SAVITRIBAI PHULE PUNE UNIVERSITY

A PRELIMINARY PROJECT REPORT ON

"Event Report Genarator"

SUBMITTED TOWARDS THE PARTIAL FULFILMENT OF THE REQUIREMENTS OF

BACHELOR OF ENGINEERING (TE COMPUTER ENGINEERING)

Academic Year: 2019-20

By:

Prajakta Bhosale (TECOB201)

Varun Gadde (TECOB204)

Chaitanya Nawale (TECOB218)

Siddhi Pardeshi (TECOB224)

Under The Guidance of

Prof. Sonal Gore

DEPARTMENT OFCOMPUTER ENGINEERING,

PIMPRI CHINCHWAD COLLEGE OF ENGINEERING

SECTOR 26, NIGDI, PRADHIKARAN

PIMPRI CHINCHWAD COLLEGE OF ENGINEERING

DEPARTMENT OF COMPUTER ENGINEERING

CERTIFICATE

This is to certify that, the project entitled

“EVENT REPORT GENERATOR”

is successfully carried out as a mini project successfully submitted by following students of “PCET's Pimpri Chinchwad College of Engineering, Nigdi, Pune-44**”.**

Under the guidance of Prof. Sonal Gore

In the partial fulfillment of the requirements for the T.E. (Computer Engineering)

Prajakta Bhosale (TECOB201)

Varun Gadde (TECOB204)

Chaitanya Nawale (TECOB218)

Siddhi Pardeshi (TECOB224)

Prof. Sonal Gore

Project Guide

INDEX

• Introduction	a. Problem Statement b. Project Idea c. Requirement Analysis
• Project Design	a. H/W , S/W , resources, requirements & their detail explanation b. E-R Model
• Module Description	a. Block diagram with explanation of each module
• Results	a. Source code b. Screen shots including GUI

INTRODUCTION

Event reporting involves producing a **report** after an **event** has concluded in order to assess whether the objectives of the **event** were met. An **event report** should also identify areas for future fine tuning and improvement.

A post-**event report** should **include** a statement of purpose. The reader needs context for the driving force behind the **event**. The statement of purpose should **include** **event** goals.

An event report plays a major role in assessing events held, maintaining curriculum activities and planning future events.

Problem Statement:

To build a web application which generates an event report in proper format on given inputs.

Project Idea:

To make the task of building an event report easy, less time consuming, easy to share and efficient by using appropriate tools and software environment.

Requirement Analysis:

Requirements include accurate data of when and for what event is held, amount of participation in event, no of days event held, event description etc.

- # Project Design

TECHNICAL STACK:

1) 'Visual Studio code' tool for Django framework.
2) MYSQL Database
3) BootStrap
4) HTML

Description:

➢ ***Why Django is the Best Web Framework for Project****?*

- **It's fast and simple.**

Django is an open-source framework for backend web applications based on Python — one of the top web development languages. Its main goals are simplicity, flexibility, reliability, and scalability.
To do that, the Django framework uses:

I. The principles of rapid development, which means developers can do more than one iteration at a time without starting the whole schedule from scratch;

II. DRY philosophy — Don't Repeat Yourself — which means developers can reuse existing code and focus on the unique one.

As a result, it takes a lot less time to get the project done.

➢ **It's secure.**

Security is also a high priority for Django. It has one of the best out-of-the-box security systems out there, and it helps developers avoid common security issues.

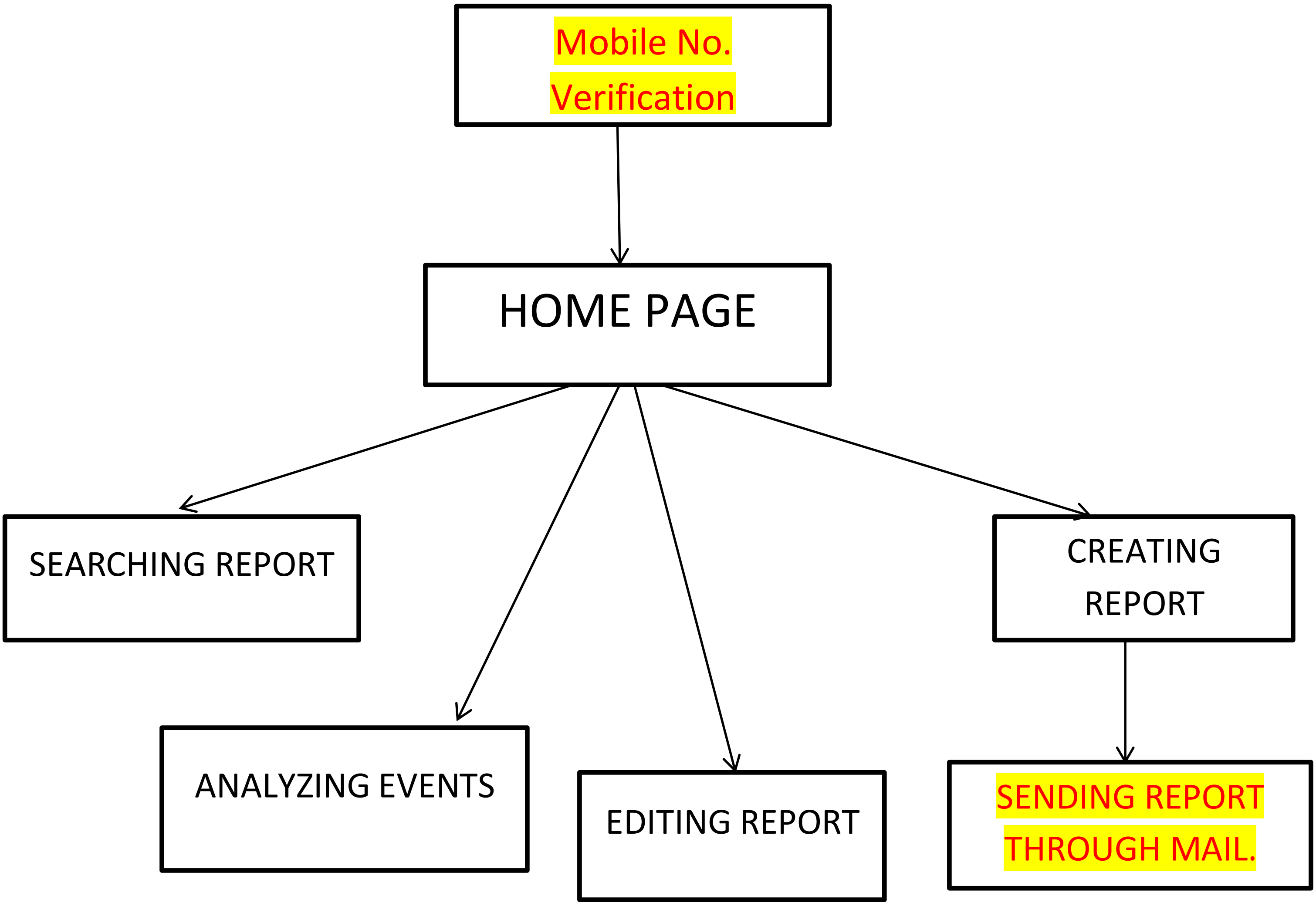
➢ **It suits any web application project.**

With Django, you can tackle projects of any size and capacity, whether it's a simple website or a high-load web application**.**

➢ **It's well-established.**

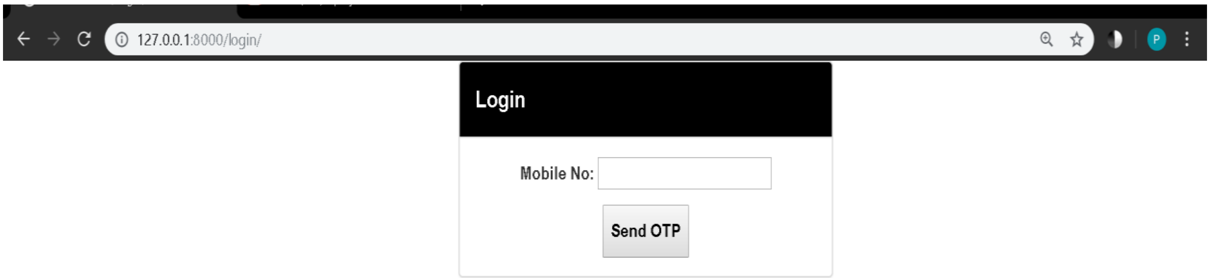
Django is time- and crowd-tested. It has a big, supportive community accessed through numerous forums, channels, and dedicated websites.

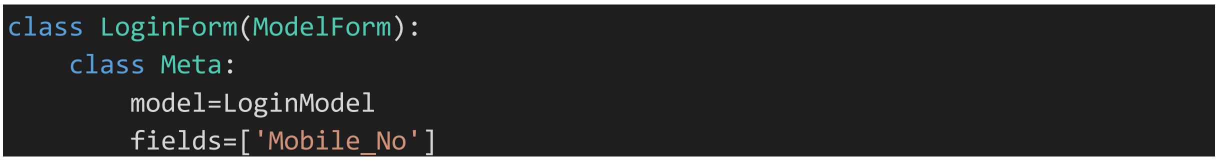
- **FLOW DIAGRAM**



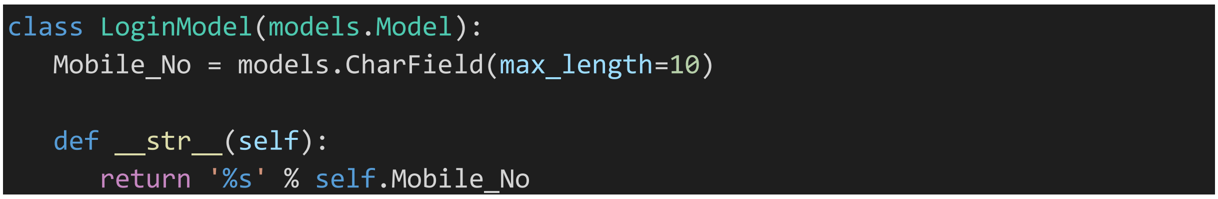
- # **Module Description**

1)Verification of authorised Event Co-ordinator:





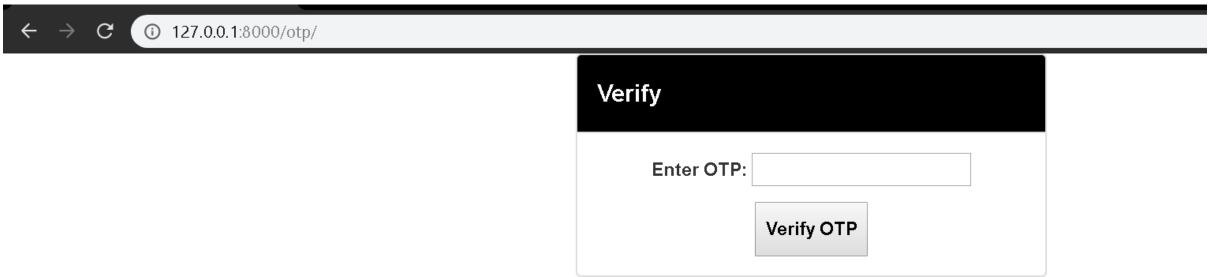
```
class LoginForm(ModelForm):
    class Meta:
        model=LoginModel
        fields=['Mobile_No']
```



```
class LoginModel(models.Model):
   Mobile_No = models.CharField(max_length=10)

   def __str__(self):
      return '%s' % self.Mobile_No
```

In the first page of WebApp an otp will be sent to entered mobile number using way-2-sms facilty.

2)ENTERING THE OTP RECEIVED

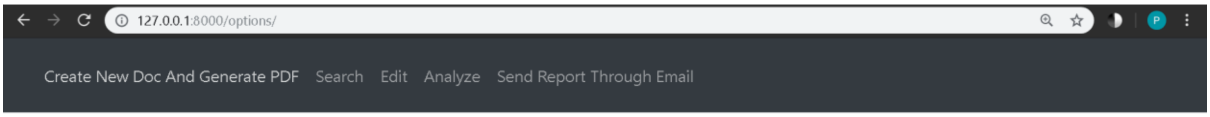


Here received otp on mobile no. is entered and mobile no. is verified.

```
class OTPForm(ModelForm):
    class Meta:
        model=OTPModel
        fields=['Enter_OTP']
```

```
class OTPModel(models.Model):
   Enter_OTP=models.CharField(max_length=4)

   def __str__(self):
      return '%s' % self.Enter_OTP
```

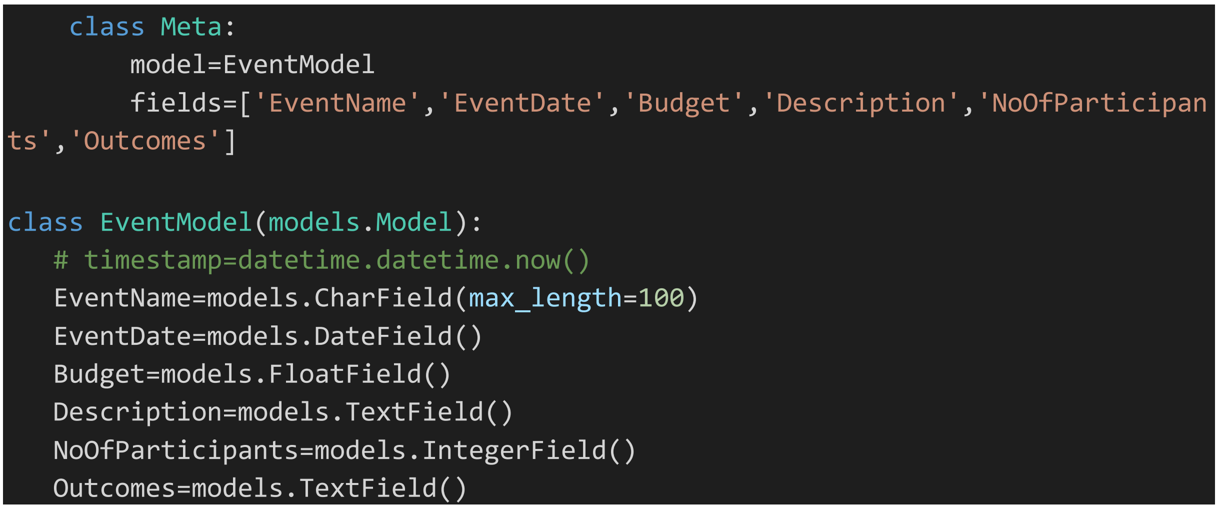
3)**Various Menu Options**



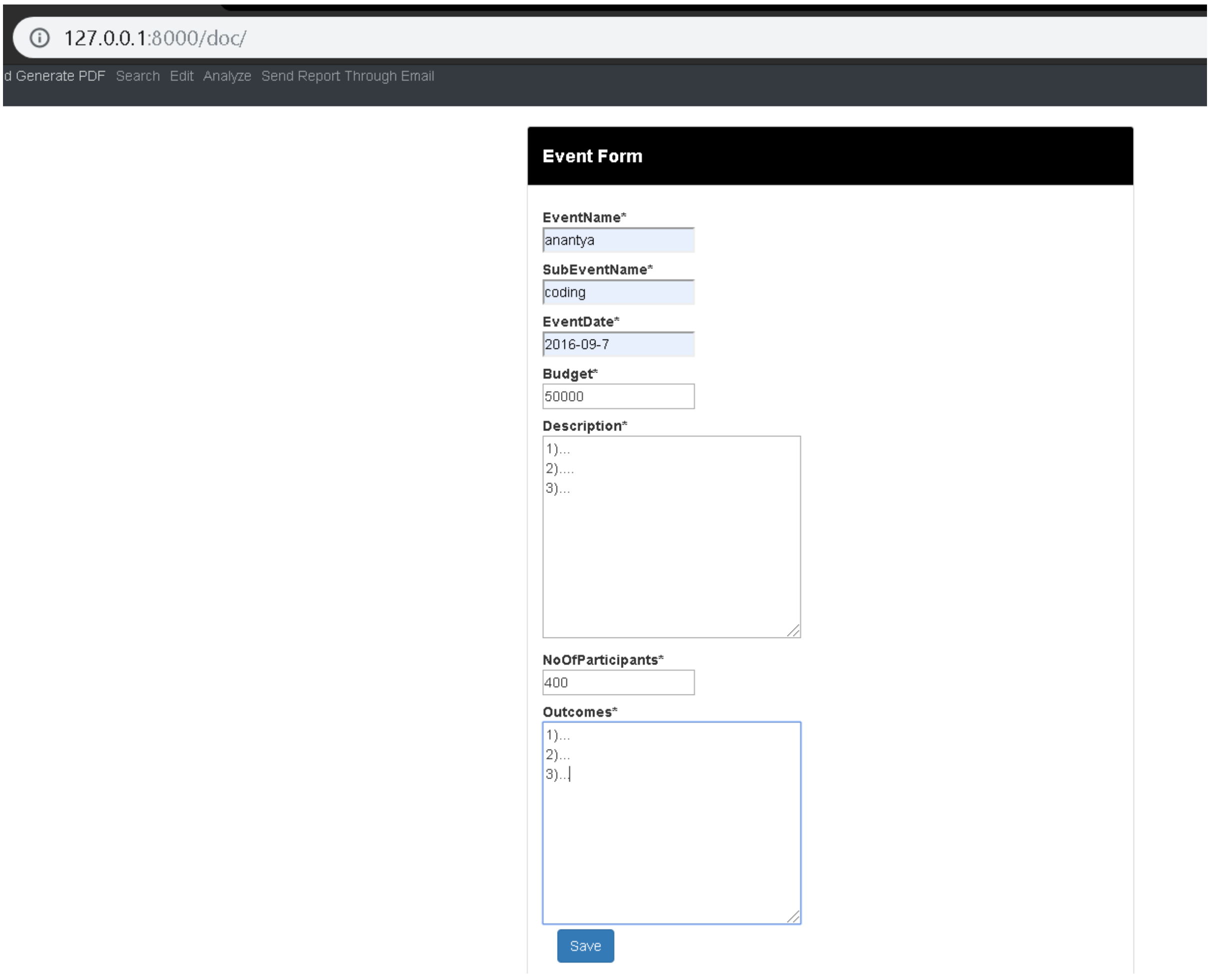
a. Create new Document and Generate it in pdf form.
b. Search an Event.
c. Edit prefilled Event
d. Analyze Events from past to present.
e. Send Event Report Through Mail.

4)**Filling the Event Details**

```
class EventForm(ModelForm):
```



```
    class Meta:
        model=EventModel
        fields=['EventName','EventDate','Budget','Description','NoOfParticipan
ts','Outcomes']

class EventModel(models.Model):
   # timestamp=datetime.datetime.now()
   EventName=models.CharField(max_length=100)
   EventDate=models.DateField()
   Budget=models.FloatField()
   Description=models.TextField()
   NoOfParticipants=models.IntegerField()
   Outcomes=models.TextField()
```



Event Form

EventName*
anantya

SubEventName*
coding

EventDate*
2016-09-7

Budget*
50000

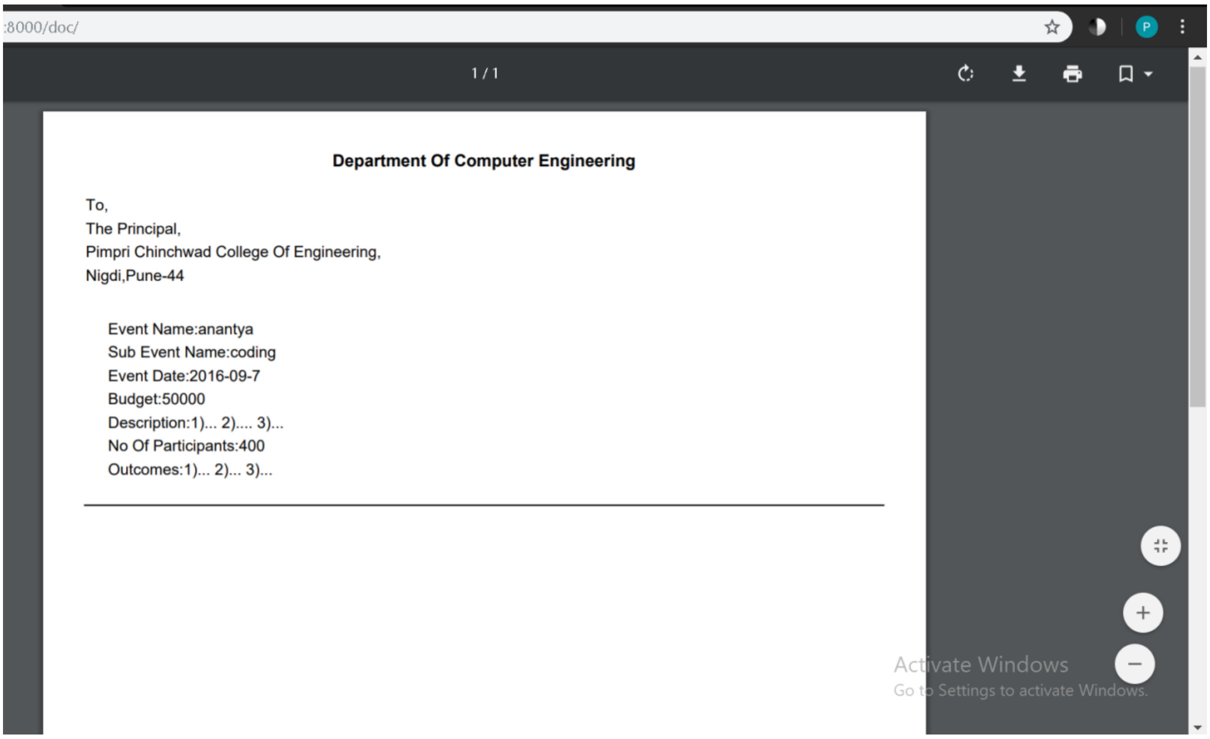
Description*
1)...
2)....
3)...

NoOfParticipants*
400

Outcomes*
1)...
2)...
3)...

Event details are filled in respective fields and save button is clicked after entering all the details.

5)**Report Generation in pdf form:**

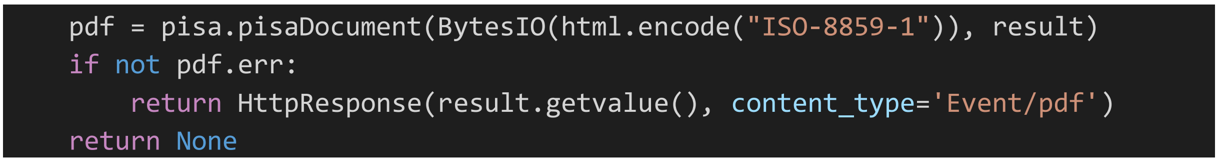


Department Of Computer Engineering

To,
The Principal,
Pimpri Chinchwad College Of Engineering,
Nigdi,Pune-44

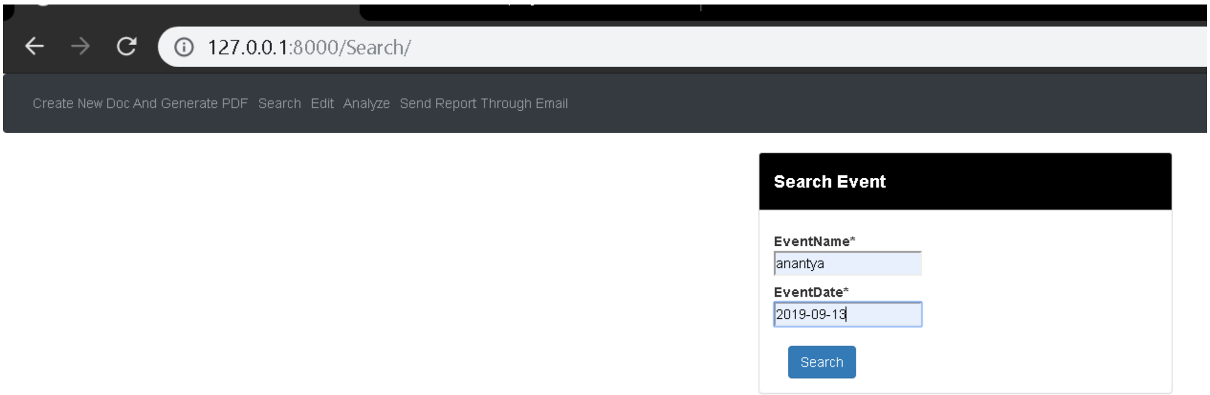
Event Name:anantya
Sub Event Name:coding
Event Date:2016-09-7
Budget:50000
Description:1)... 2).... 3)...
No Of Participants:400
Outcomes:1)... 2)... 3)...

```
from io import BytesIO
from django.http import HttpResponse
from django.template.loader import get_template

from xhtml2pdf import pisa

def render_to_pdf(template_src, context_dict={}):
    template = get_template(template_src)
    html  = template.render(context_dict)
    result = BytesIO()
```



```
    pdf = pisa.pisaDocument(BytesIO(html.encode("ISO-8859-1")), result)
    if not pdf.err:
        return HttpResponse(result.getvalue(), content_type='Event/pdf')
    return None
```

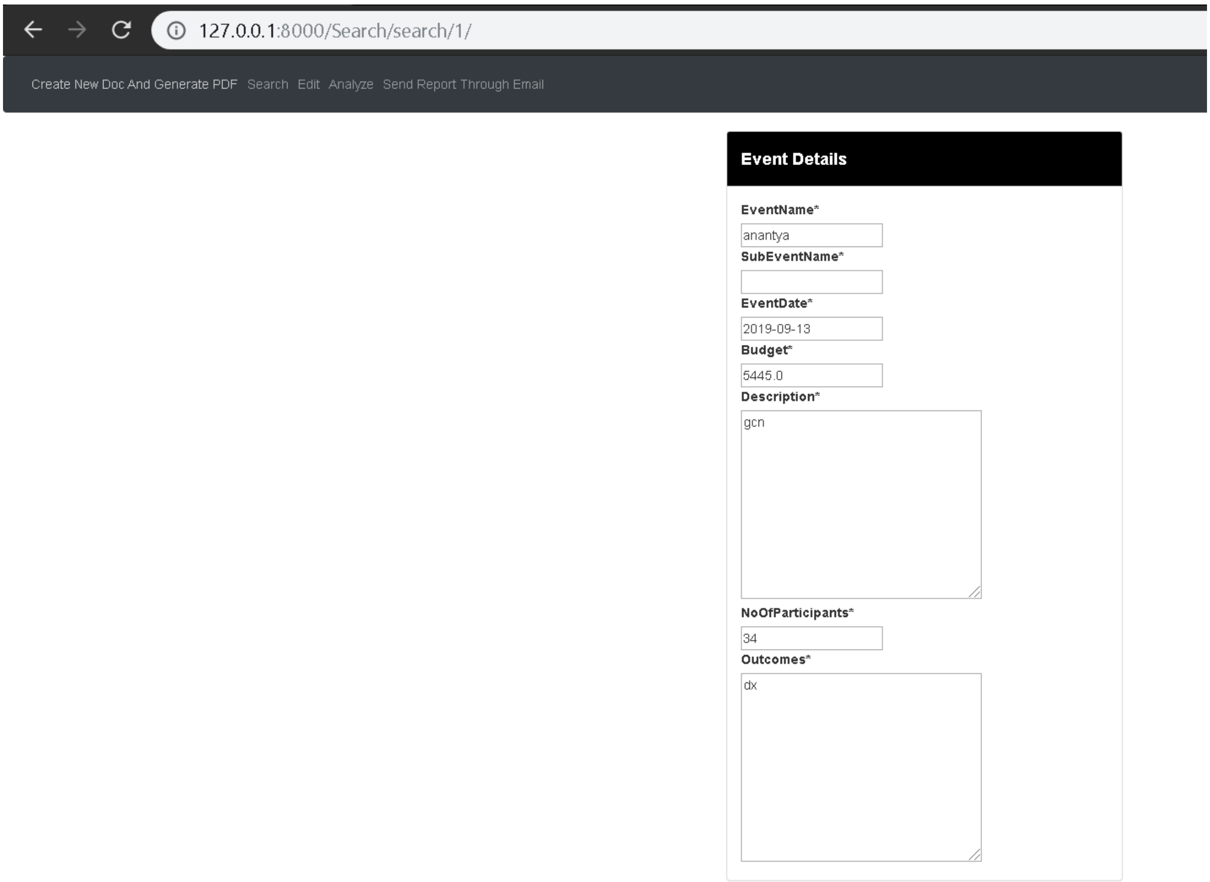
6)**Searching a prefilled Event Details:**



Search Event

EventName* anantya

EventDate* 2019-09-13


127.0.0.1:8000/Search/search/1/
Create New Doc And Generate PDF
Search
Edit
Analyze
Send Report Through Email

Event Details

EventName* anantya

SubEventName*

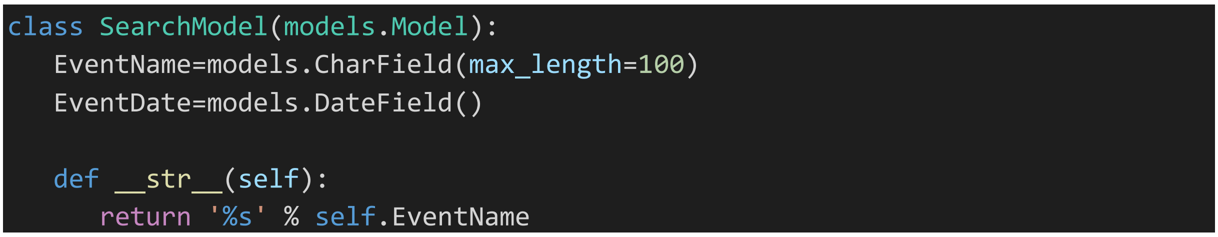
EventDate* 2019-09-13

Budget* 5445.0

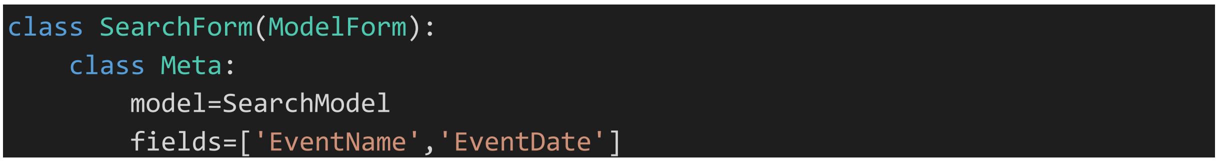
Description* gcn

NoOfParticipants* 34

Outcomes* dx

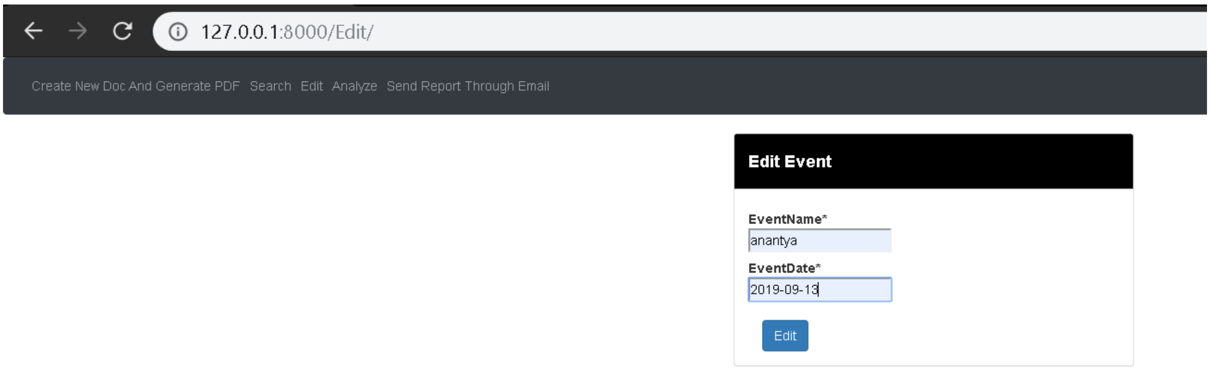


```python
class SearchModel(models.Model):
   EventName=models.CharField(max_length=100)
   EventDate=models.DateField()

   def __str__(self):
      return '%s' % self.EventName
```



```python
class SearchForm(ModelForm):
    class Meta:
        model=SearchModel
        fields=['EventName','EventDate']
```

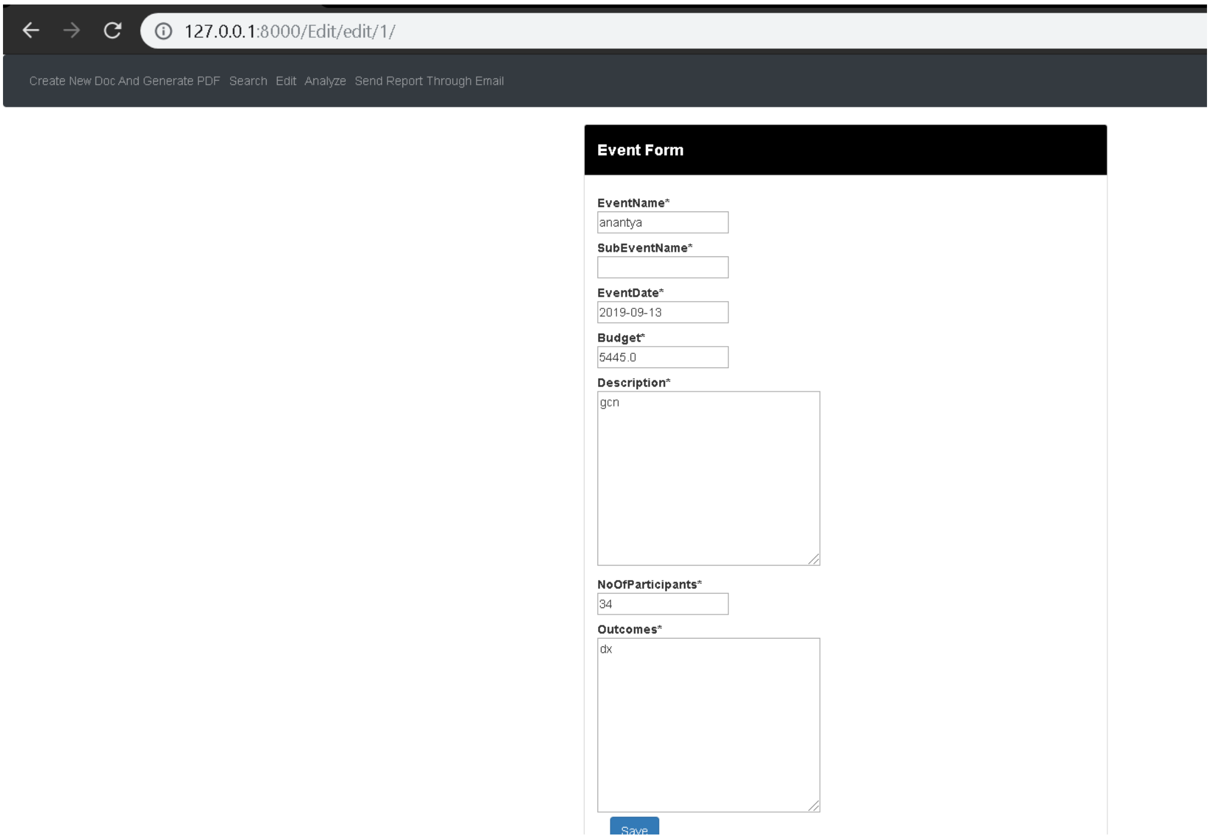
6) Editng prefilled Event Details:



Edit Event

EventName* anantya

EventDate* 2019-09-13


127.0.0.1:8000/Edit/edit/1/
Create New Doc And Generate PDF
Search
Edit
Analyze
Send Report Through Email
Save

Event Form

EventName* anantya

SubEventName*

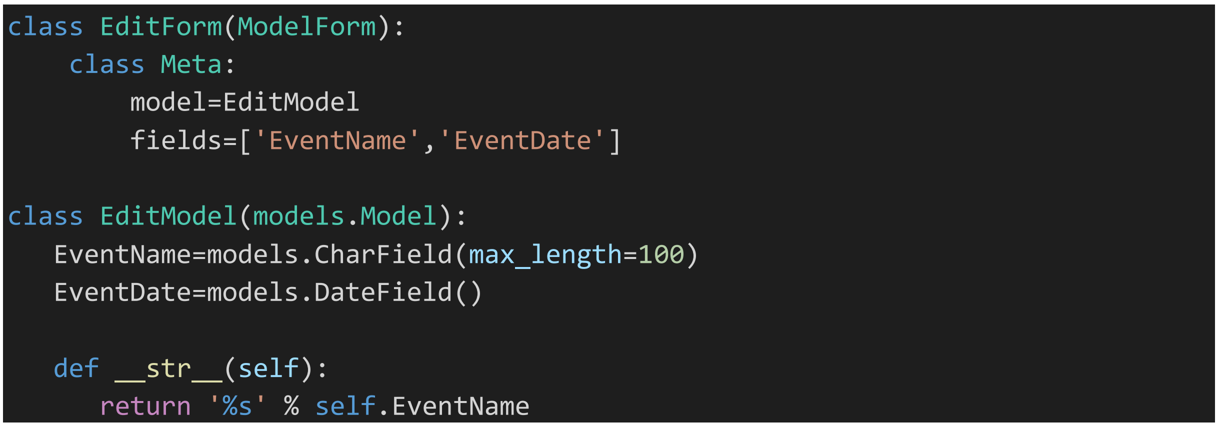
EventDate* 2019-09-13

Budget* 5445.0

Description* gcn

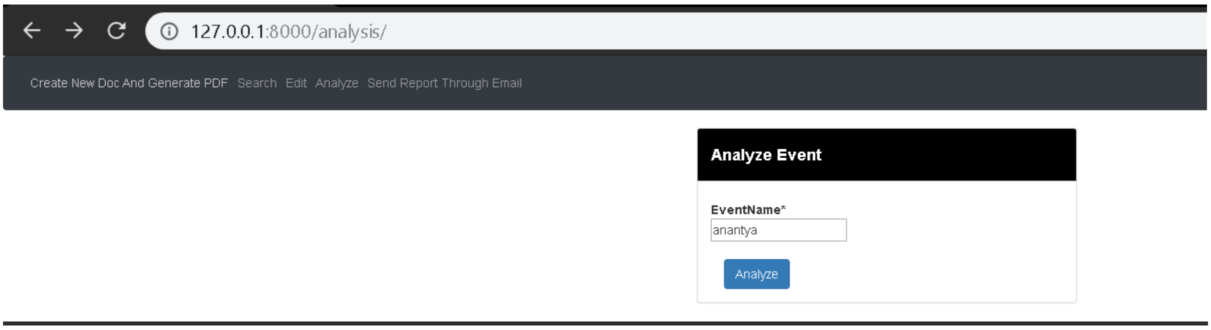
NoOfParticipants* 34

Outcomes* dx



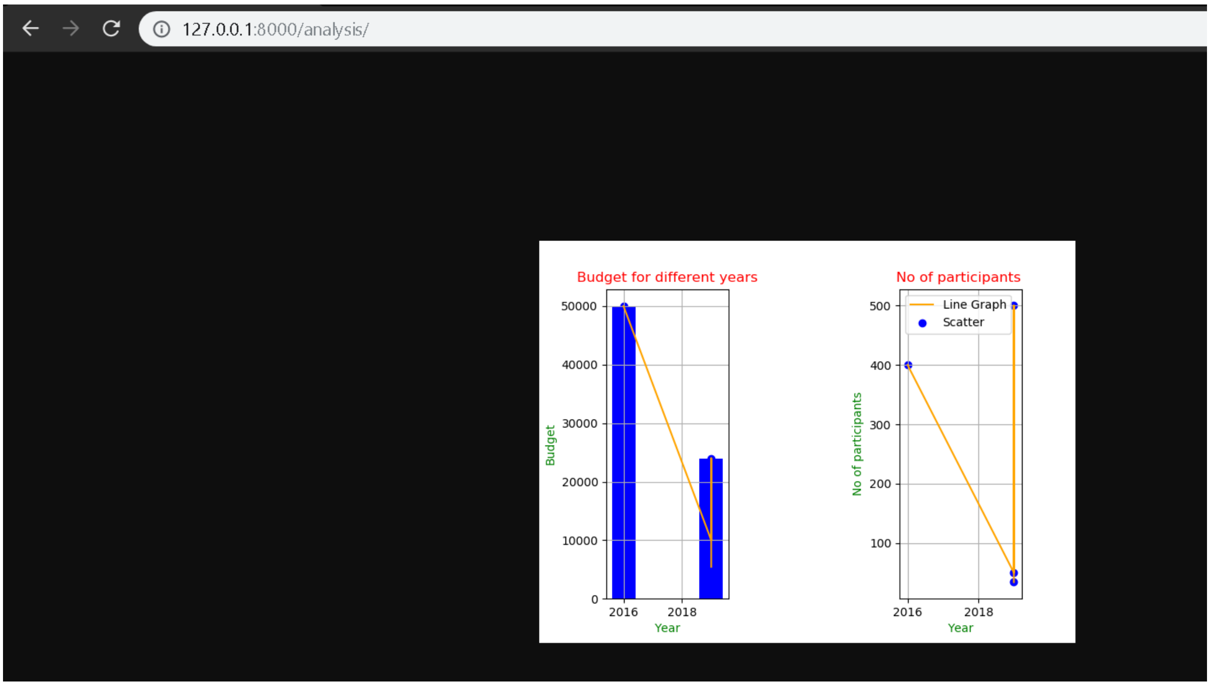
```
class EditForm(ModelForm):
    class Meta:
        model=EditModel
        fields=['EventName','EventDate']

class EditModel(models.Model):
   EventName=models.CharField(max_length=100)
   EventDate=models.DateField()

   def __str__(self):
      return '%s' % self.EventName
```

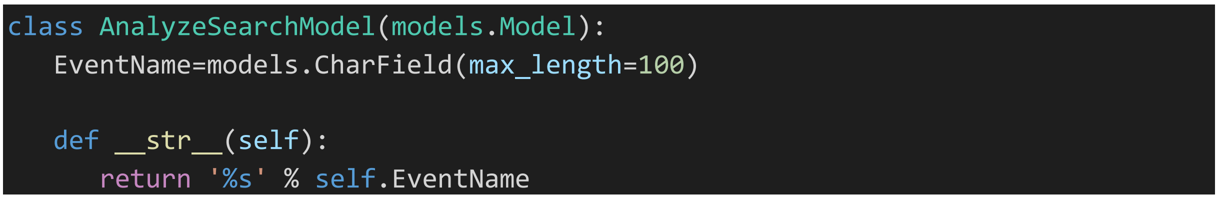
7) Editng prefilled Event Details:



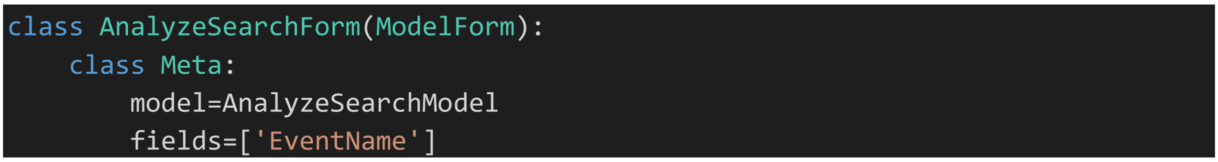
Analyze Event

EventName* anantya



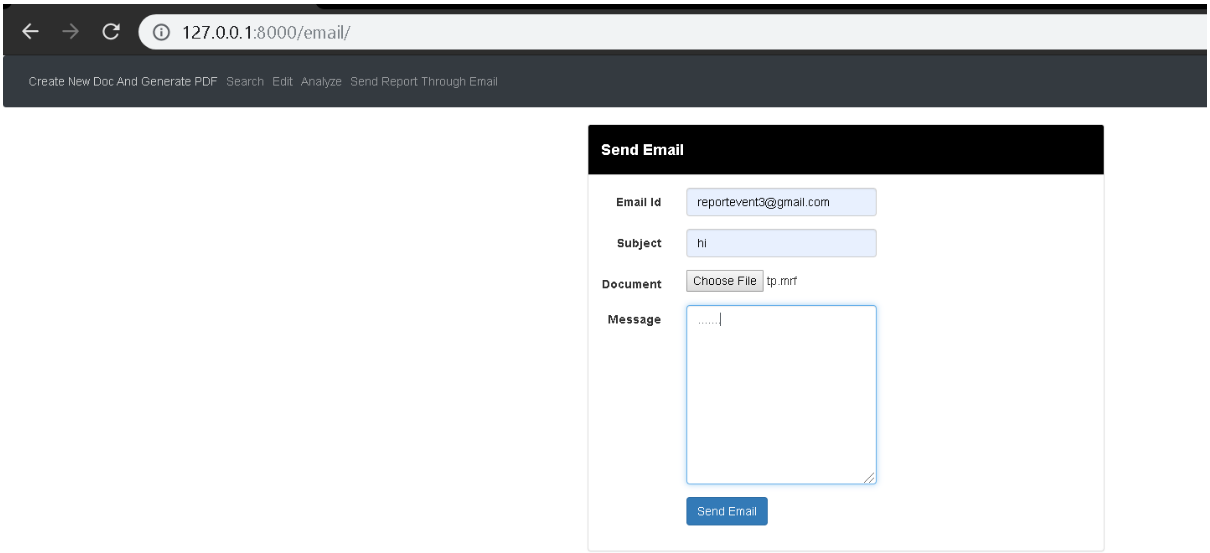


```
class AnalyzeSearchModel(models.Model):
   EventName=models.CharField(max_length=100)

   def __str__(self):
      return '%s' % self.EventName
```



```
class AnalyzeSearchForm(ModelForm):
    class Meta:
        model=AnalyzeSearchModel
        fields=['EventName']
```

8) <u>Sending Event Details through mail:</u>



Email Id: reportevent3@gmail.com

Subject: hi

Document: tp.mrf

Message: ......|

```
def email(request):
    if request.method == "POST":
        form = EmailForm(request.POST,request.FILES)
        if form.is_valid():
            post = form.save(commit=False)
            post.published_date = timezone.now()
            post.save()
            email = request.POST.get('email')
            subject = request.POST.get('subject')
            message = request.POST.get('message')
            document = request.FILES.get('document')
            email_from = settings.EMAIL_HOST_USER
            recipient_list = [email]
            email = EmailMessage(subject,message,email_from,recipient_list)
            base_dir = 'media/documents/'
            email.attach_file('media/documents/'+str(document))
            email.send()
    else:
        form = EmailForm()

    return render(request, 'sendemail.html', {'form': form})
```

```
class EmailForm(forms.ModelForm):
    email = forms.EmailField(max_length=200,widget=forms.TextInput(attrs={'class': "form-control",'id': "clientemail"}))
    message = forms.CharField( widget=forms.Textarea(attrs={'class': "form-control"}))
    subject = forms.CharField( widget=forms.TextInput(attrs={'class': "form-control"}))
    class Meta:
        model = Mails
        fields = ('email','subject','message','document',)
```

➢ Role of MYSQL DATABASE

1) Establishing connectivity between Django and Mysql.

```
➢ DATABASES = {
➢     'default': {
➢         'ENGINE': 'django.db.backends.mysql',
➢         'NAME': 'reportevent',
➢         'USER':'root',
➢         'PASSWORD':'9881487034',
➢         'HOST':'localhost',
➢         'PORT':3306
➢     }
➢ }
```

In settings.py username and password of mysql is provided.

2) Queries used to retrieve and manipulate data in MYSQL database.

a. To search an Event:.

```
 searched_data = EventModel.objects.raw('SELECT * FROM eve
nt_eventmodel WHERE EventName=%s AND EventDate=%s',[n,d])[
0]
```

Data retrieval from table

b. To edit an event:

- Searching the event details first:

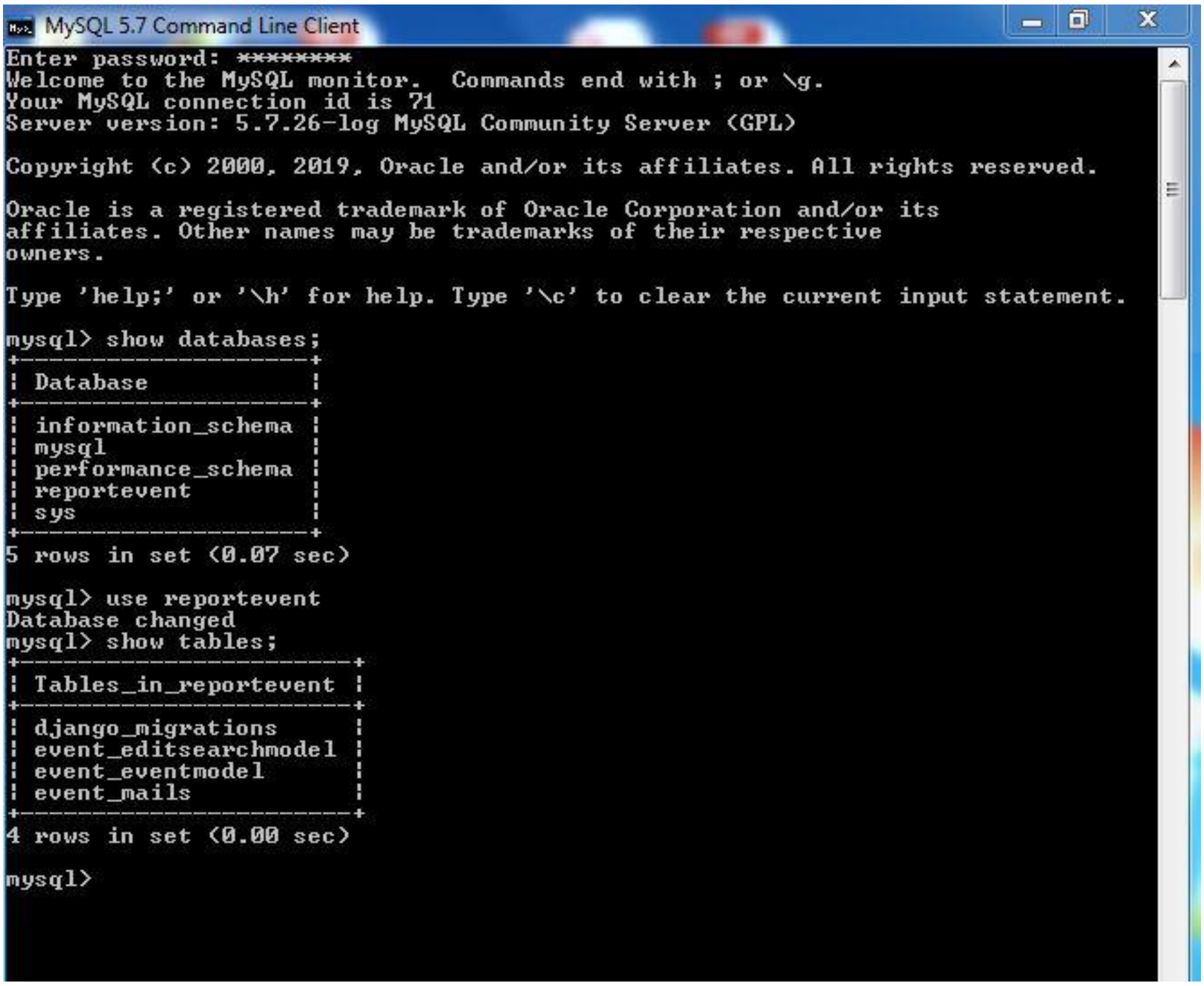
```
 searched_data = EventModel.objects.raw('SELECT * FROM eve
nt_eventmodel WHERE EventName=%s AND EventDate=%s',[n,d])[
0]
```

- Updating the edited values.

```
cursor.execute('UPDATE event_eventmodel SET EventName=%s,E
ventDate=%s,Budget=%s,Description=%s,NoOfParticipants=%s,O
utcomes=%s WHERE id=%s',[name1,date1,budg,desc,no,outc,id]
)
```

- **SQL Interface:**

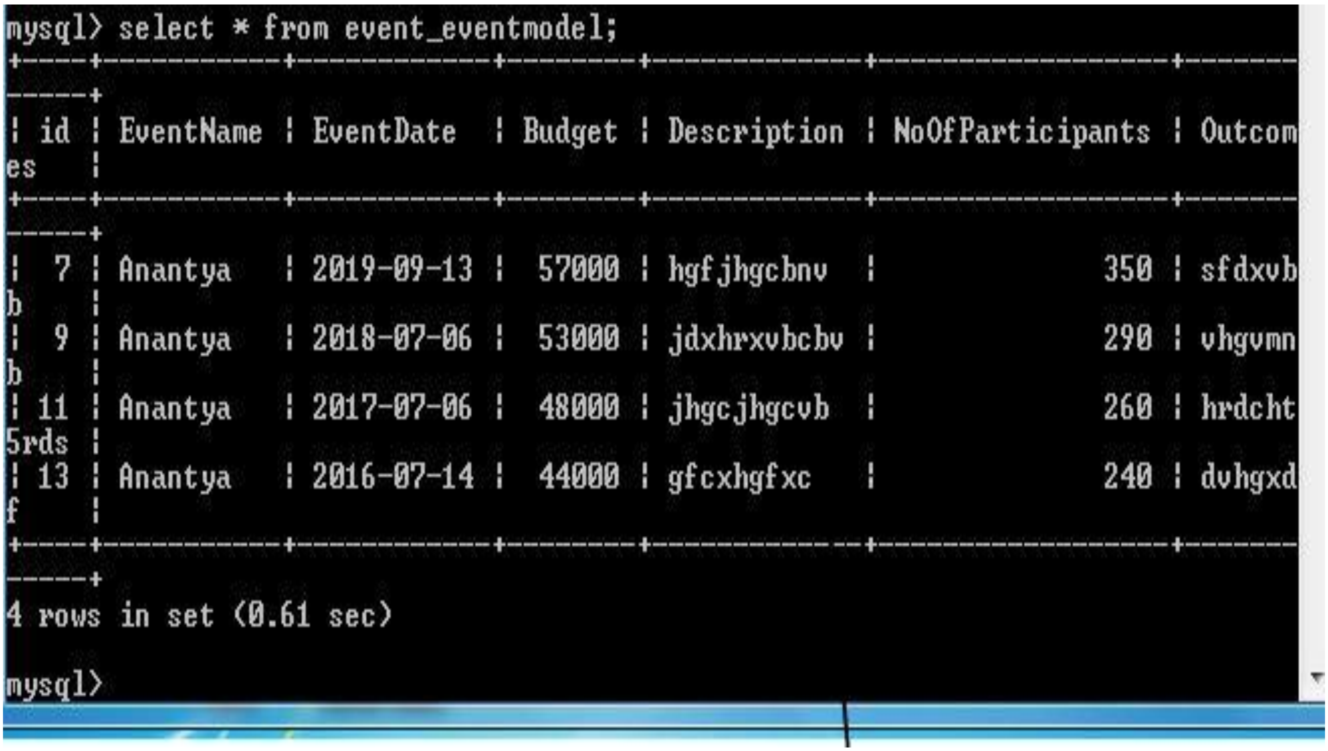
a.Using reportevent database.



```
Enter password: ********
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 71
Server version: 5.7.26-log MySQL Community Server (GPL)

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| reportevent        |
| sys                |
+--------------------+
5 rows in set (0.07 sec)

mysql> use reportevent
Database changed
mysql> show tables;
+-----------------------+
| Tables_in_reportevent |
+-----------------------+
| django_migrations     |
| event_editsearchmodel |
| event_eventmodel      |
| event_mails           |
+-----------------------+
4 rows in set (0.00 sec)

mysql>
```

b. Table event_eventmodel contents.

The entered contents in Event report form are stored in sql table having tablename event_eventmodel with the help of save().



```
mysql> select * from event_eventmodel;
+----+-----------+------------+--------+-------------+------------------+-------
-----+
¦ id ¦ EventName ¦ EventDate  ¦ Budget ¦ Description ¦ NoOfParticipants ¦ Outcom
es   ¦
+----+-----------+------------+--------+-------------+------------------+-------
-----+
¦  7 ¦ Anantya   ¦ 2019-09-13 ¦  57000 ¦ hgfjhgcbnv  ¦              350 ¦ sfdxvb
b    ¦
¦  9 ¦ Anantya   ¦ 2018-07-06 ¦  53000 ¦ jdxhrxvbcbv ¦              290 ¦ vhgvmn
b    ¦
¦ 11 ¦ Anantya   ¦ 2017-07-06 ¦  48000 ¦ jhgcjhgcvb  ¦              260 ¦ hrdcht
5rds ¦
¦ 13 ¦ Anantya   ¦ 2016-07-14 ¦  44000 ¦ gfcxhgfxc   ¦              240 ¦ dvhgxd
f    ¦
+----+-----------+------------+--------+-------------+------------------+-------
-----+
4 rows in set (0.61 sec)

mysql>
```

Hence Mysql database plays a major role in manipulating the data of Event report.

- **Conclusion:**

  Thus we have successfully implemented a web application to build an Event Report with less Efforts.