1. **Created the new Django project:** I created a new Django project by running the following command:

    - **django-admin startproject eventmanagement**
    - **Python manage.py startapp event**

    - **Django-based for an event management system.**

    - This command will create a new directory named ‘eventmanagement’, which will contain the basic structure of a Django project.

2. **Created new apps:** A Django project can have apps, each containing a specific set of functionalities. I created a new app by running the following command:

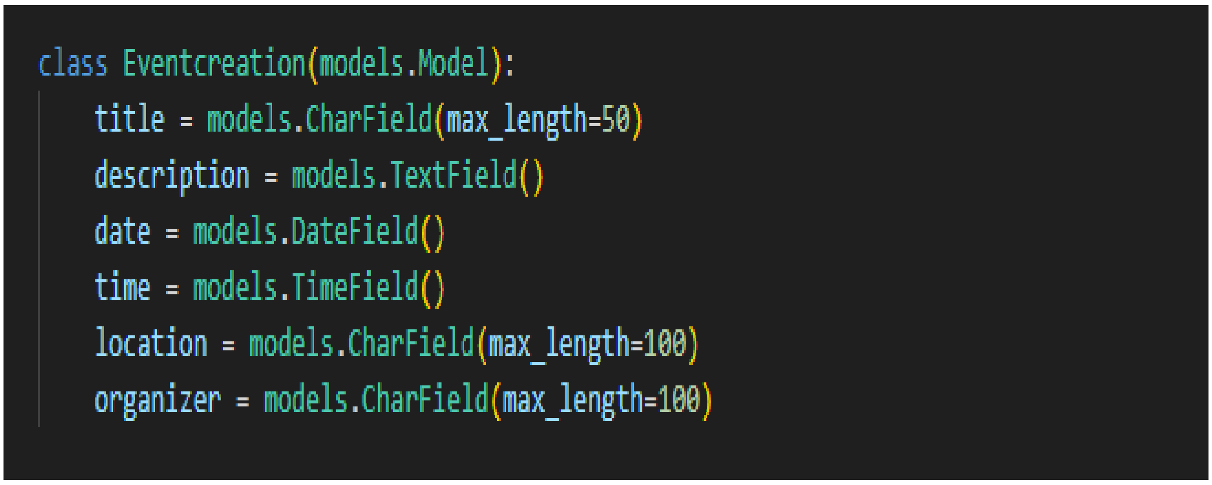
    - **python manage.py startapp event**

    - These commands will create a new directory, which will contain the basic structure of a Django app.

3. **Defined models:** In Django, models define the structure of your database tables. I define the models in the 'models.py' file of the app.

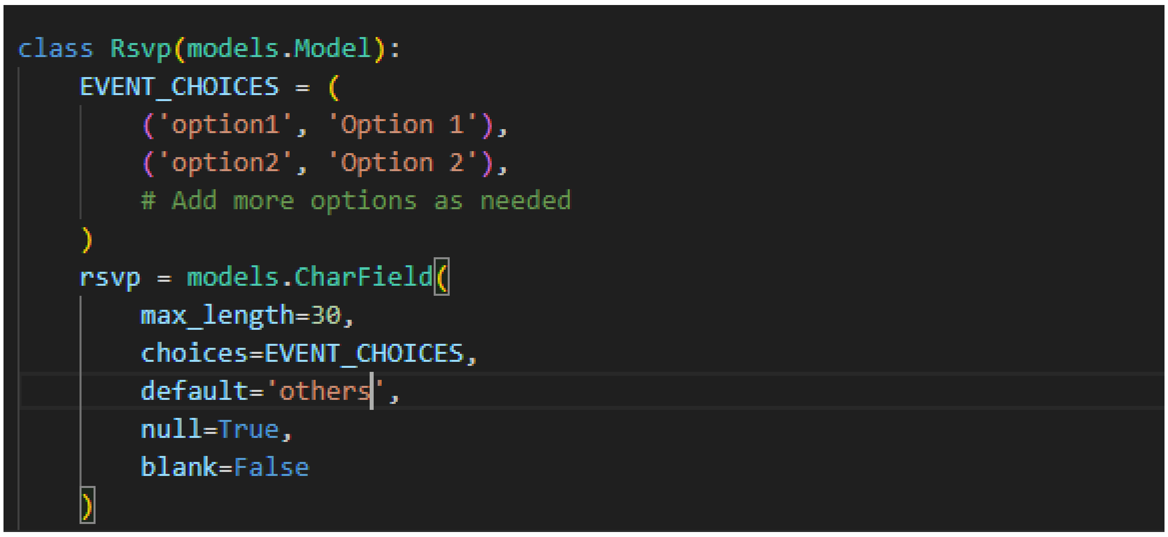
    - For this project, i needed to define two models:

        a. **Event Model:** This model will have many fields 'title', ‘description’, ‘date’, ‘time’, ‘location’, ‘organiser’ which will store the event model table.



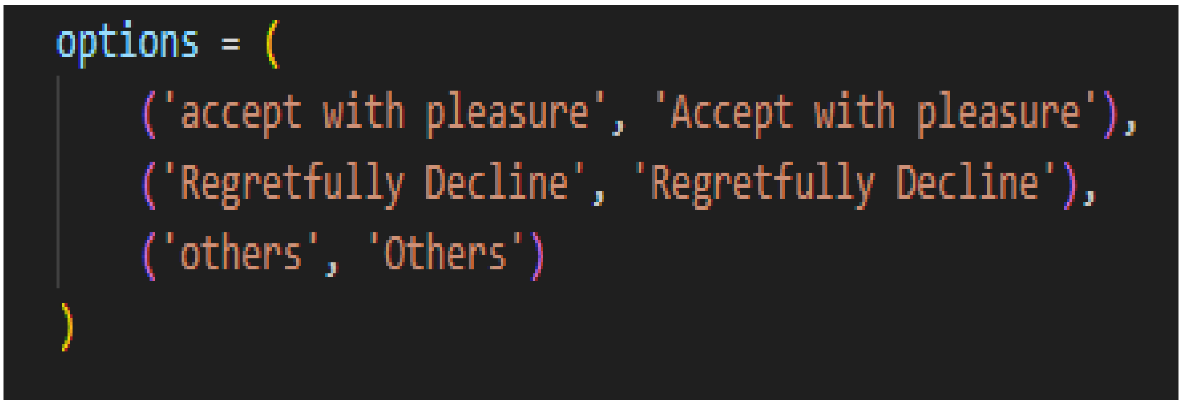
```
class Eventcreation(models.Model):
    title = models.CharField(max_length=50)
    description = models.TextField()
    date = models.DateField()
    time = models.TimeField()
    location = models.CharField(max_length=100)
    organizer = models.CharField(max_length=100)
```

b. **rsvp model:** This model will have the following fields: Event choice, rsvp.



```python
class Rsvp(models.Model):
    EVENT_CHOICES = (
        ('option1', 'Option 1'),
        ('option2', 'Option 2'),
        # Add more options as needed
    )
    rsvp = models.CharField(
        max_length=30,
        choices=EVENT_CHOICES,
        default='others',
        null=True,
        blank=False
    )
```

c. **options model:** This model will have following fields: accept with pleasure, regretfully decline and by default others.

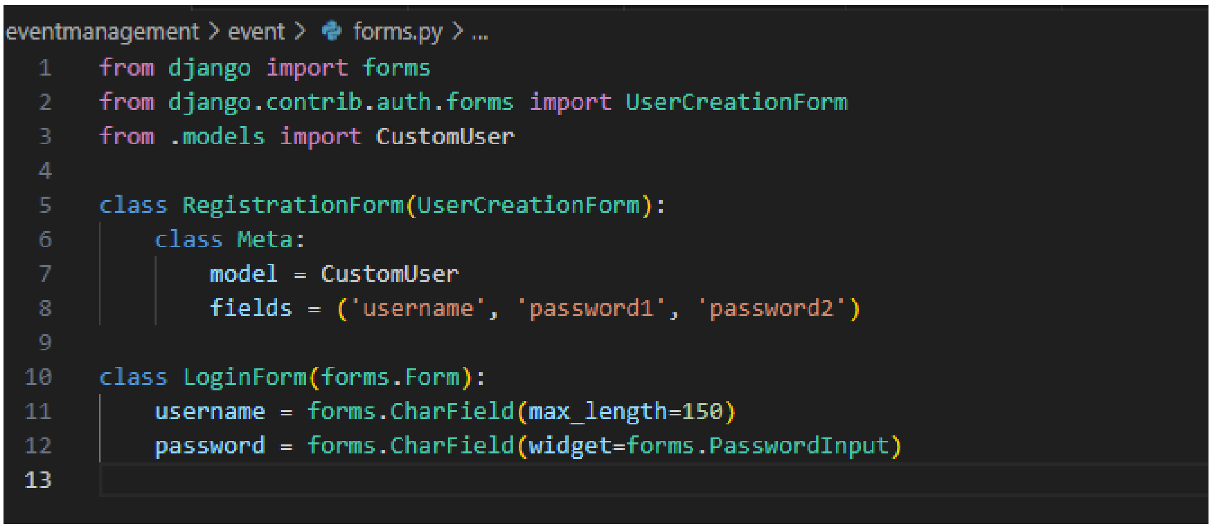


```python
options = (
    ('accept with pleasure', 'Accept with pleasure'),
    ('Regretfully Decline', 'Regretfully Decline'),
    ('others', 'Others')
)
```

4. **Created database tables:** After defining the models, I needed to create the corresponding database tables. I did this by running the following command:

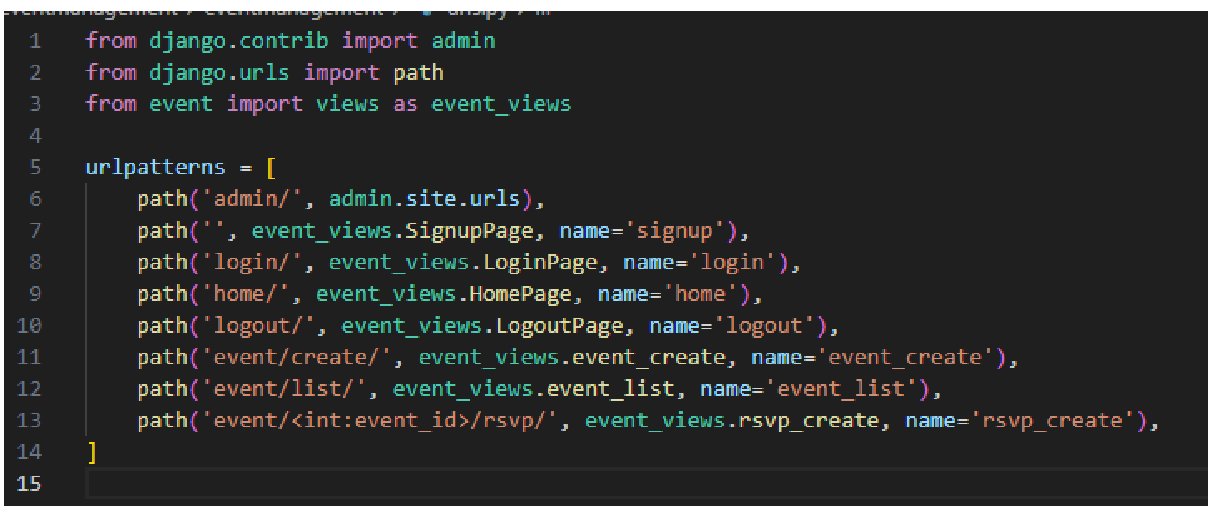
   - **python manage.py makemigrations**
   - **python manage.py migrate**

5. **Forms.py -** This is my forms.py is responsible for registration and signup page


eventmanagement > event > forms.py > ...

```python
from django import forms
from django.contrib.auth.forms import UserCreationForm
from .models import CustomUser

class RegistrationForm(UserCreationForm):
    class Meta:
        model = CustomUser
        fields = ('username', 'password1', 'password2')

class LoginForm(forms.Form):
    username = forms.CharField(max_length=150)
    password = forms.CharField(widget=forms.PasswordInput)

```

7. Urls.py - There are following end points such as

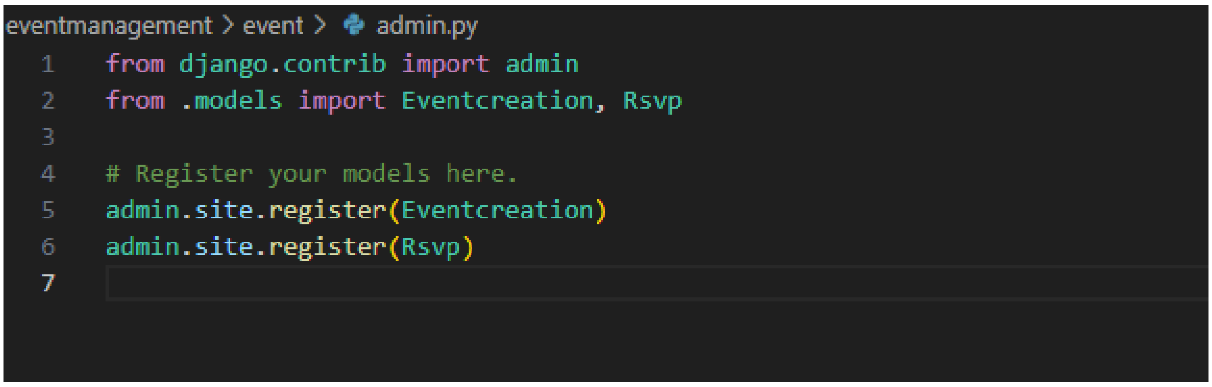


```python
from django.contrib import admin
from django.urls import path
from event import views as event_views

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', event_views.SignupPage, name='signup'),
    path('login/', event_views.LoginPage, name='login'),
    path('home/', event_views.HomePage, name='home'),
    path('logout/', event_views.LogoutPage, name='logout'),
    path('event/create/', event_views.event_create, name='event_create'),
    path('event/list/', event_views.event_list, name='event_list'),
    path('event/<int:event_id>/rsvp/', event_views.rsvp_create, name='rsvp_create'),
]

```

- At admin urls admin can see and add the events.
- At http://127.0.0.1:8000 registration page is shown.
- At the login endpoint login form is rendered.

- At the home endpoint the home page is rendered.
- At event/create/ event creation form rendered.
- At event/list/ all the events are shown.

8. Admin.py : The admin.py file is used to register models and customise the admin interface for those models.



eventmanagement > event > admin.py

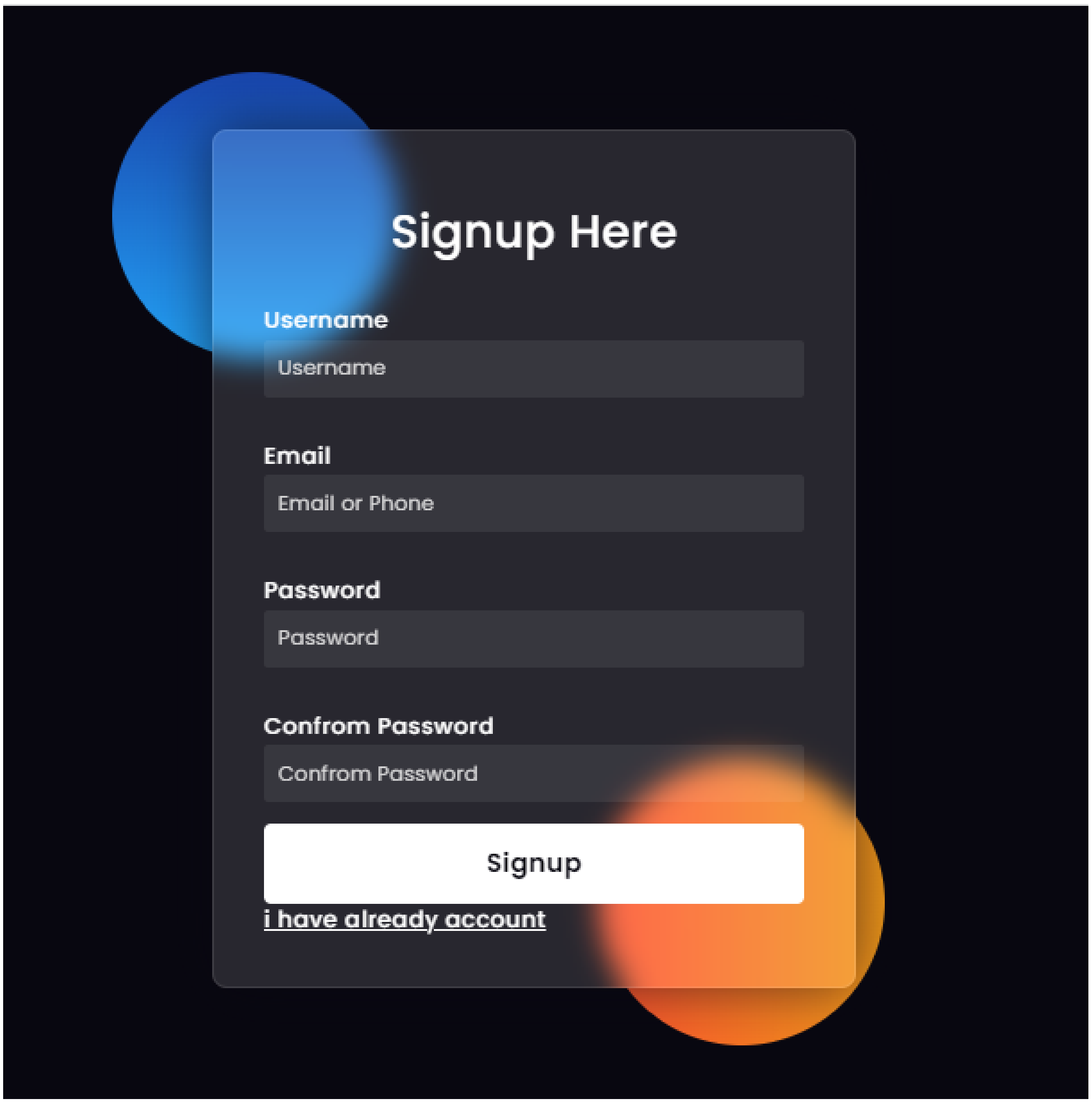
```python
from django.contrib import admin
from .models import Eventcreation, Rsvp

# Register your models here.
admin.site.register(Eventcreation)
admin.site.register(Rsvp)
```

10. Eventmanagement file : This is the inner project folder.

Settings.py file : This is the main file of the whole project. All the register apps are present in settings.py file and database connectivity is also defined in this file.
Database connectivity

```python
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'event',
        'USER': 'root',
        'PASSWORD': 'Admin@123',
        'HOST': '127.0.0.1',
        'PORT': '3306',
        'OPTIONS': {
            'init_command': "SET sql_mode='STRICT_TRANS_TABLES'"
        }
    }
}
```

Registration page - This is registration page username, email, password and confirm password is required for the registration.


Signup Here

Username

Username

Email

Email or Phone

Password

Password

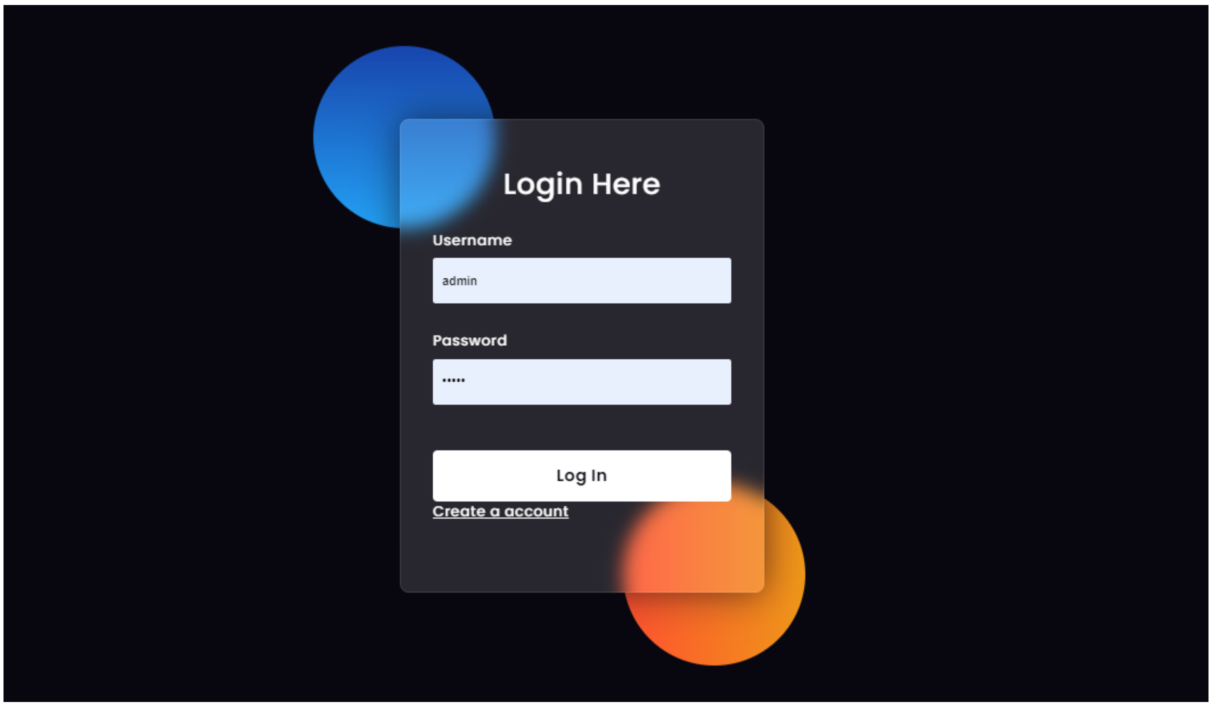
Confrom Password

Confrom Password

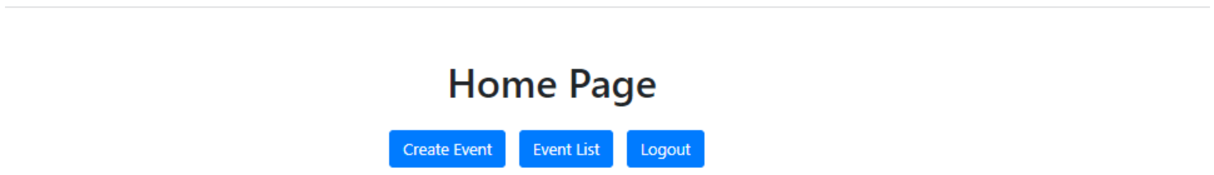
Signup

i have already account

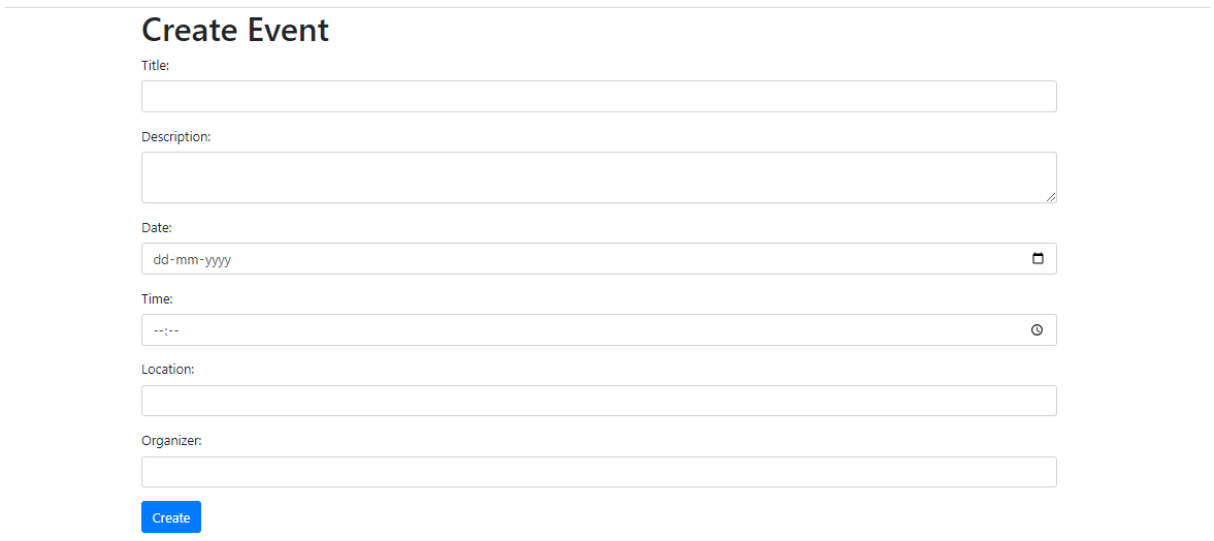
Login page- This is login page for registered users in login form user put the username and password and click the login then he/she render to the home page.



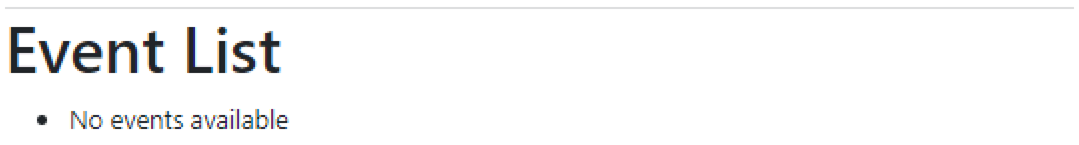
Home page - Home page has three buttons: create event, event list, logout.



Create event - Create event has following fields such as title, description, date, times, location and organiser.



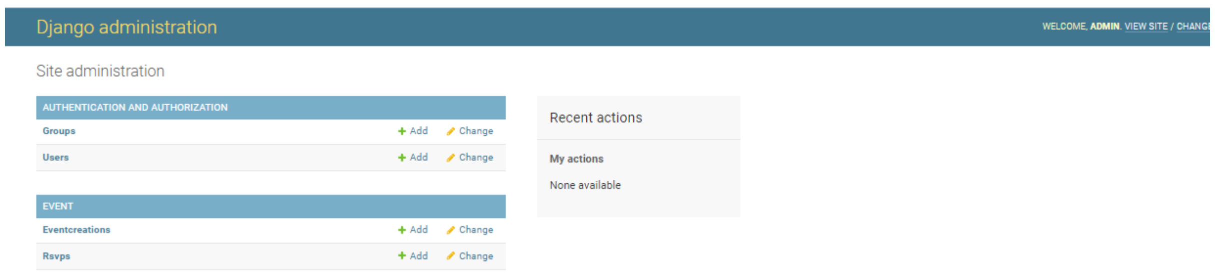
Event list - Event list shows the list of events.



Event List

- No events available

Admin Panel - Admin can see the event and rsvp.



Django administration

WELCOME, ADMIN. VIEW SITE / CHANG

Site administration

AUTHENTICATION AND AUTHORIZATION		
Groups	+ Add	Change
Users	+ Add	Change

EVENT		
Eventcreations	+ Add	Change
Rsvps	+ Add	Change

Recent actions

My actions

None available