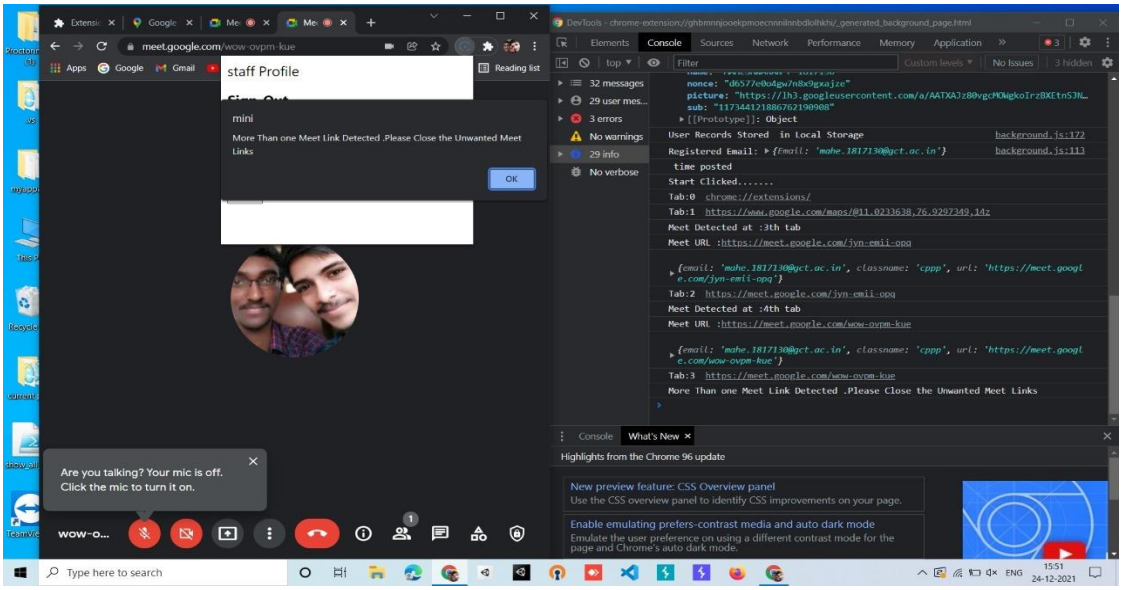


Figure No. 2 Checking Meet tabs

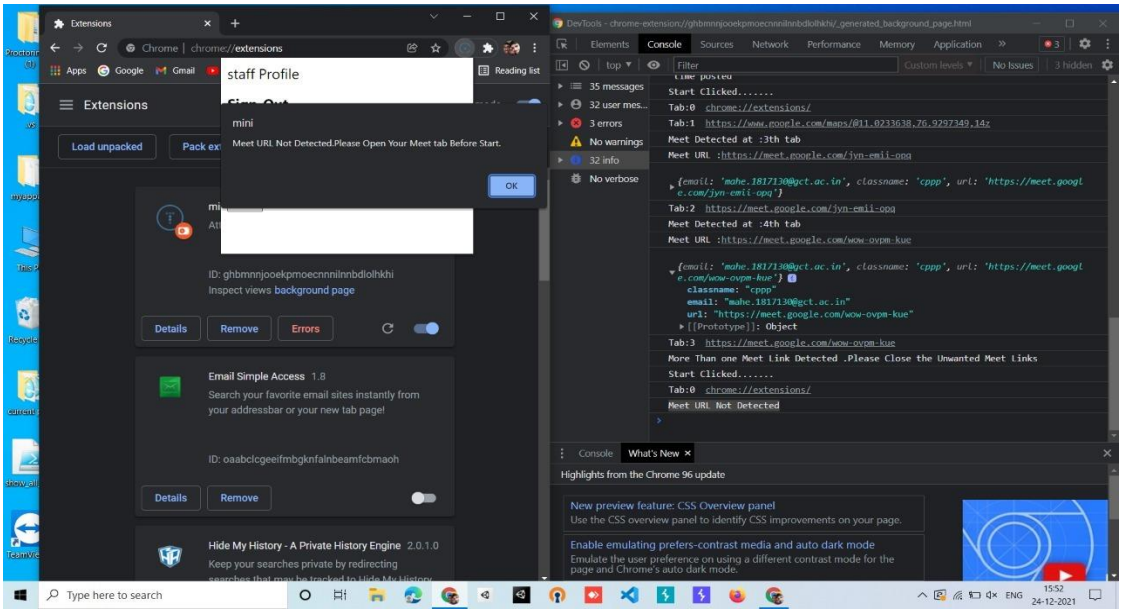


Figure No. 3 Alerting user to open Gmeet.

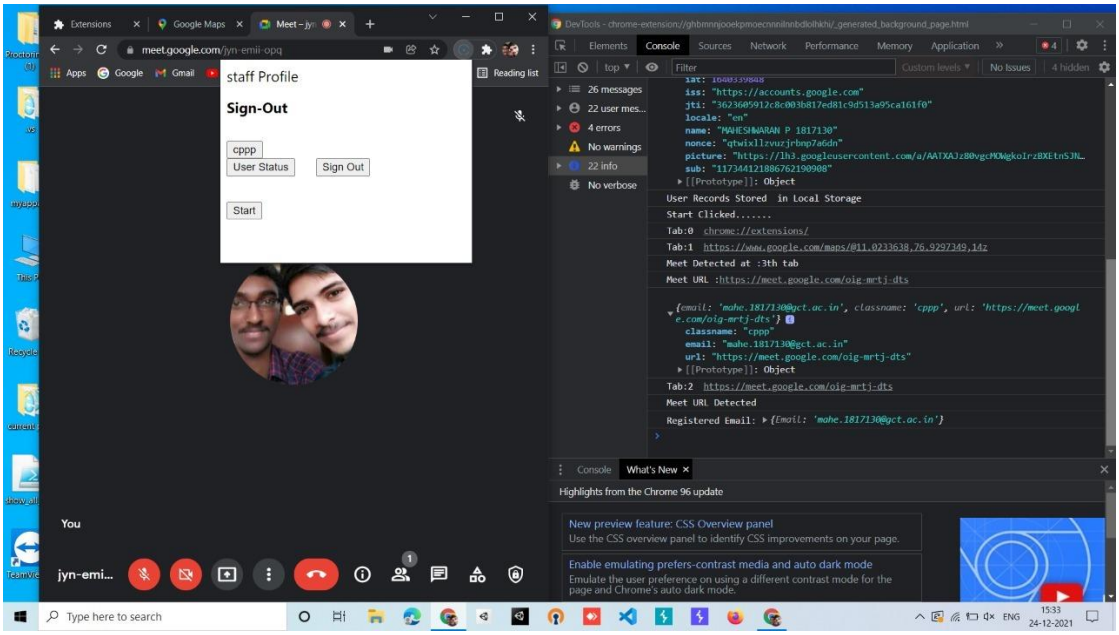


Figure No. 4 Staff Profile(Extension)

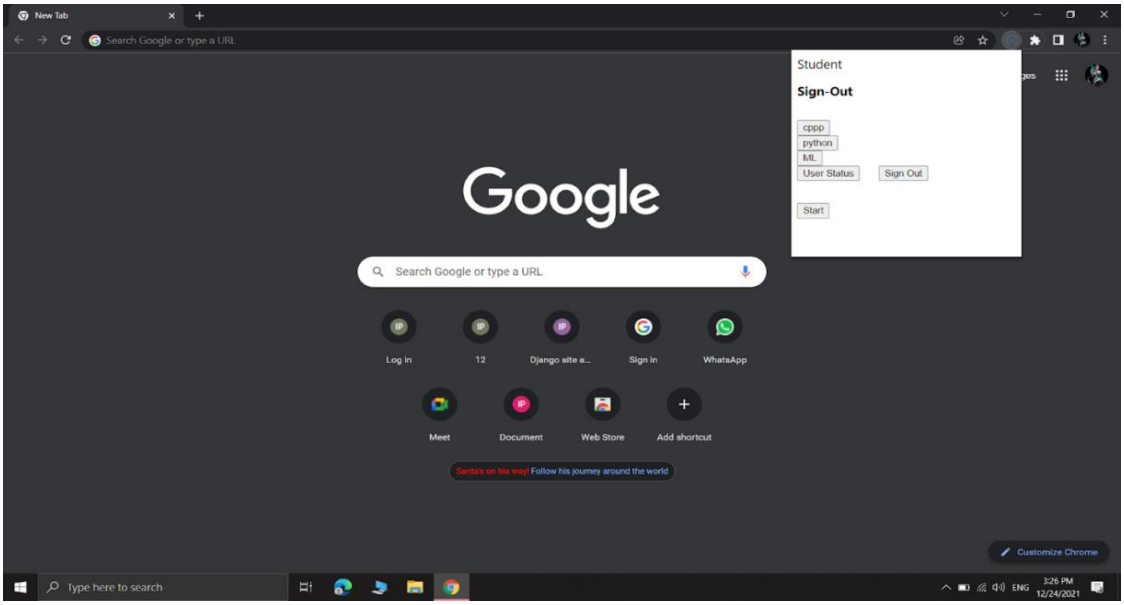


Figure No. 5 Listing Classes for Student

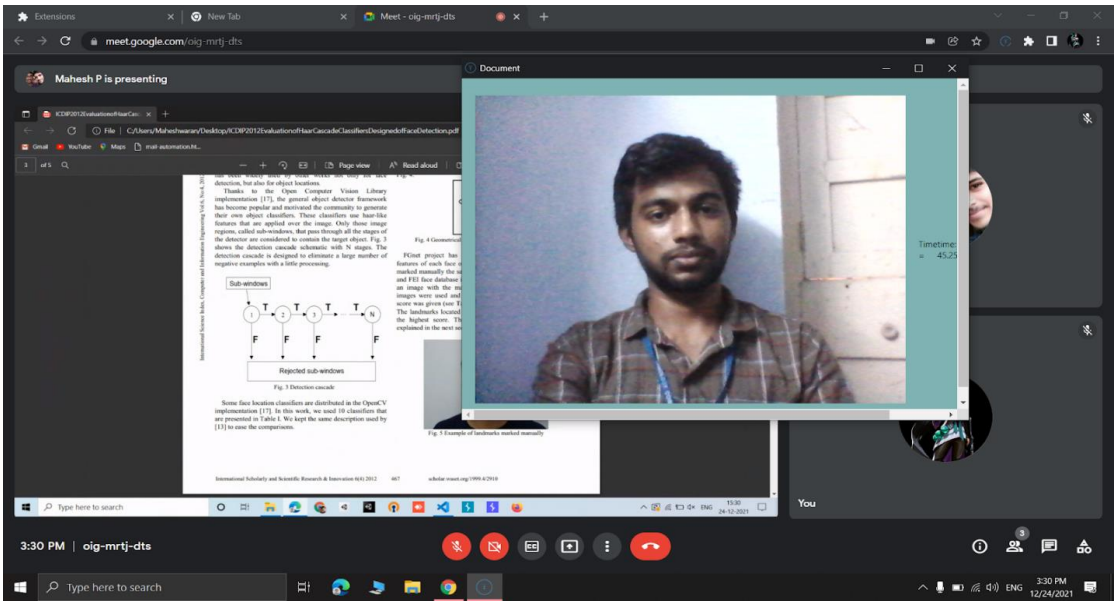


Figure No. 6 Detecting Screen Time

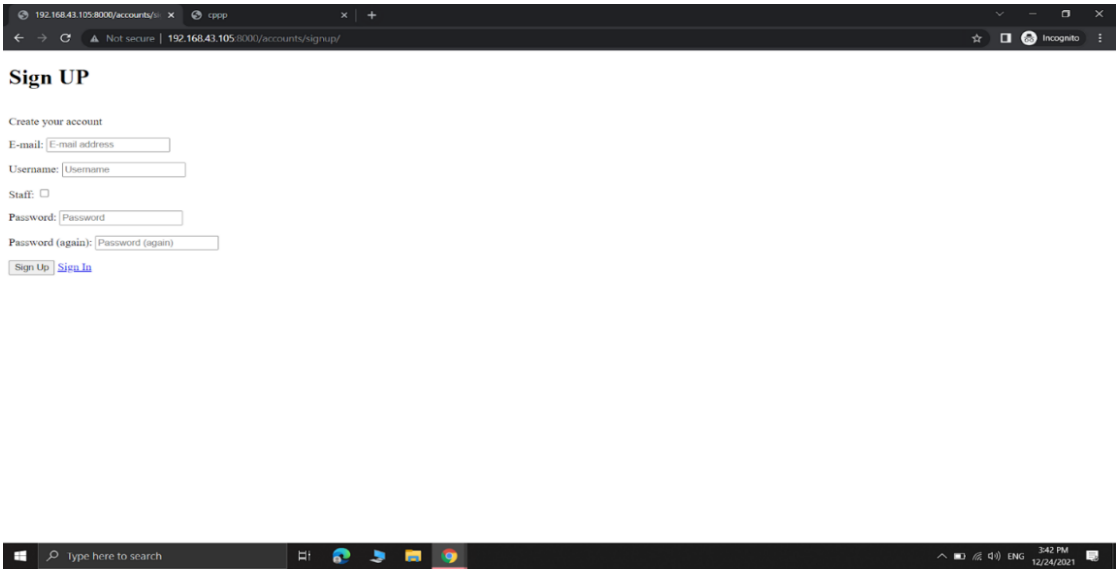


Figure No. 7 SignUp in Website.

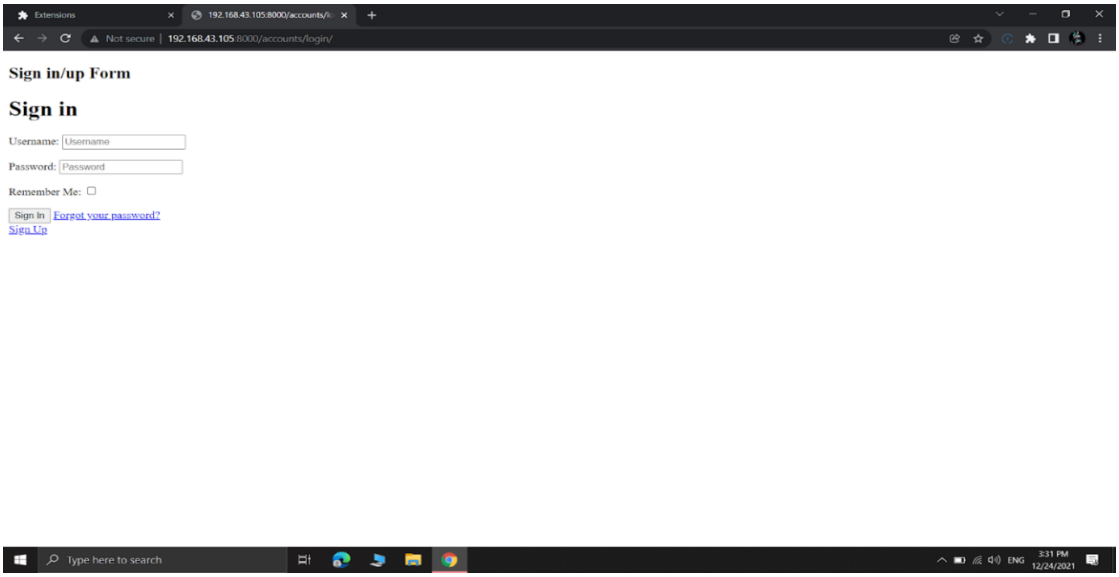


Figure No. 8 Signin in Website.

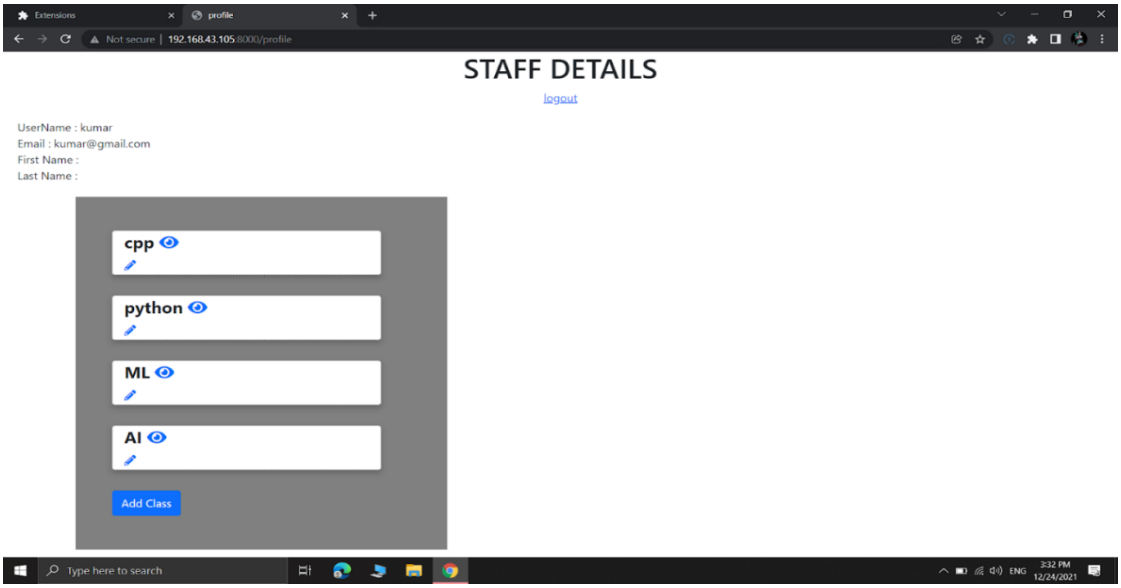


Figure No. 9 Staff profile in Website.

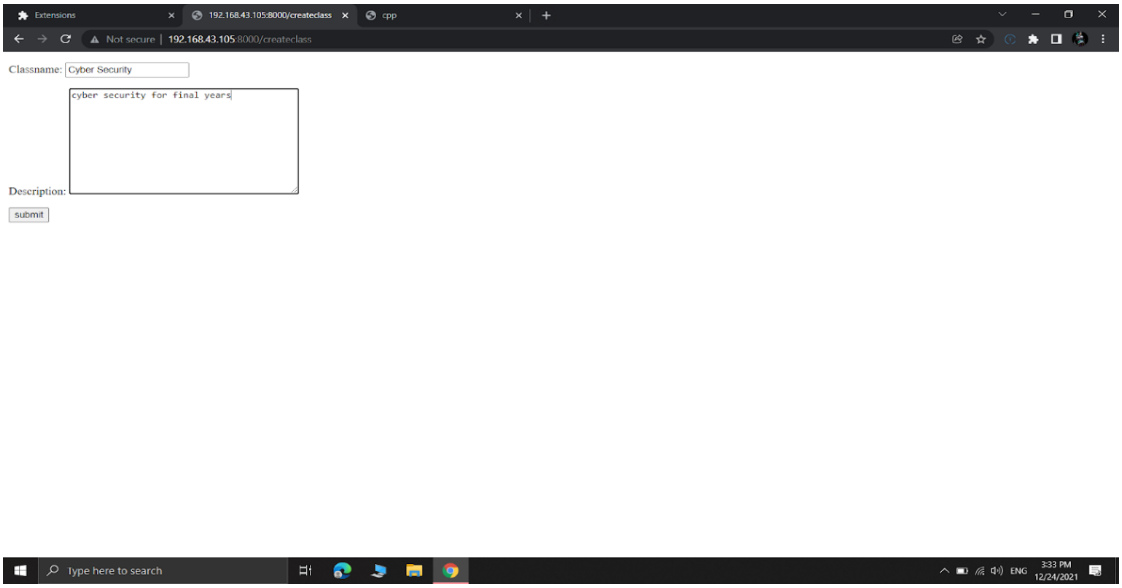


Figure No. 10 Adding new Classes.

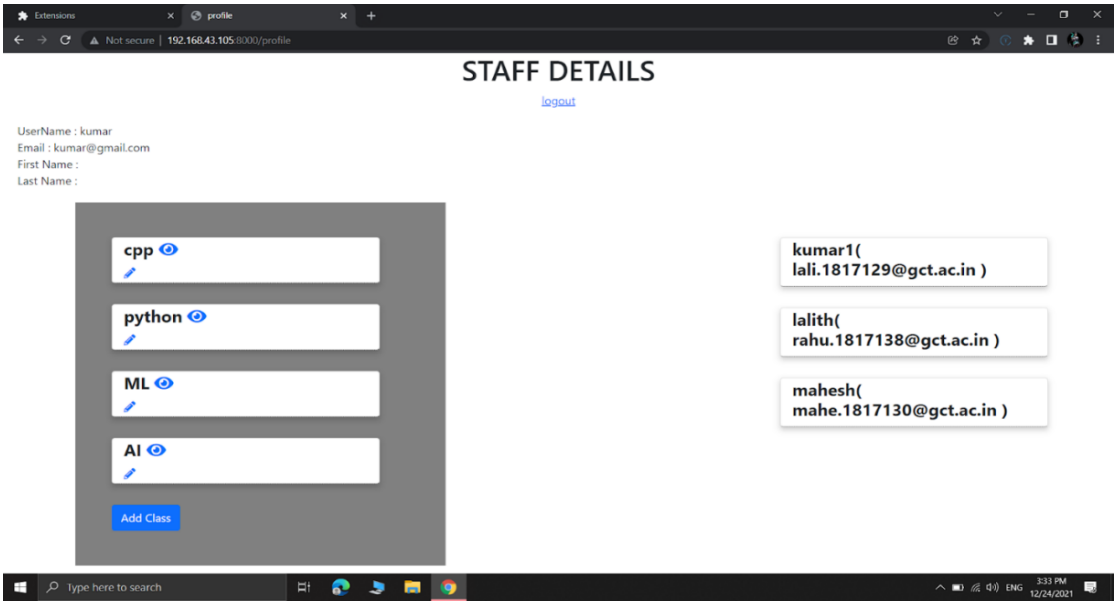


Figure No. 11 Listing Students in Class.

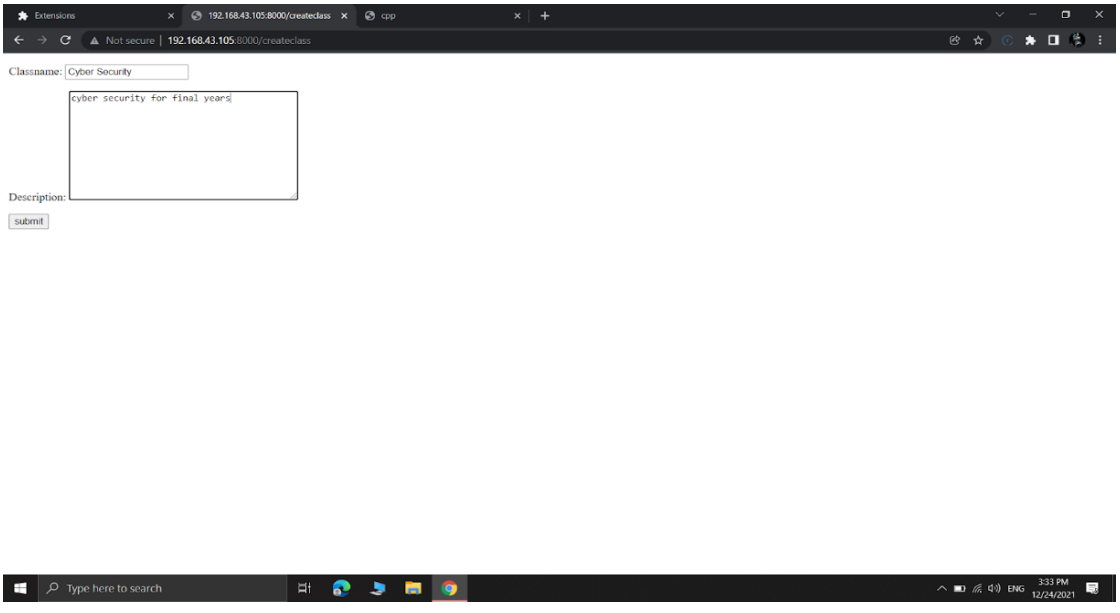


Figure No. 12 Updating Class Info.

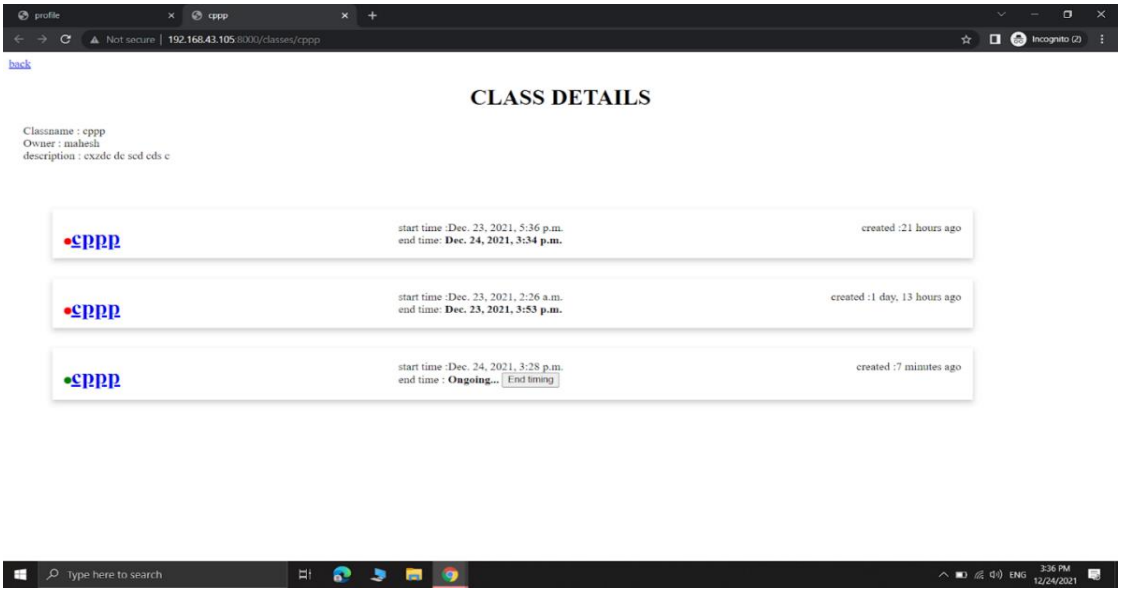


Figure No. 13 List of Class Ongoing and Finished.

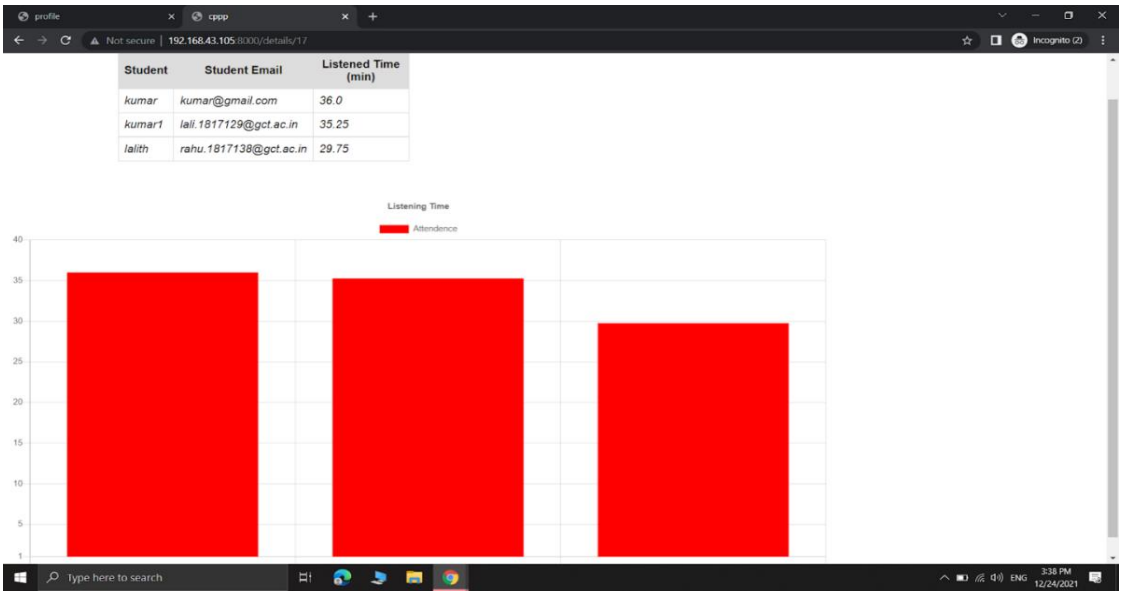


Figure No. 14 Stats(Attendance) for the Classes.

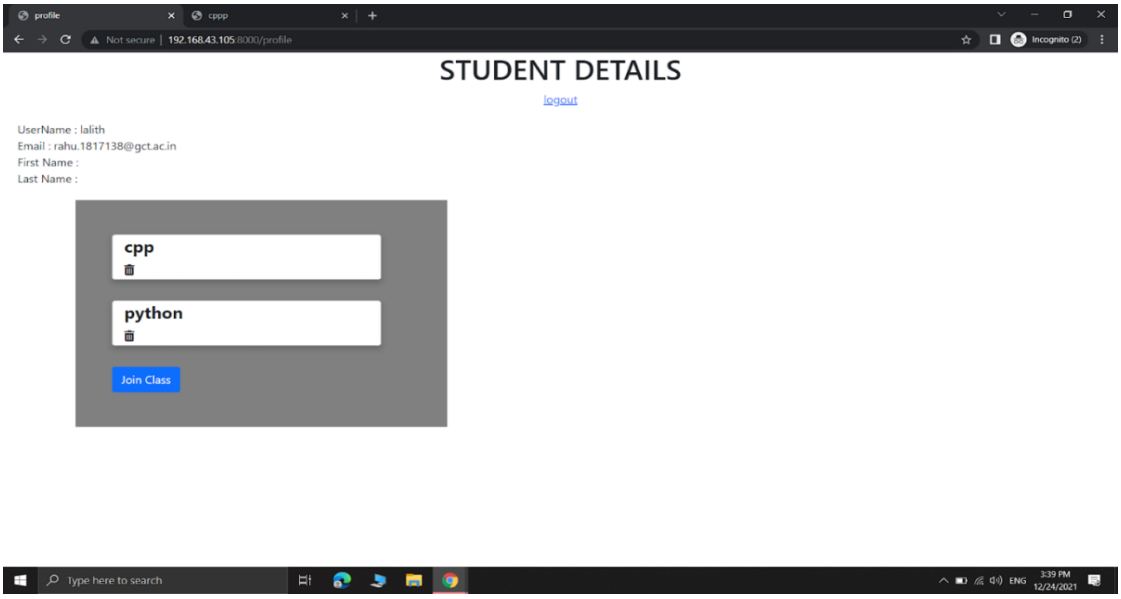


Figure No. 15 Student Profile in Website.

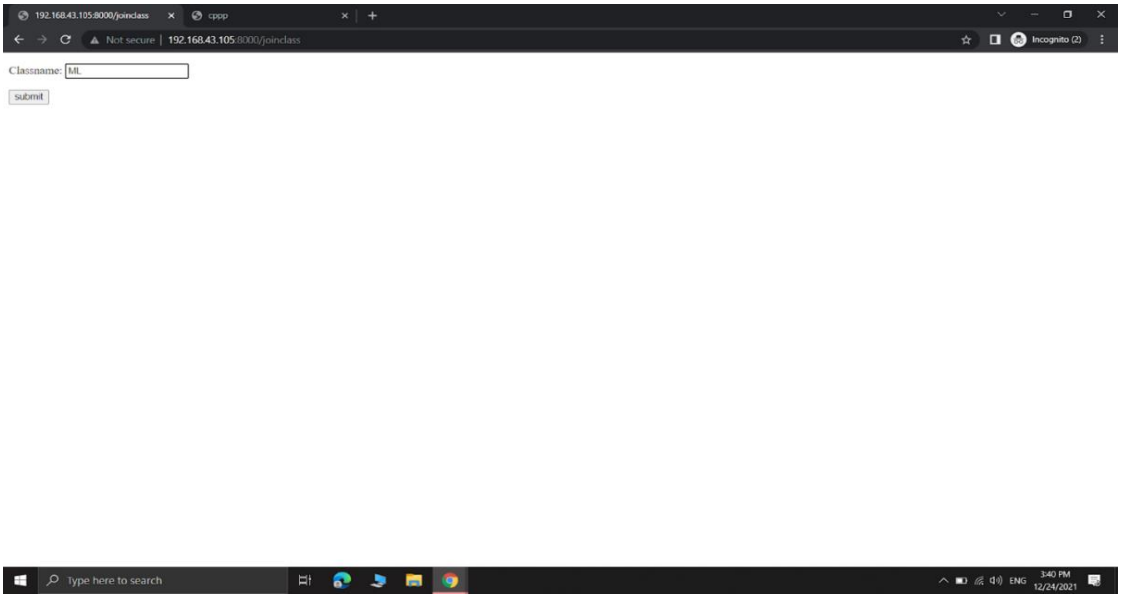


Figure No. 16 Student Joining Class.

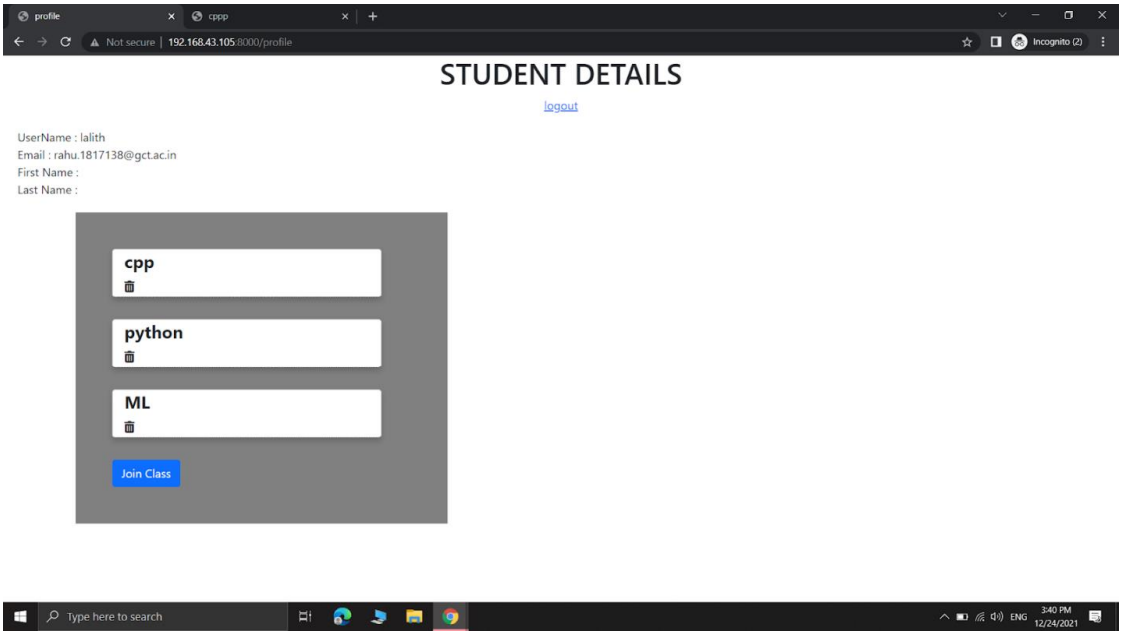


Figure No. 17 Updating joined Class.

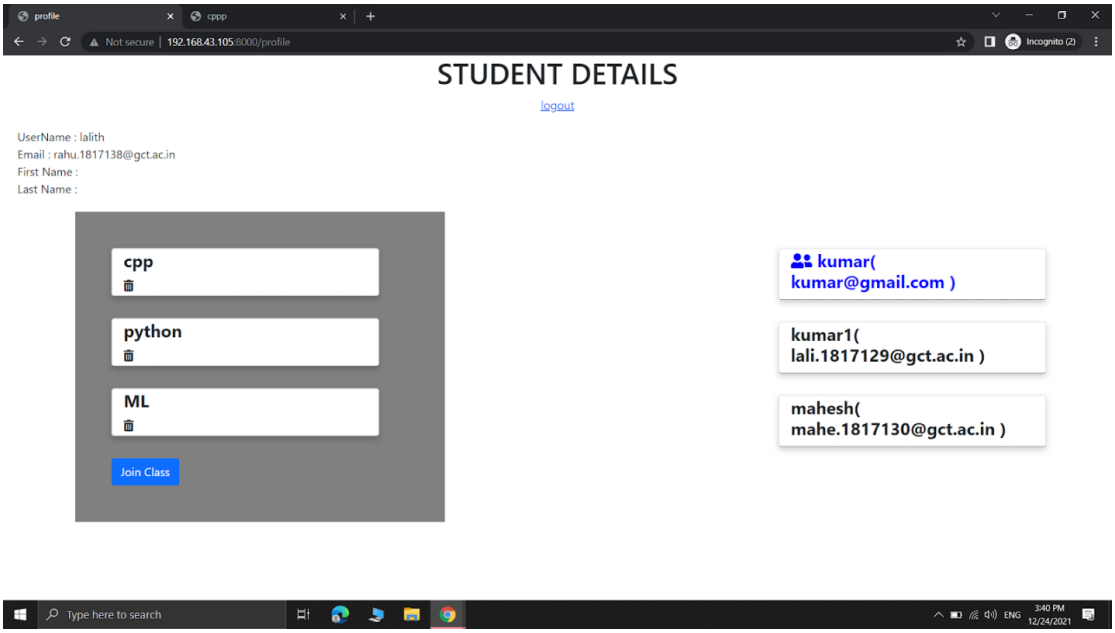


Figure No. 18 List of Students in Class.

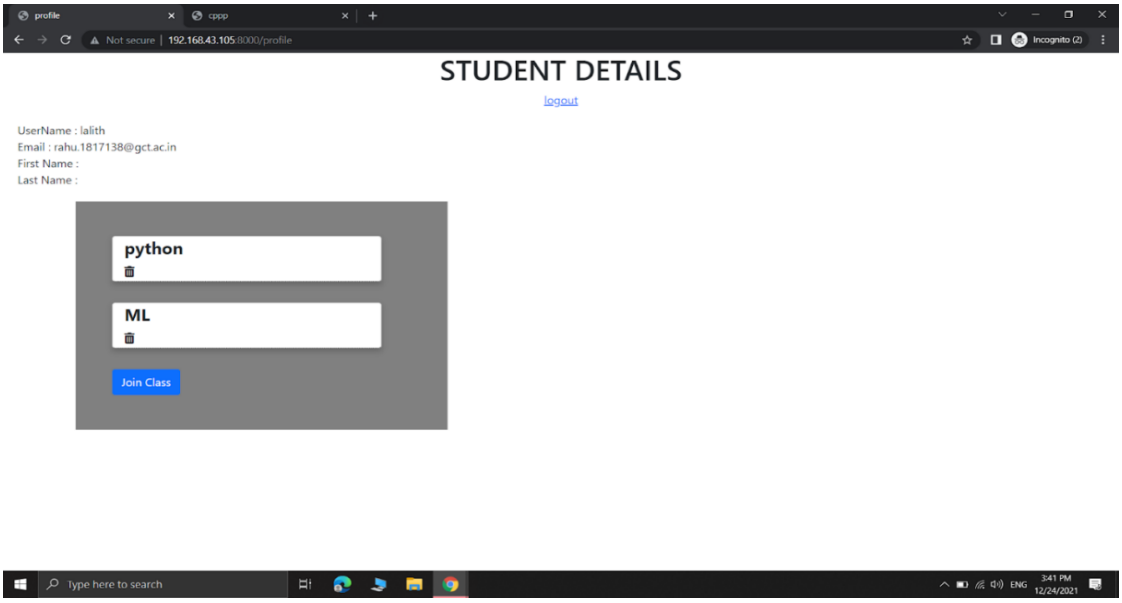


Figure No. 19 Deleting Class.

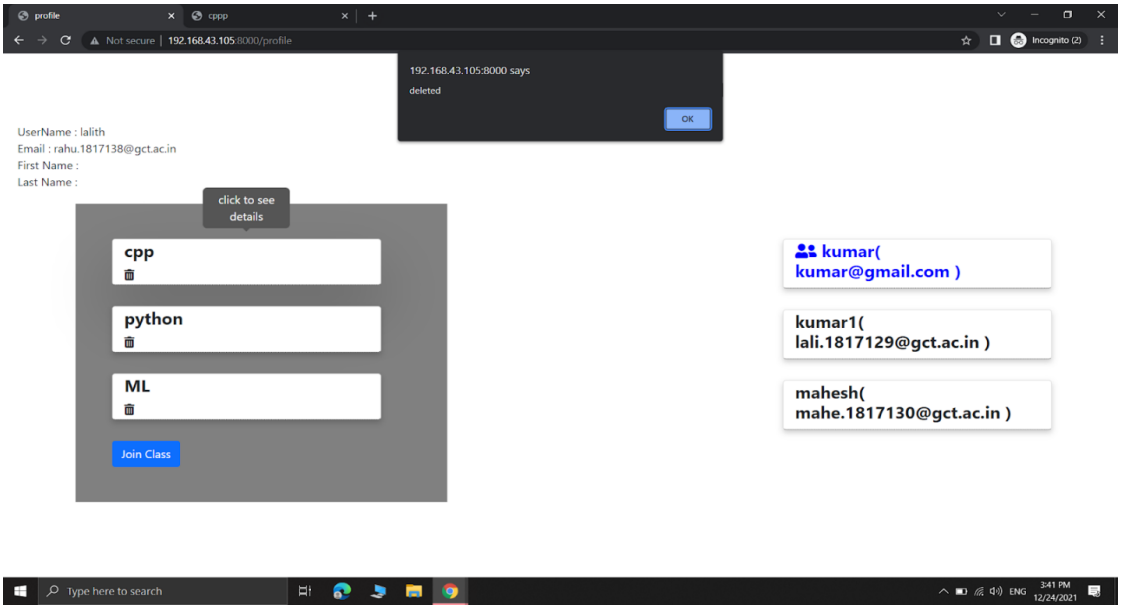


Figure No. 20 Deleting Class.

CONCLUSION

In this project, with the use of Haarcascade algorithm for the face detection (to calculate the time, particular user is present in front of the Gmeet) and chrome extension for screen time (to calculate the time that user has not used other applications in the foreground other than Gmeet) is gathered and the overall time particular user listened the Gmeet is calculated.

The calculated data from both face time and screen time is sent to the server. The Staff can analyze those data and can mark attendance for students depending on the attendance criteria.

Hence the aim of the project (marking attendance and increase the student's attention) is achieved and the project works perfectly and can be considered for future purpose.

REFERENCES

[1] Sudakshina Singha Roy and K.P. Jevitha,
CBEAT: Chrome Browser Extension Analysis Tool
https://www.researchgate.net/publication/320951117.

[2] R. Padilla, C. F. F. Costa Filho and M. G. F. Costa,
Evaluation of Haar Cascade Classifiers Designed for Face Detection
https://www.researchgate.net/publication/303251696.

[3] Chrome Extension Documentation -
https://developer.chrome.com/docs/extensions/

[4]Django Documentation - https://docs.djangoproject.com/en/4.0/

[5] Aws Documentation - https://docs.aws.amazon.com/