# Event Tracker

Guided By:
Mrs. A . Madhavi
Asst. Professor, CSE dept.

Saketh Reddy -18071A0547
Sairam Tammisetti-18071A0555
Sarvotam Reddi Vancha-18071A0558

# Abstract


- *In these days almost every college is conducting technical / non-technical events where we can gain some knowledge by participating.*
- *College, which is conducting events, this project can be organizer for the various events offered by them. This provides college management to schedule events online and assign organizers for an event.*
- *The students register with the site and some of them register as coordinators. Only event organizers and volunteers can upload event content to the site.*
- *Here comes the salient feature of the Event Tracker which is enabled by Django, it can automatically generate certificates and distribute them to the participants via Email and also provides facility for the students to download it from their dedicated account .*

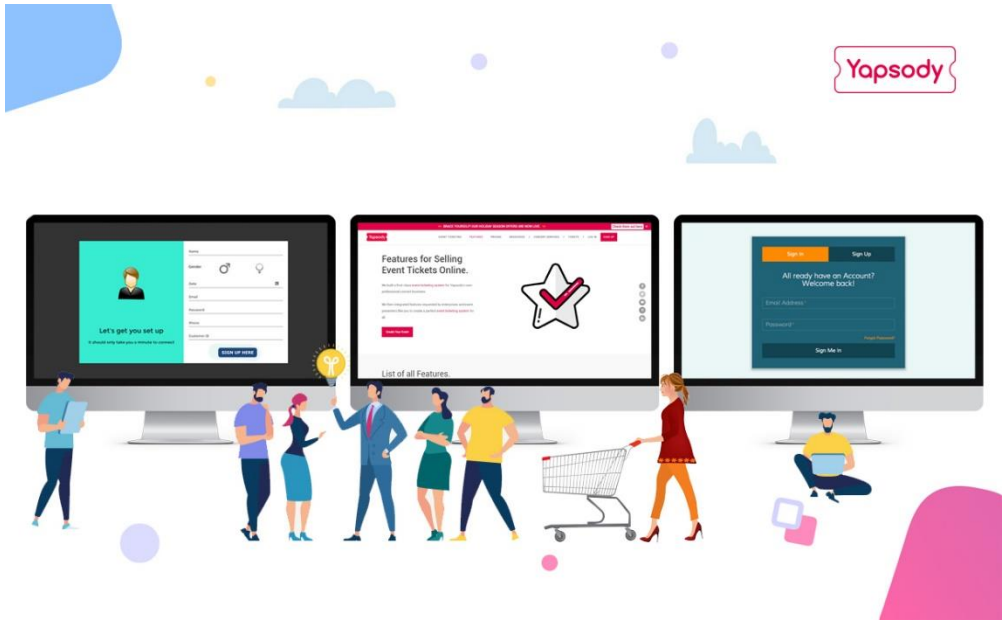
# Introduction to Framework

Django is a high-level Python web framework that enables rapid development of secure and maintainable websites. Built by experienced developers, Django takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel. It is free and open source, has a thriving and active community, great documentation, and many options for free and paid-for support.


# Literature Review

Existing System v/s Proposed System

# Existing System




**Current System of Organizing a Event**

- **Registration Process** – Google Forms
- **Registration Fee Payment –** Registration Desk , Google pay
- **Tracking the Event –** Informed through Watsapp Groups
- **Marking Attendance of the Event –** Recorded in some form
- **Issuing Certificates for the participants –** Paper Work done manually

# Proposed System

## Event Tracker

- *One of the main agenda of the Event Tracker is to centralize all the events happening in campus. An authority is given access to create organizer accounts so that any organizer should approach him to register his personalized account which he can use to track the student registrations , mark attendance and view event statistics.*
- *Here comes the salient feature of the Event Tracker which is enabled by Django, it can automatically generate certificates and distribute them to the participants via Email and also provides facility for the students to download it from their dedicated account . This enables students to view their certificate and download them whenever require at any time in the career.*

# Requirement Analysis

## Hardware Requirements

This web application is compatible to run on any Web server in which domain name is registered.

Hardware requirements may include :

- RAM : 4 GB and higher.
- Processor : Intel i3 and above
- Hard Disk : 10 GB and above

## Software Requirements

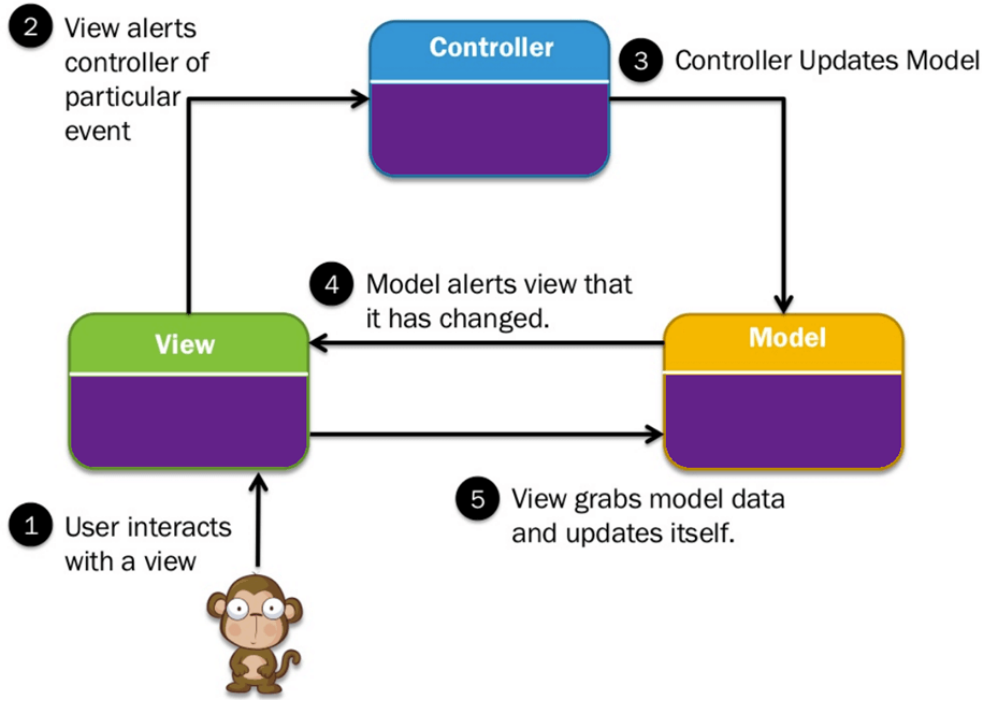
Building of this web application used many software applications. Here are the major among them:

- Python
- Django
- Vscode
- PIL – python imaging Library
- NOSQL database - postgreSQL

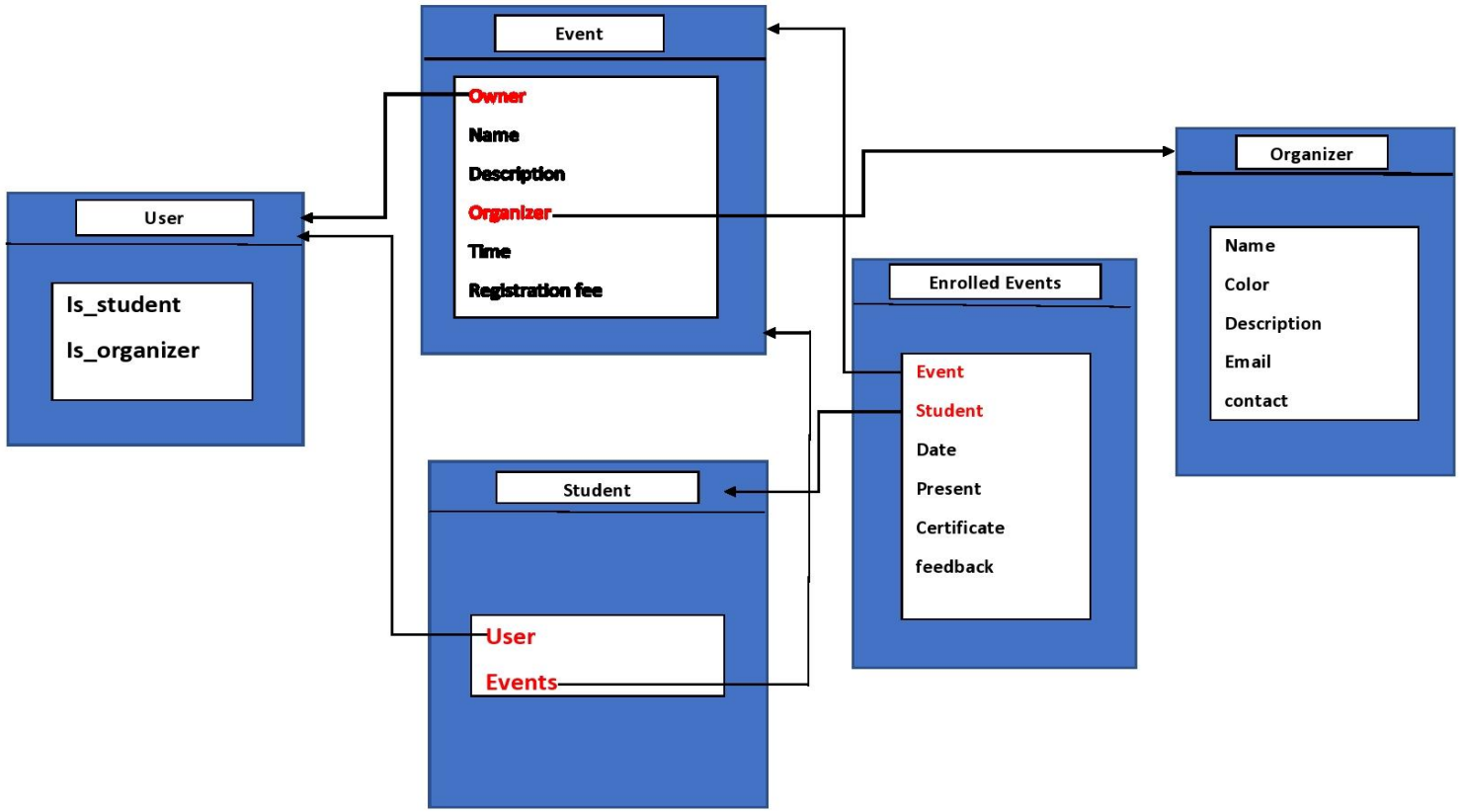
# System Architecture

## MVC Architecture

The Model-View-Controller (MVC) framework is an architectural pattern that separates an application into three main logical components Model, View, and Controller. It was traditionally used for desktop graphical user interfaces (GUIs).



# Database Structure



# Salient Features of Event Tracker

**Certificate Issuing Process:**

- Certificates are automatically generated digitally by accessing the details of the students from their registration details.
- Once generated certificates are sent through mail to the students.

**Organizers :**

- Can Add Event
- Can track the Participants
- Can modify the event details
- Can mark attendance for the participants
- Can review and generate certificates.

**Students:**

- Students can view the list of all events happening in the campus
- Can enroll for any event
- Can Download certificate from the website once it is issued

# Backend “Behind the Scenes”

```
    student_instance.update(certificate=img.tobytes())
    subject = "Certificate of " + event.name + " from Terrific Trio!"
    message = "Congratulations!!! You have succesfully achieved Certificate from Terrif
    recepient = s.student.user.email
    msg = EmailMessage(subject, message, settings.EMAIL_HOST_USER, [recepient])
    msg.content_subtype = "html"
    msg.attach_file(file_path)
    msg.send()

return redirect("organizers:event_change_list")
```

**Code of mailing a generated certificate**

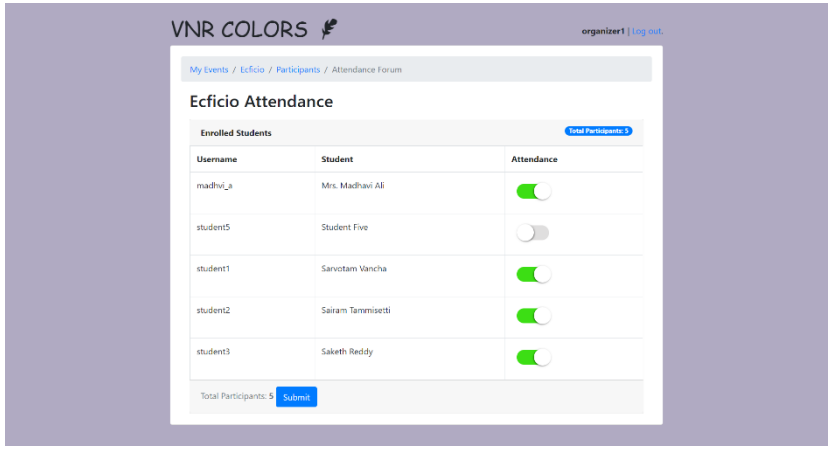
# Code to mark attendance

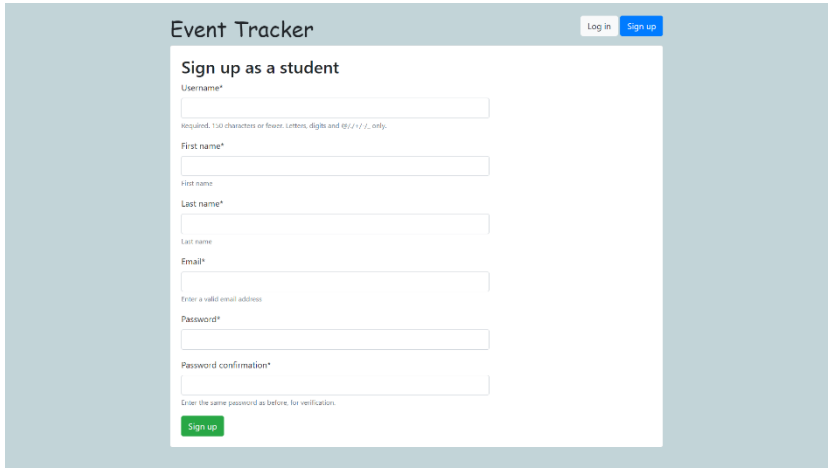
```
@login_required
@organizer_required
def save_data(request, pk):
    if request.method == "POST":
        # print(request.POST.getlist("present"))
        student_list = request.POST.getlist("present")
        event = Event.objects.get(pk=pk)
        # print(event)

        EnrolledEvents.objects.filter(event=event).filter(
            student__in=student_list
        ).update(present=True)

        EnrolledEvents.objects.filter(event=event).exclude(
            student__in=student_list
        ).update(present=False)
        messages.success(request, "Attendance was Updated Successfully! Go Ahead!")
        return redirect("organizers:event_change_list")
```

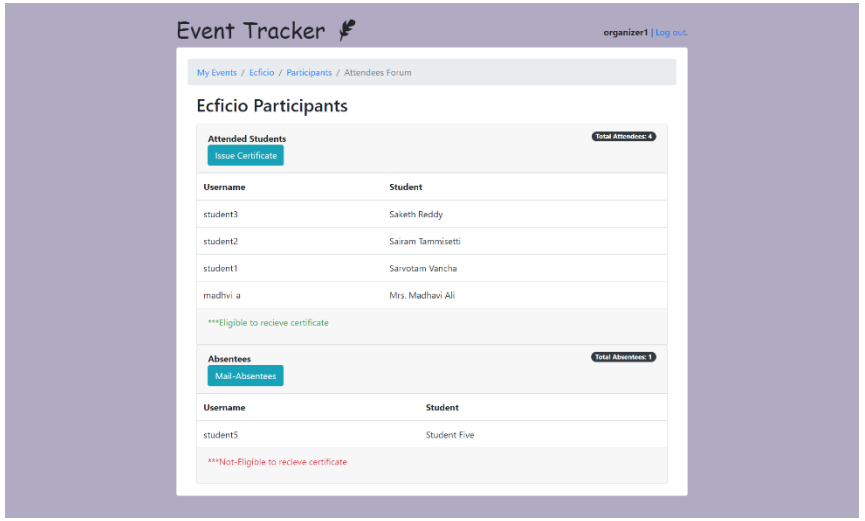
# Code to make Certificate Downloadable

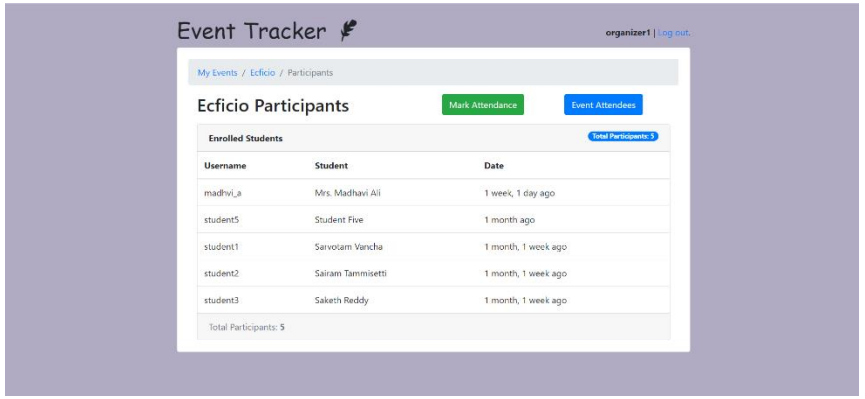
```
if request.method == "GET":
    bin_img = EnrolledEvents.objects.get(event=event, student=student).certificate
    image = Image.frombytes("RGB", (image_width, image_height), bin_img, "raw")
    # print(image)
    file_path = os.path.join(settings.MEDIA_ROOT, "certi.jpg")
    image.save(file_path)
    if os.path.exists(file_path):
        with open(file_path, "rb") as fh:
            response = HttpResponse(
                fh.read(), content_type="application/vnd.ms-excel"
            )
            response["Content-Disposition"] = (
                "inline; filename="
                + os.path.basename(
                    event.name + "_" + request.user.first_name + ".jpg"
                )
            )
            return response
    raise Http404
```

# USER INTERFACE – “Face of the website”



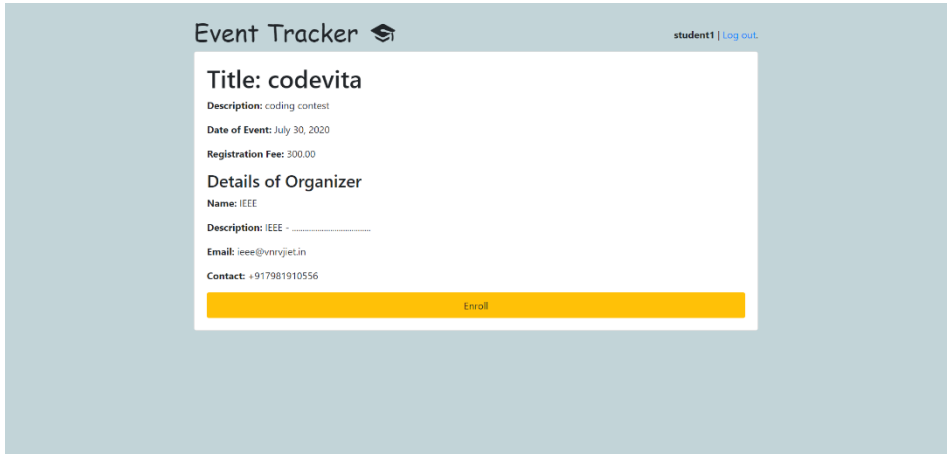



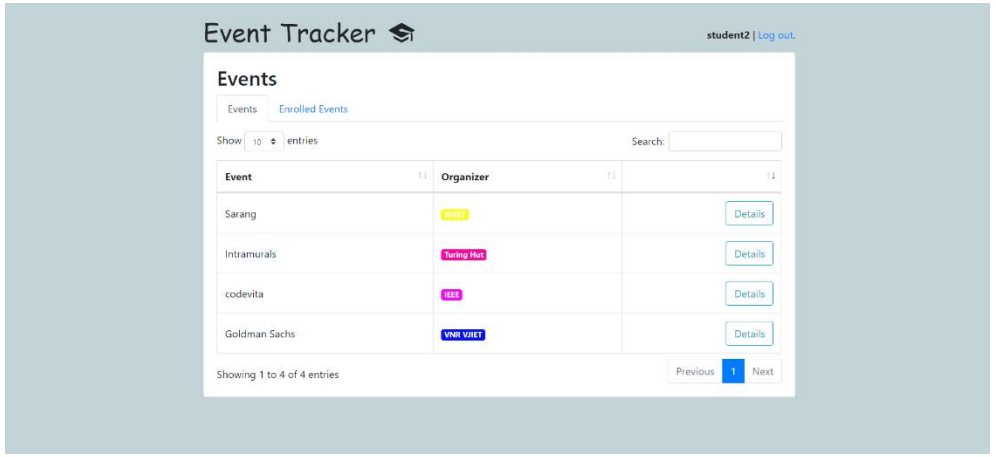



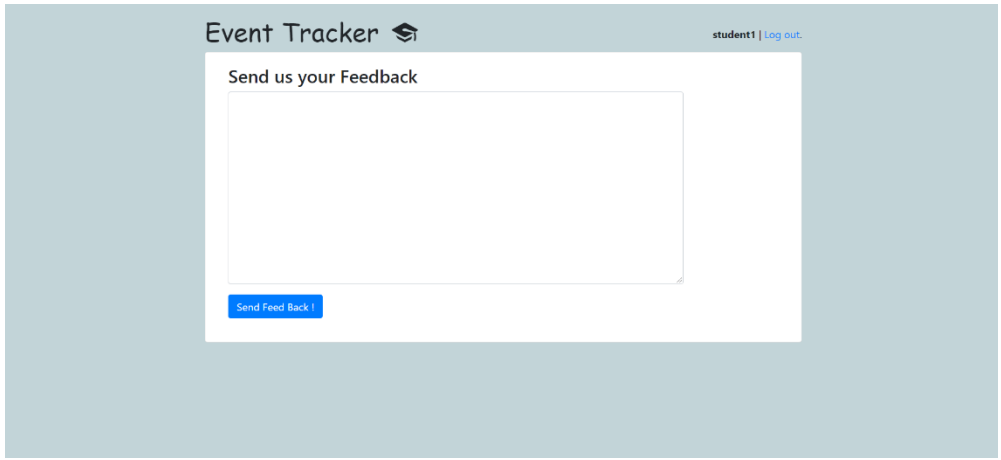



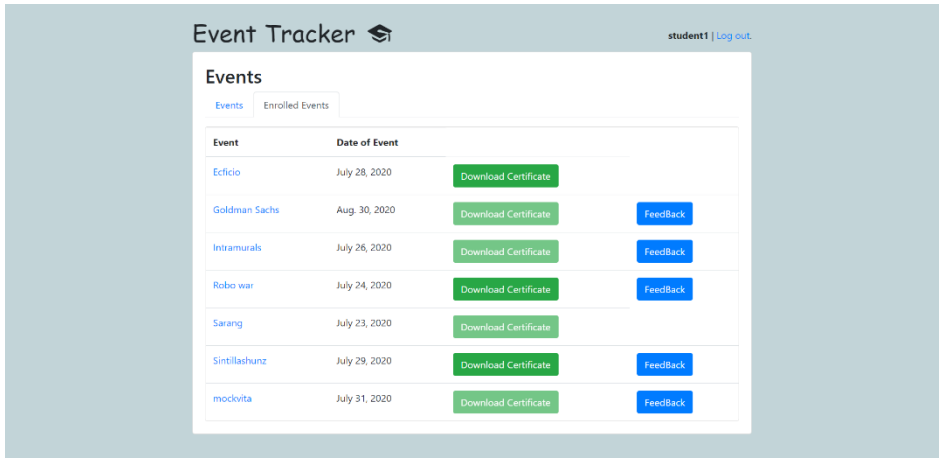

**User Interface View Of the Organizer**

# USER INTERFACE – “Face of the website”



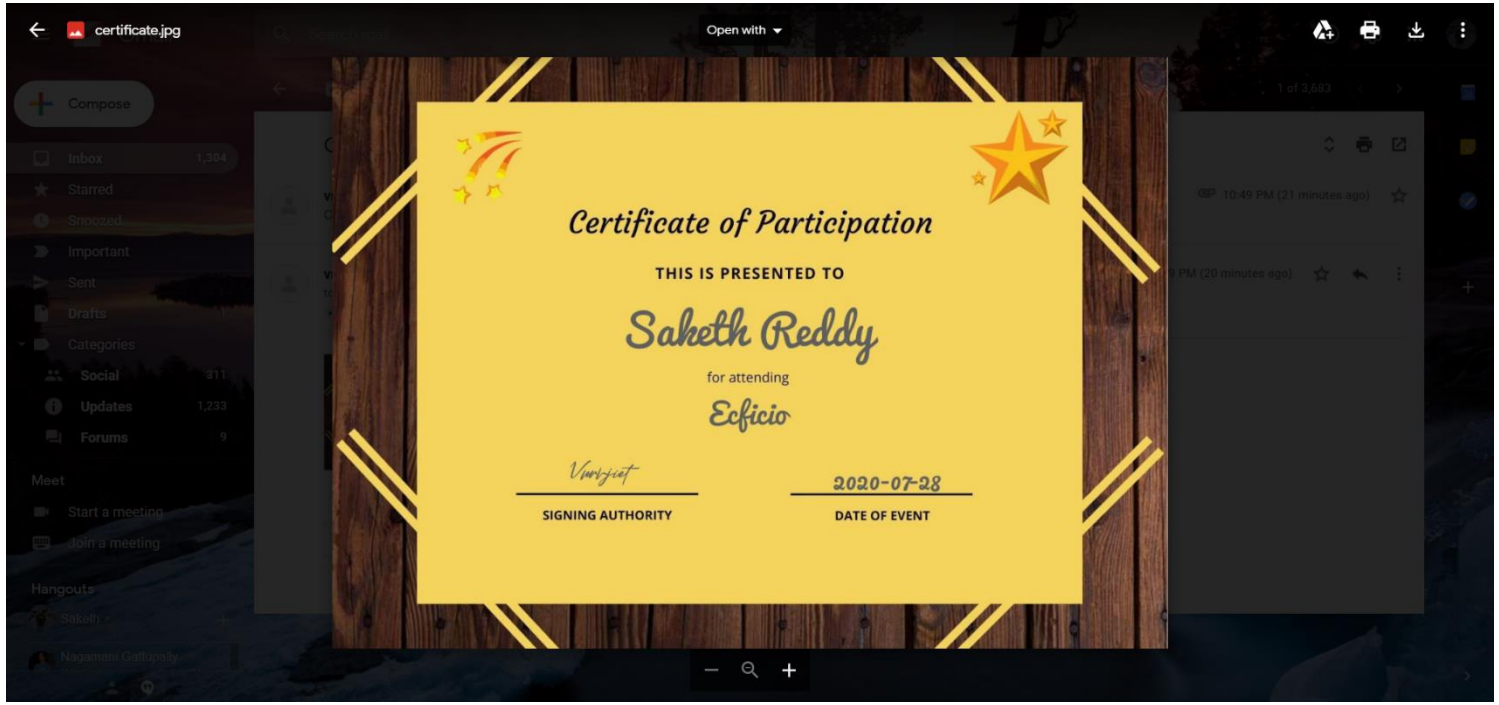



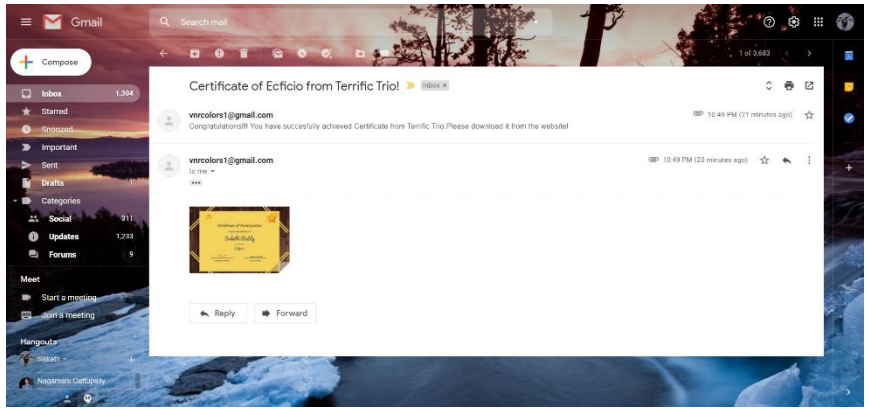



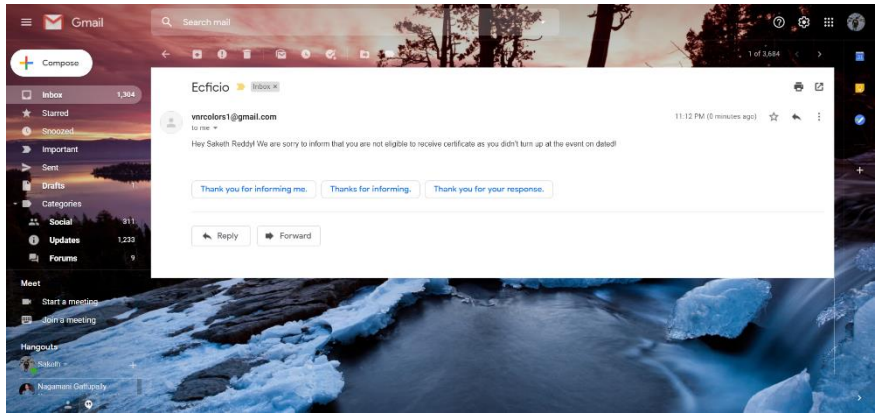




**User Interface View of Student**

# Certificates Generated and mailed automatically



# Conclusion

- We have prepared new system after identifying issues in existing manual system. In which easy to use GUI is proposed by which Student coordinators, Staff coordinator can view all the records which are necessary.
- Participant students can view and register for events online. The record maintenance are use of previous records becomes easy and effective communication between staff coordinators and student coordinators.
- Thus we will implement Event Tracker system to address the problem faced by event organizers with respect to communication and working methods.

# Conclusion


The main goals of the project are:

- To reduce the communication gap between students / participants and event coordinators of the events.
- Online registration / application of participants. -Proper utilization of the schedule and digital resources.
- Easy to assign students for campaigning.
- To reduce the overhead of the organizing committee.
- To generate and send certificates online reducing lot of paper work as well as manual work.

# Future Scope

- Our proposed system is basically a prototype application for future developers.
- As a team we are planning to extend the members and going to improve the system architecture.
- We need a front-end developer who is highly skilled to convert our project to make ready for production.
- A few systems can be improved like Embedding the payment gateway , Making the mail issuing process to work faster.
- We are very enthusiastic to work on this project and make it ready for production so that our college can embed an excellent Event Tracker system.

# References

## WEBSITES


- https://cloudinary.com/documentation/django_image_manipulation#image_optimizations
- https://www.geeksforgeeks.org/create-certificates-using-python-pil/
- https://djangoschools.herokuapp.com/
- https://simpleisbetterthancomplex.com/tutorial/2018/01/18/how-to-implement-multiple-user-types-with-django.html
- http://django-todo.org/todo/
- https://simpleisbetterthancomplex.com/tutorial/2018/01/18/how-to-implement-multiple-user-types-with-django.html
- https://django-sis.readthedocs.io/en/latest/
- https://www.geeksforgeeks.org/python-sessions-framework-using-django/
- https://www.tutorialspoint.com/django/django_sessions.htm
- https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Tutorial_local_library_website
- https://stormy-cove-10463.herokuapp.com/catalog/
- https://studygyaan.com/django/best-practice-to-structure-django-project-directories-and-files
- https://cloudinary.com/documentation/django_image_manipulation#image_optimizations

# References – Journals and Books


## JOURNALS

- Review on College Event Organizer - International Research Journal of Engineering and Technology (IRJET)- Volume: 04 Issue: 3 | Mar -2017 - https://www.irjet.net/archives/V4/i3/IRJET-V4I3121.pdf

## Books

- **The definitive guide to Django: Web development done right**
  **https://books.google.com/books?hl=en&lr=&id=h2tR8p-4a9QC&oi=fnd&pg=PR27&dq=django&ots=Voqaj0cMMb&sig=Lw785f3qhHWEUjnwFLAVOjlXx5w**
- **Python web development with Django**
  **https://books.google.com/books?hl=en&lr=&id=M2D5nnYlmZoC&oi=fnd&pg=PT31&dq=django&ots=vY-LKoeQKU&sig=U8H0yDmg3Frr5uNuMHz3fKZPPD0**
- **Practical Django Projects**
  **https://books.google.com/books?hl=en&lr=&id=qfp4BAfAvVkC&oi=fnd&pg=PR11&dq=django&ots=_0kEPcZZZd&sig=weYaaHqwQQPhNVD_FyHqqKuqULo**


# Thank you