

**PROJECT DOCUMENTATION ON  
ORPHANAGE MANAGEMENT SYSTEM**



**GROUP - 31**  
*BACHELOR OF TECHNOLOGY B. TECH*  
**(COMPUTER SCIENCE & ENGINEERING)**



**INDIAN INSTITUTE OF INFORMATION TECHNOLOGY  
SRICITY.**

## **TEAM MEMBERS OF WAD GROUP – 31**

- BHARATH KAMASANI – S20190010080(LEADER)**
- ROKESH REDDY KONGAMUDI – S20190010098**
- HARSSHATH MATCHA – S20190010116**
- PRABHU TEJA SABHAVAT – S20190010155**
- GOVARDHAN NAIK PATHLAVATH – S20190010135**

## **ACKNOWLEDGEMENTS**

My sincerest thanks to the WAD FACULTY whose Grace and wisdom has been sufficient for us without him I couldn't have gotten this far. I also own special debts in many ways to those who helped in diverse ways in making this project work a reality.

My profound gratitude also goes to WAD FACULTY (DR. BALAJI RAMAN, DR.GIRISH.GN, DR. HIMANGSHU SARMA, DR. BHEEMAPPA HALAVAR, DR. RAKESH KUMAR SANODIYA), who were objective but quite supportive and offered valuable suggestions, guidance and moral support.

Also to my Honorable Director Prof. GANESAN KANNABIRAN for the moral support, mentorship and efforts.

May God bless you all.

# **DOCUMENTATION**

## **PROBLEM STATEMENT:**

Orphans are the children who lost their parents or abounded from the family. The orphans face very challenging conditions in the society. orphanage provides very good environment to the orphans by providing good quality education and decent food. But all orphanages can't afford everything to orphans. Food can't be served to all children in the orphanage or small amount of food is distributed. due to this, children are not getting sufficient healthy food. this will definitely affect their physical and mental growth. Children should spend their childhood happily and healthily. this reflects their behavior and character of future. all the time orphanage people can't provide the clothes and toys to all children. In everyone's life birthday is a special day for children its a very big thing. But in orphanage food and education combinedly is a big problem to orphanage people. So they can't afford for this celebrations. Orphanage people can't monitor and review each individually child in the orphanage. So many people outside in the world are willing to donate and help orphan children. But they don't know the way to donate amount, sponsor a child and how to share their old clothes and toys. And there are some people how are willing to do volunteer works in orphanages. But due to lack of guidelines they are also not volunteering.

## **PROBLEM SOLUTION:**

So to overcome the Above discussed problems our team decided to build a web application which can help the orphanages and support the children and guides the donors. And also instruct the process of adoption to adopters. This Orphan Home Management System will help to improve the performance of current situation and overcome the problems that arise nowadays. We collaborate with a set of orphanage homes in a specific city and we will work for them.

How to run:

First you need to download and install Django :

Steps for installing is:

- 1.open terminal and install pip
- 2.now create a folder Django .in which we will create an environment for Django to work
- 3.In terminal go to that folder and run this command “python -m pip install Django”.
- 4.Now ,your django environment is created. You will be given a file [myproject.zip] file .Download it and extract it. And keep it in this django folder.
- 5.Now open Vscode and open this myproject folder in that .
- 6.open the terminal in vscode and run these commands:

“Python manage.py makemigrations”

“Python manage.py sqlmigrate <Name of the App> 0001”

“Python manage.py migrate”

“Python mange.py runserver”

7.now the server has began and it's time to get into website.

Open browser and type “<http://127.0.0.1:8000>” in the search bar. From that point you are on our website.

## SYSTEM SPECIFICATION:

SOFWARE REQUIREMENT SPECIFICATION:

OPERATING SYSTEM: WINDOWS

FRONT END: HTML, CSS, Java Script

BACK END: MY SQL(PostgreSQL), PYTHON ,DJANGO.

## HARDWARE SPECIFICATION:

The following are the hardware devices that will be needed for the system

- A personal computer (Core i3, 1.70GHz or higher recommendation)
- Minimum RAM of 2GB

➤ Uninterrupted Power Supply (UPS)

➤ Printer.

## **SOFTWARE DESCRIPTION:**

➤ Operating System: Windows Xp, Windows 7, 8, 10

➤ Browser: Mozilla Firefox, Internet explorer, Chrome.

➤ Macromedia products: Dreamweaver Photoshop flash

➤ Database tool: PostgreSQL (PgAdmin 4)

## **PostgreSQL:**

➤ PostgreSQL is a database server.

➤ PostgreSQL is a multi-threaded, multi-user server.

➤ PostgreSQL supports standard SQL.

➤ PostgreSQL compiles on a number of platforms.

➤ PostgreSQL is free to download and use.

## SETTING MANAGEMENT IN DJANGO :

Defining the App applications in the INSTALLED\_APPS of settings.py file.

```
# Application definition

INSTALLED_APPS = [
    'orphanage', #for indexpage(home page)
    'food', #for surplus food page
    'Eventp', #for event celebration page
    'volunteer', #for volunteer page
    'donatingthings', #for donating things page
    'DONATE', #for donating page
    'sponsering', #for sponsoring page
    'gallery', #for gallery page
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
]
```

## STATIC FILES AND MEDIA FILES :

```
129  # Static files (CSS, JavaScript, Images)
130  # https://docs.djangoproject.com/en/3.1/howto/static-files/
131  STATIC_URL = '/static/' #static is for storing images,css,js files
132  STATICFILES_DIRS=[
133      os.path.join(BASE_DIR,'static')
134  ]
135
136  STATIC_ROOT = os.path.join(BASE_DIR,'staticfiles')
137
138
139  MEDIA_URL = '/media/' #files uploaded from admin are stored in this folder
140  MEDIA_ROOT = os.path.join(BASE_DIR,'media')
141
142
```

## DATABASES :

```
# Database
# https://docs.djangoproject.com/en/3.1/ref/settings/#databases

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql', #postgres database is used
        'NAME': 'Orphanage_DB', #name of the database of the website
        'USER': 'postgres', #user of the database of the website
        'PASSWORD': 'rokesh#123',
        'HOST': 'localhost',
    }
}
```

## TEMPLATE FOLDER : HOLDING HTML FILES.

```
ROOT_URLCONF = 'ORPHANAGE_MANAGEMENT_SYSTEM.urls'

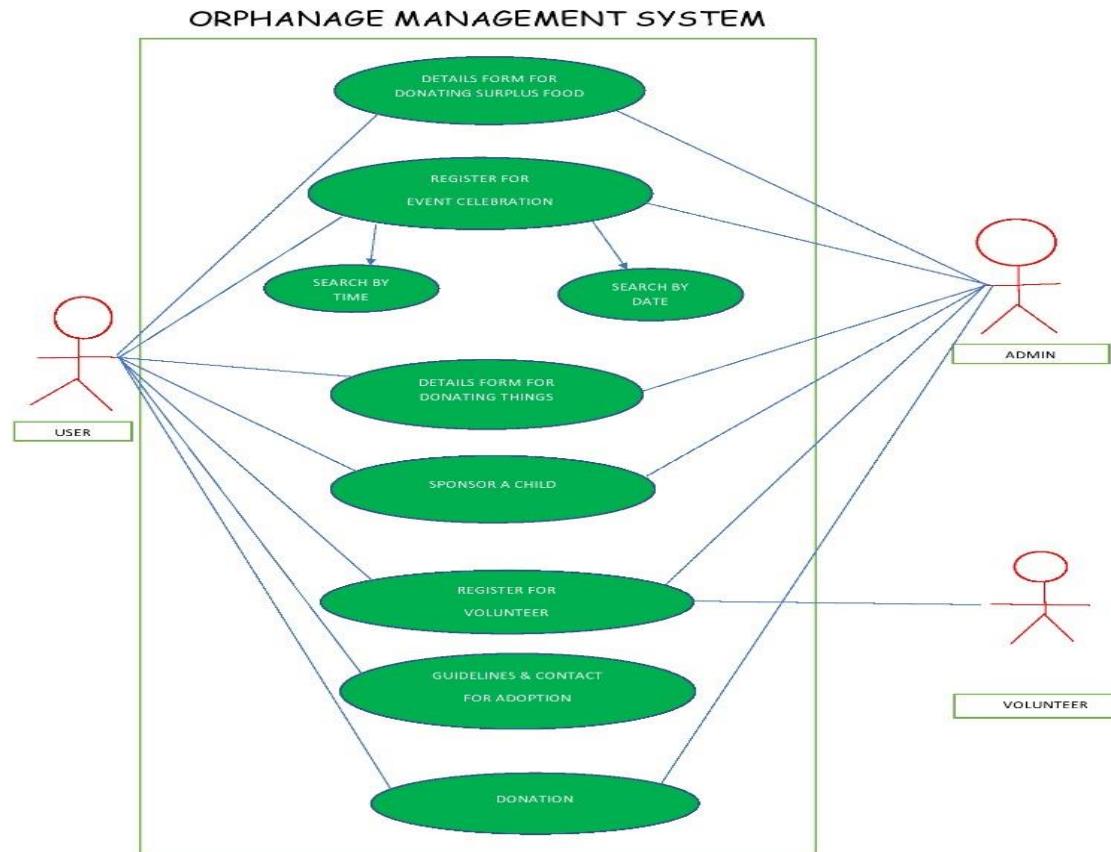
TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': ['template'], #template folder is for holding html files
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]
```

## URLS.PY FILE.

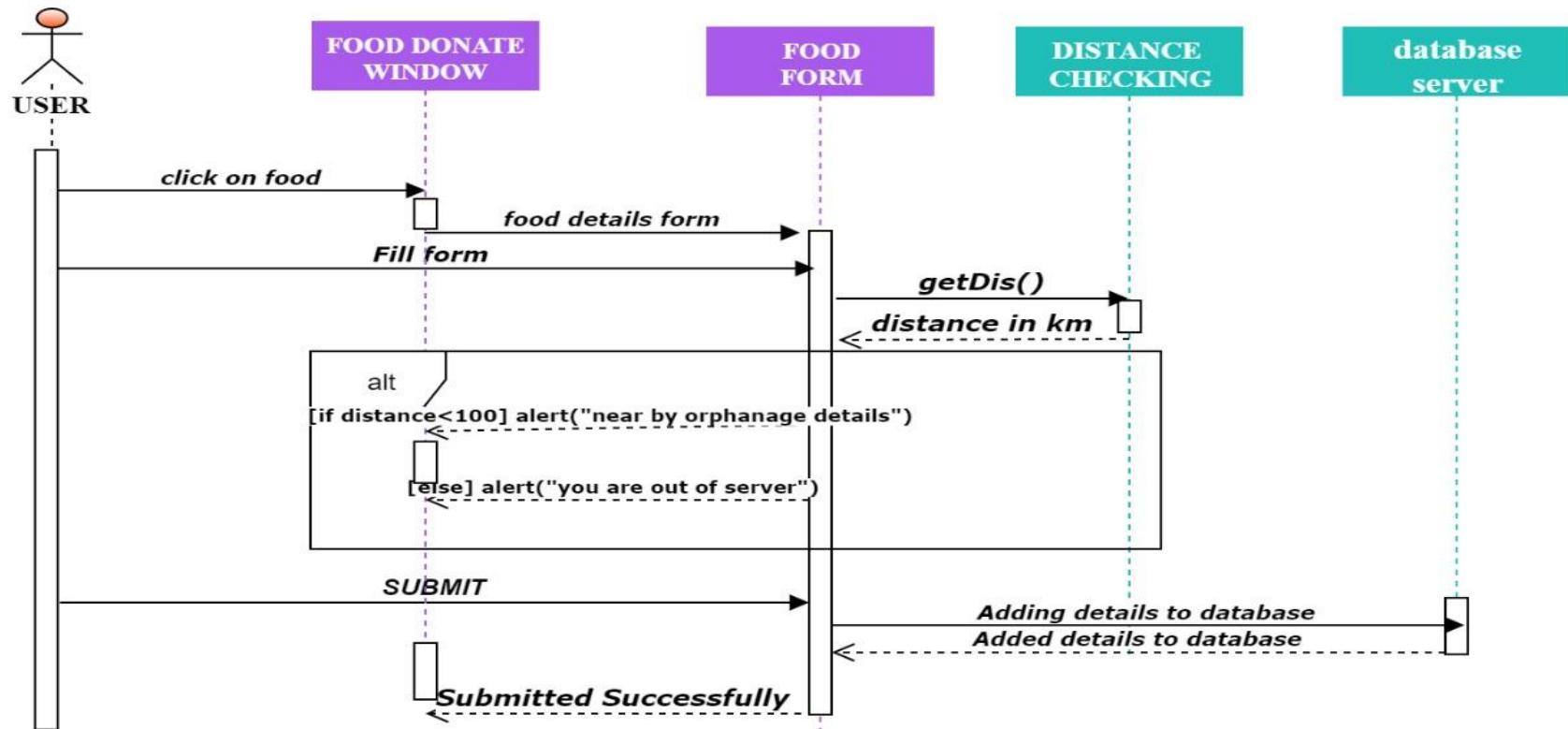
ORPHANAGE MANAGEMENT SYSTEM > ORPHANAGE MANAGEMENT SYSTEM > urls.py > ...

```
13     1. Import the include() function: from django.urls import include, path
14     2. Add a URL to urlpatterns:  path('blog/', include('blog.urls'))
15 """
16 from django.contrib import admin
17 from django.urls import path, include
18 from django.conf import settings
19 from django.conf.urls.static import static
20
21 urlpatterns = [
22     path('', include('orphanage.urls')),
23     path('food/', include('food.urls')),
24
25
26     path('Eventp/', include('Eventp.urls')),
27     path('volunteer/', include('volunteer.urls')),
28     path('donatingthings/', include('donatingthings.urls')),
29     path('DONATE/', include('DONATE.urls')),
30
31     path('sponsoring/', include('sponsoring.urls')),
32     path('gallery/', include('gallery.urls')),
33
34     path('admin/', admin.site.urls),
35
36 ]
37 urlpatterns = urlpatterns + static(settings.MEDIA_URL, document_root = settings.MEDIA_ROOT)
38 #This describes Django's file access APIs for files such as those uploaded by a user. If you want to handle "static files" (JS, CSS, etc.),
39 # see Managing static files (e.g. images, JavaScript, CSS). ... By default, Django stores files locally, using the MEDIA_ROOT and MEDIA_URL
40 # settings
41
```

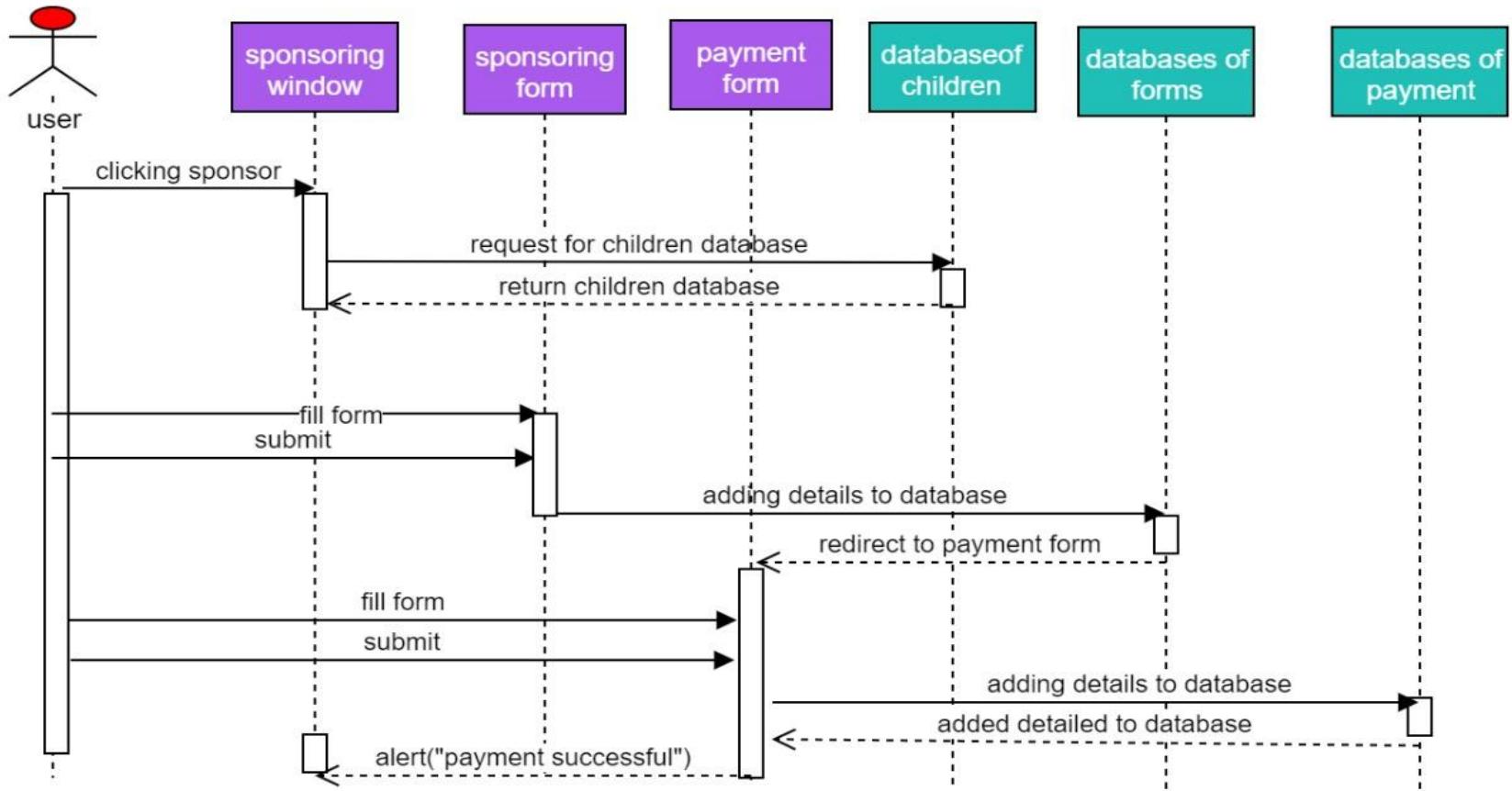
## USE CASE DIAGRAM:



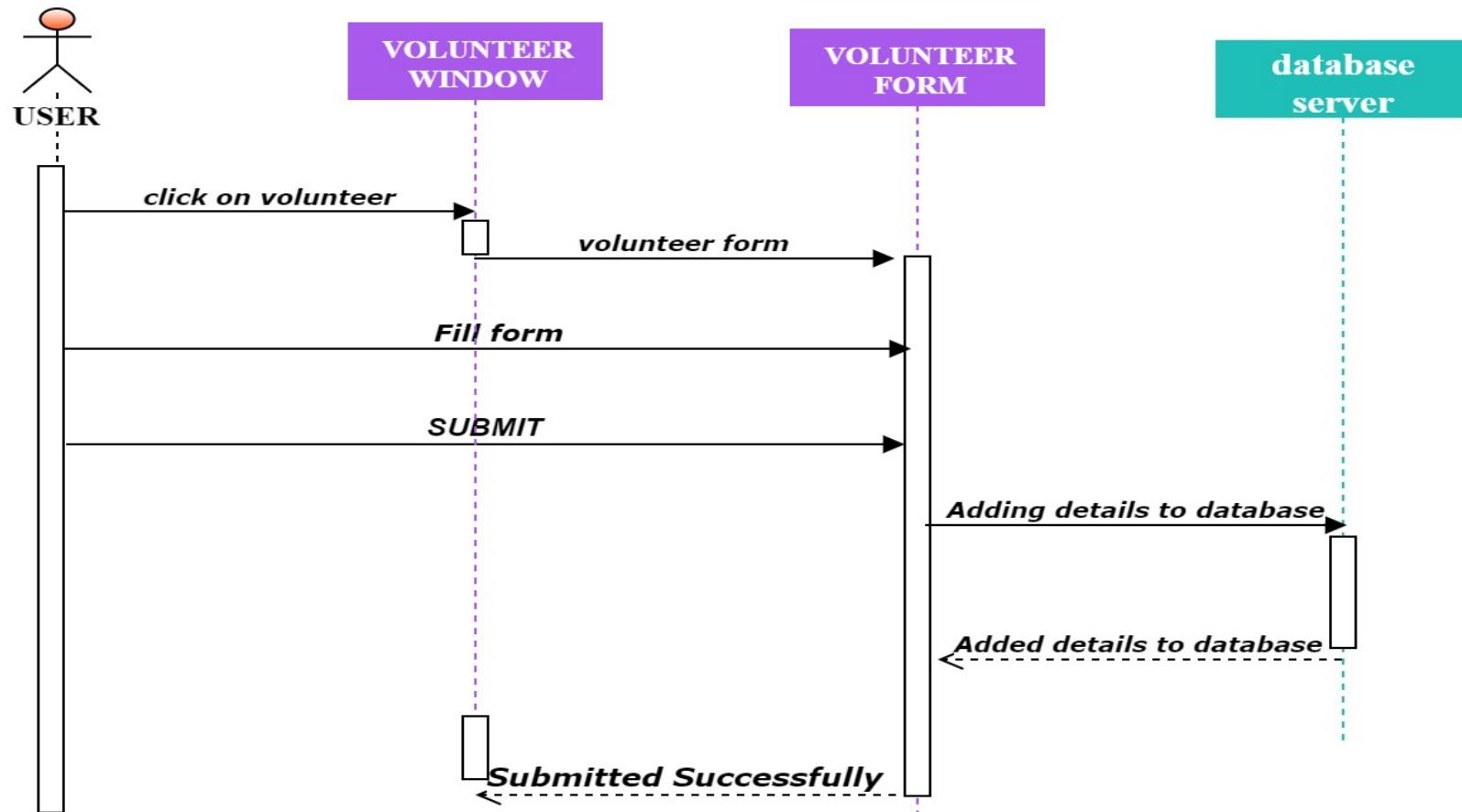
## SEQUENCE CASE DIAGRAMS: SURPLUS FOOD:



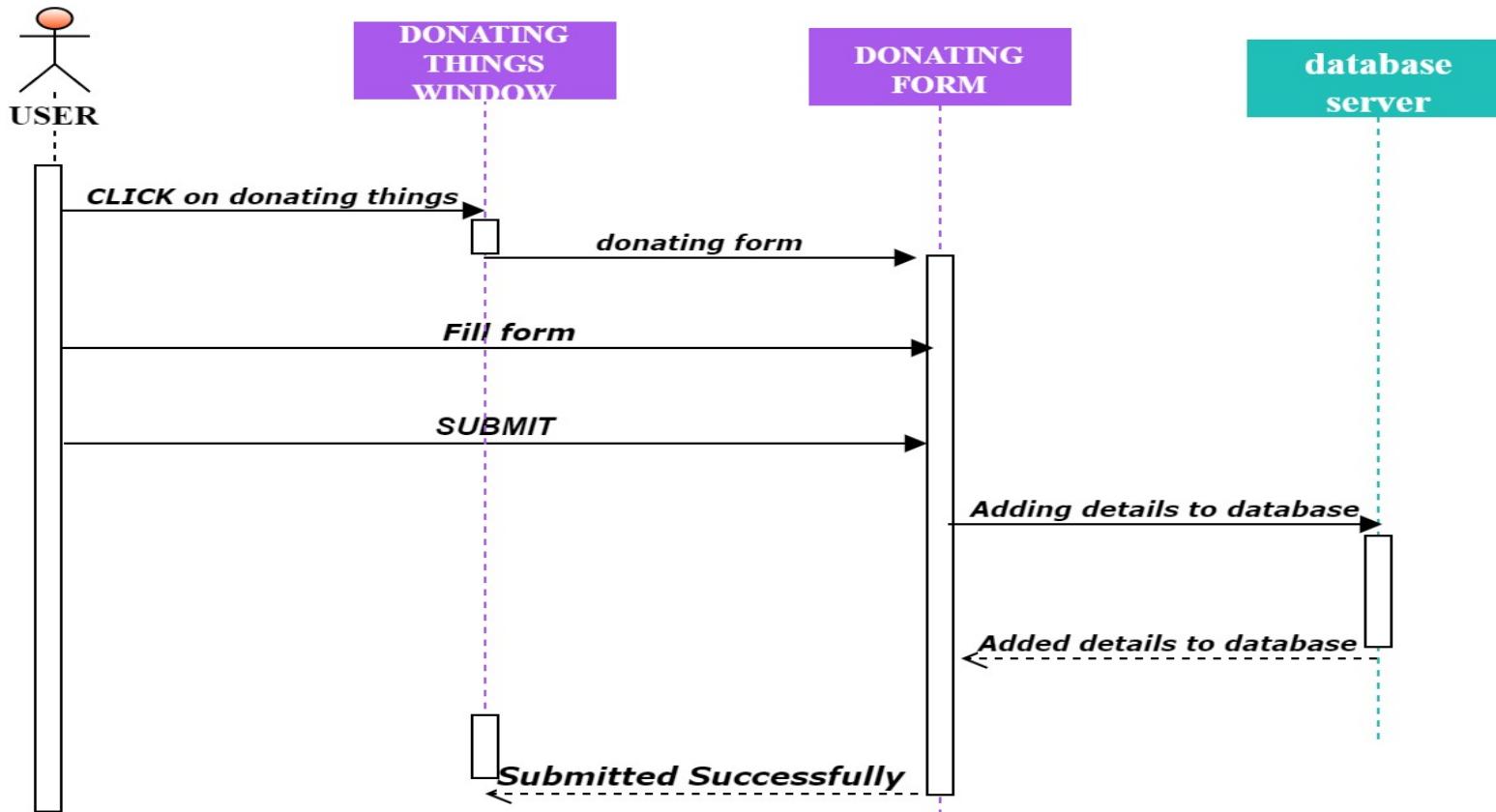
## SPONSOR A CHILD:



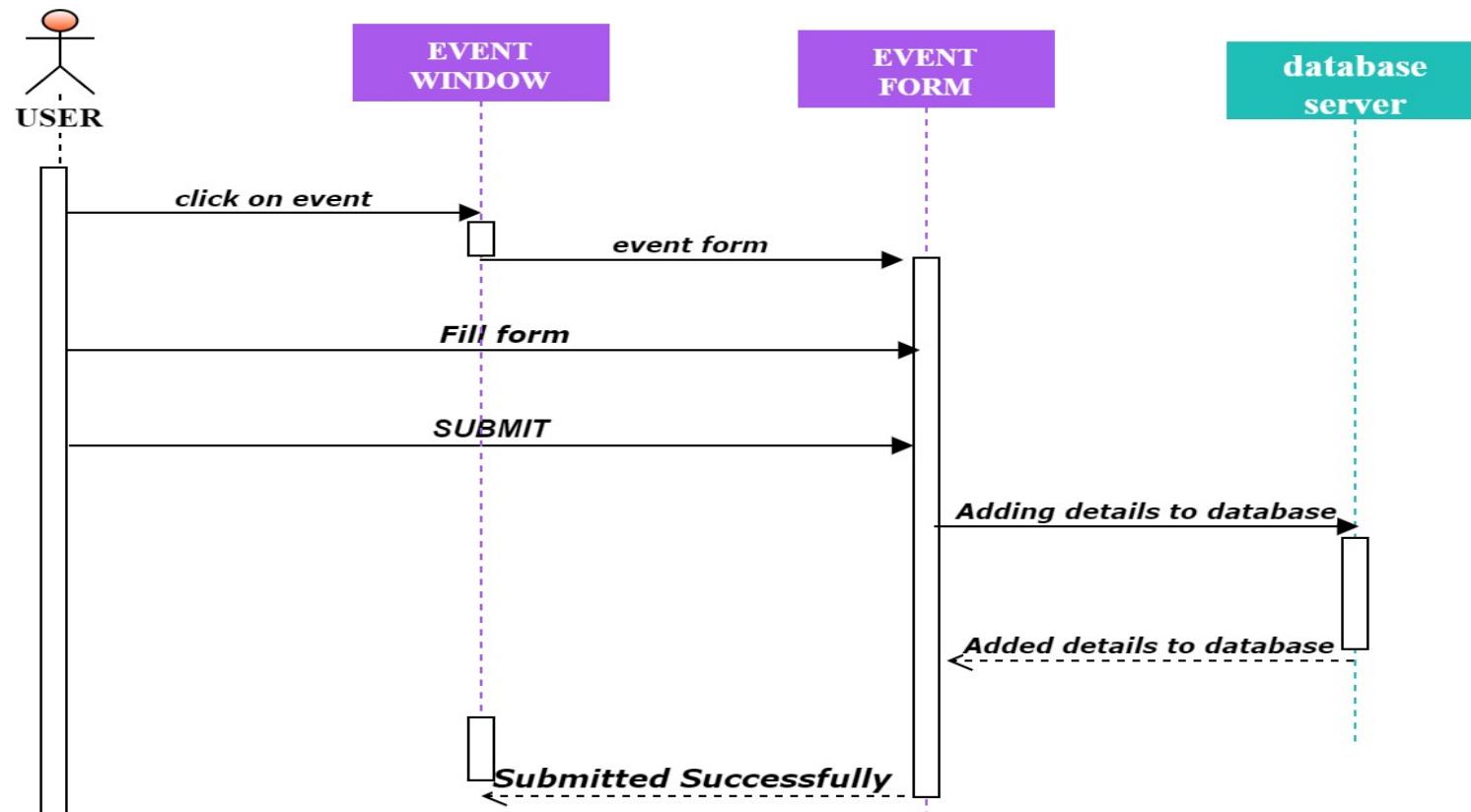
## VOLUNTEER:



## DONATING THINGS:



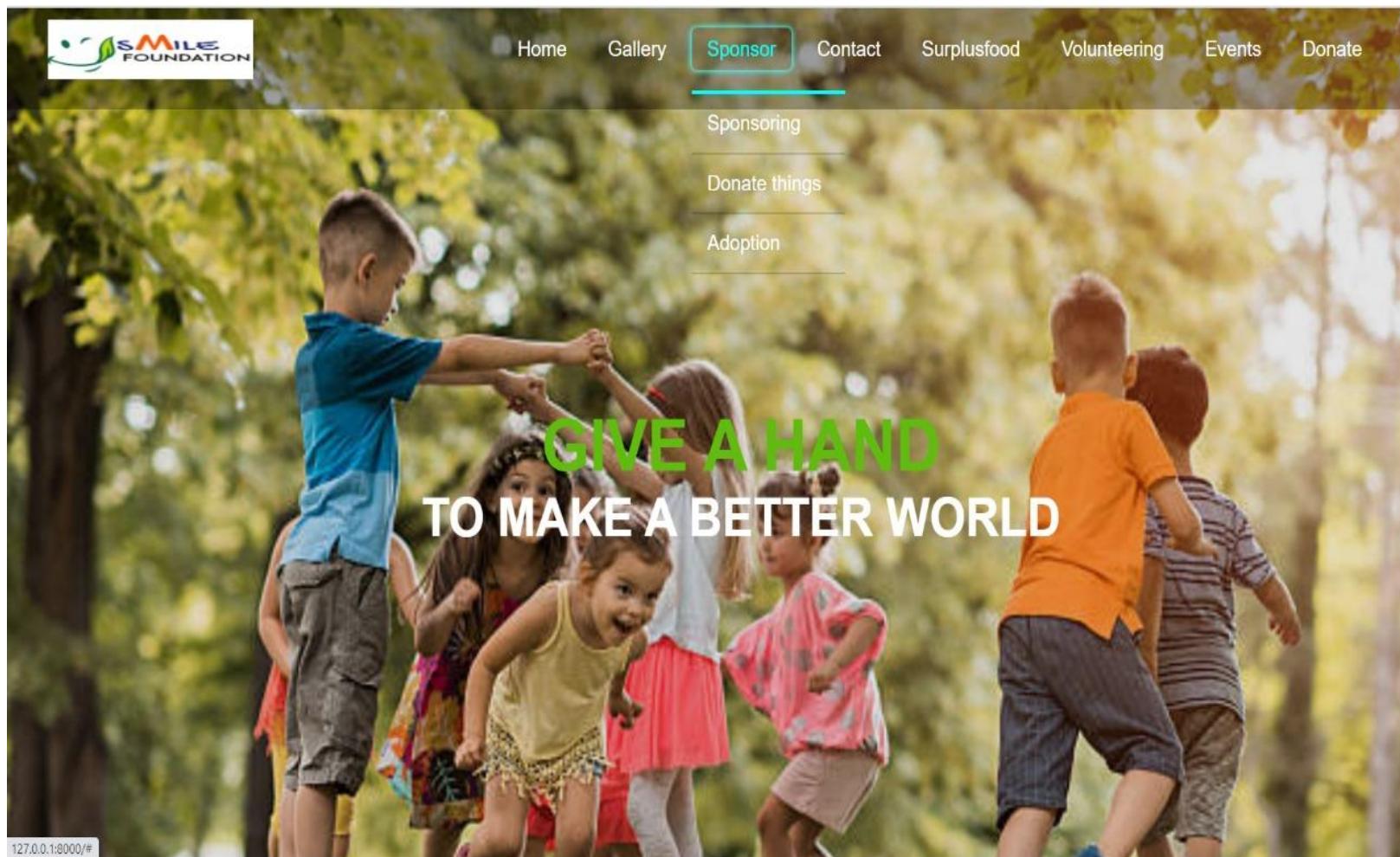
## EVENTS:



## OVERVIEW OF PAGES:

### HOME PAGE OF OUR WEBSITE:

The Home Page of the website. The Navigation bar in the Home page navigates the user to find the respective features. All features are mentioned in the navigation bar. The user can easily select the features they want.

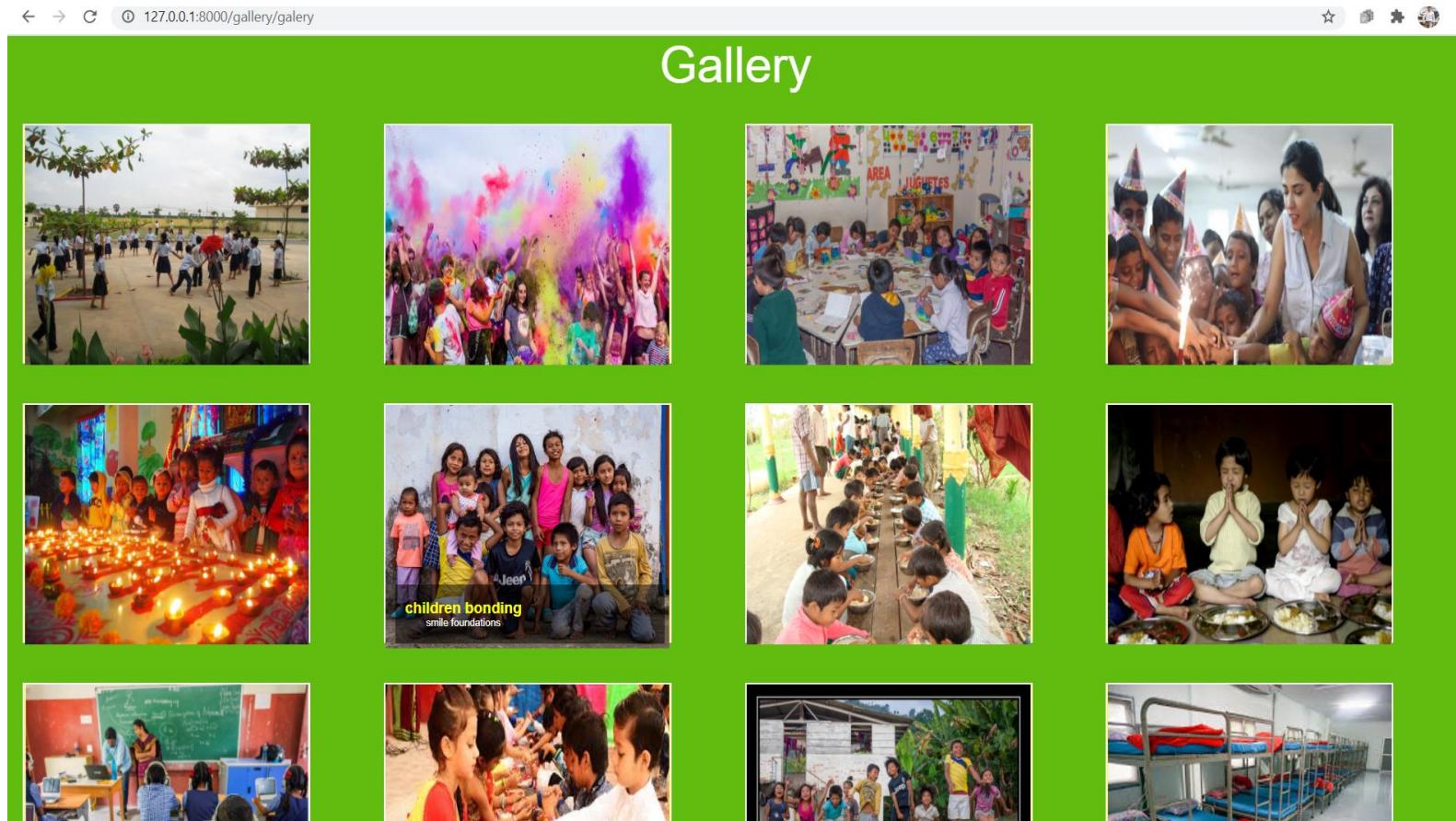


THIS INDEX FUNCTION IS TO RECEIVE WEB REQUEST AND OPENS HOME PAGE OF THE WEBSITE.

```
ORPHANAGE MANAGEMENT SYSTEM > orphanage > views.py > ...
1 #this is homepage app
2 from django.shortcuts import render
3
4 # Create your views here.
5 def index(request):
6     return render(request, "index.html")
7 #this fuction is to recieve web request and opens home | page of the website(web response)
8
9
```

## GALLERY OF THE WEBSITE:

On clicking the GALLERY button in the navigation bar , it takes the user to the gallery page. In the gallery the user can find the orphanage home related images.



## BACKEND PART OF GALLERY:

All the orphanage home related images are uploaded by the Admin through the admin panel. At any time The Admin can update and delete the images of the gallery( in the backend).

The screenshot shows the Django administration interface at the URL `127.0.0.1:8000/admin/gallery/gallery/add/`. The title bar says "Django administration" and "WELCOME, ORPHANAGE". The left sidebar has a "GALLERY" section with a yellow background, showing a "Galleries" item with a green plus icon. The main content area is titled "Add gallery". It has two fields: "Img:" with a "Choose File" button containing "crafty.jpg" and "Info:" with the value "Craft works of children". At the bottom right are three buttons: "Save and add another", "Save and continue editing", and a larger "SAVE" button.

This “gallery” function is to receive web request from user and redirect to the gallery of the website(web response)

```
ORPHANAGE_MANAGEMENT_SYSTEM > gallery > views.py > ...
1  from django.shortcuts import render
2  from .models import gallery
3  # Create your views here.
4  def galery(request):
5      photo=gallery.objects.all()
6      return render(request,'gallery.html',{'photo':photo})
7  #this view fuction is to recieve web request from user and redirect to the gallery of the website(web response)
8  # this is for gallery page
```

This "gallery" class in models.py represents tables or collection in our database for gallery page of our website

```
ORPHANAGE_MANAGEMENT_SYSTEM > gallery > models.py > ...
1  from django.db import models
2  from django.core.files.storage import FileSystemStorage
3  # Create your models here.
4  class gallery(models.Model):
5      img=models.ImageField(upload_to="pictures",default="")
6      info=models.CharField(max_length=50,default="")
7  #this "gallery" class in models.py represents tables or collection in our database for gallery page of our website
8  #ImageField is used for storing valid image files into the database. One can any type of image file in ImageField.
9
```

## SPONSOR A CHILD:

After Clicking the SPONSOR button in the navigation bar of the home page . It will take the user to the sponsor page.



**CHANGE**  
*Everything*  
**FOR AN ORPHAN**

[SPONSOR NOW](#)

The journey to end poverty  
for good begins with  
a child.

You can also contact us at this number  
+91 6309716107

Sponsoring a child is a personal way to show God's love to a child in need. But with support from people like you, we can help children get the health care, education, life skills, job-readiness training and confidence they need to create lasting change in their lives and communities.

[Sponsor A Child](#)

127.0.0.1:8000/sponsoring/sponsored

After clicking the Sponsor Now button it will takes the user to the sponsoring form which asks the user to fill some basic details.

← → ⌂ 127.0.0.1:8000/sponsoring/sponsored

Meet children waiting for a sponsor

Child sponsorship connects two people — a sponsor with a generous heart and a child. And Sponsorship continues to provide life-changing help to children.

**GIVE NOW**

First Name  
KONGAMUDI

Last Name  
rokesh

Gender  
Male

Email  
kmrokeshreddy@gmail.com

Phone Number  
09381540744

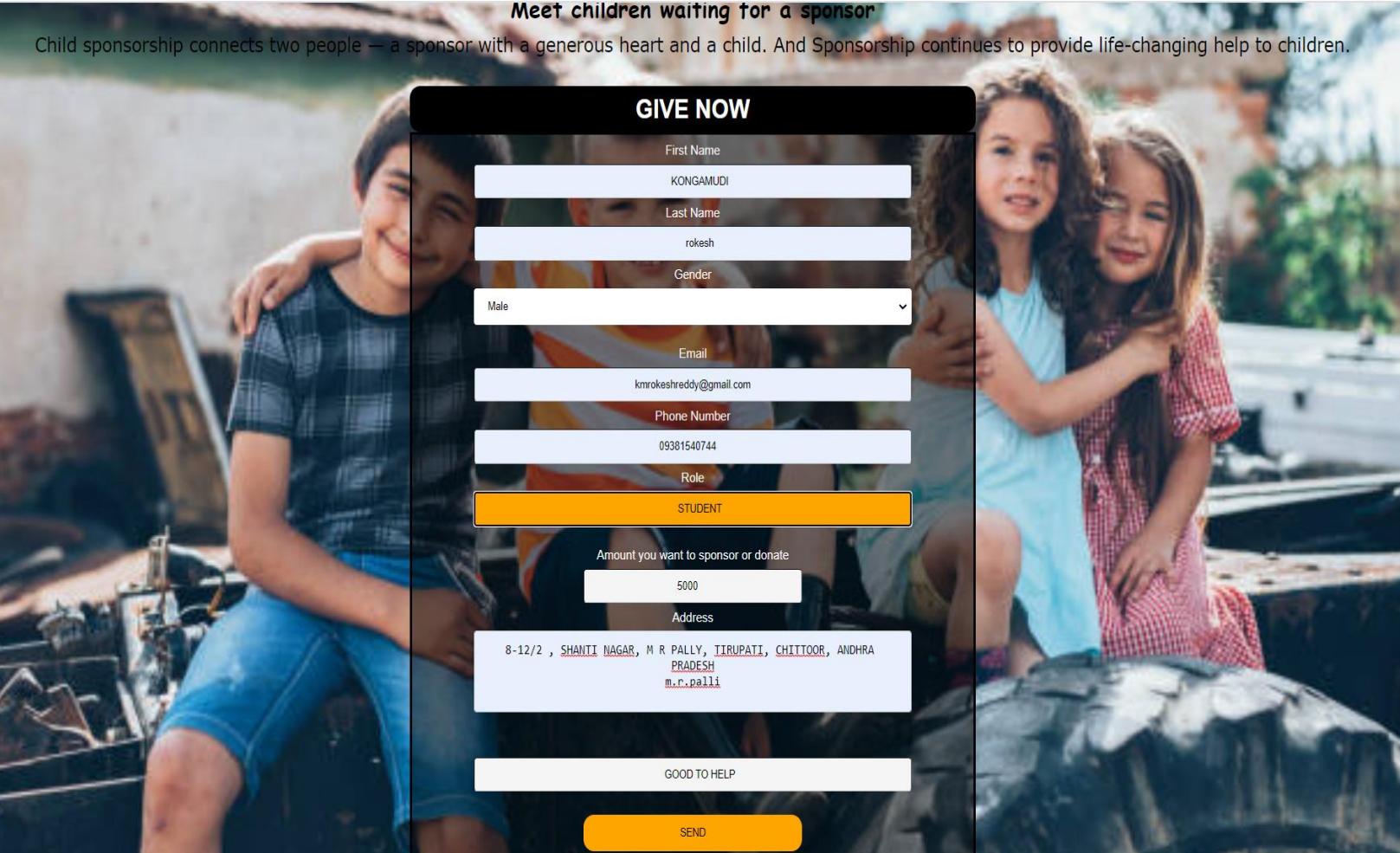
Role  
STUDENT

Amount you want to sponsor or donate  
5000

Address  
8-12/2 , SHANTI NAGAR, M R PALLY, TIRUPATI, CHITTOOR, ANDHRA PRADESH  
m.r.pally

GOOD TO HELP

SEND



# BACKEND PART OF THE SPONSOR FORM:

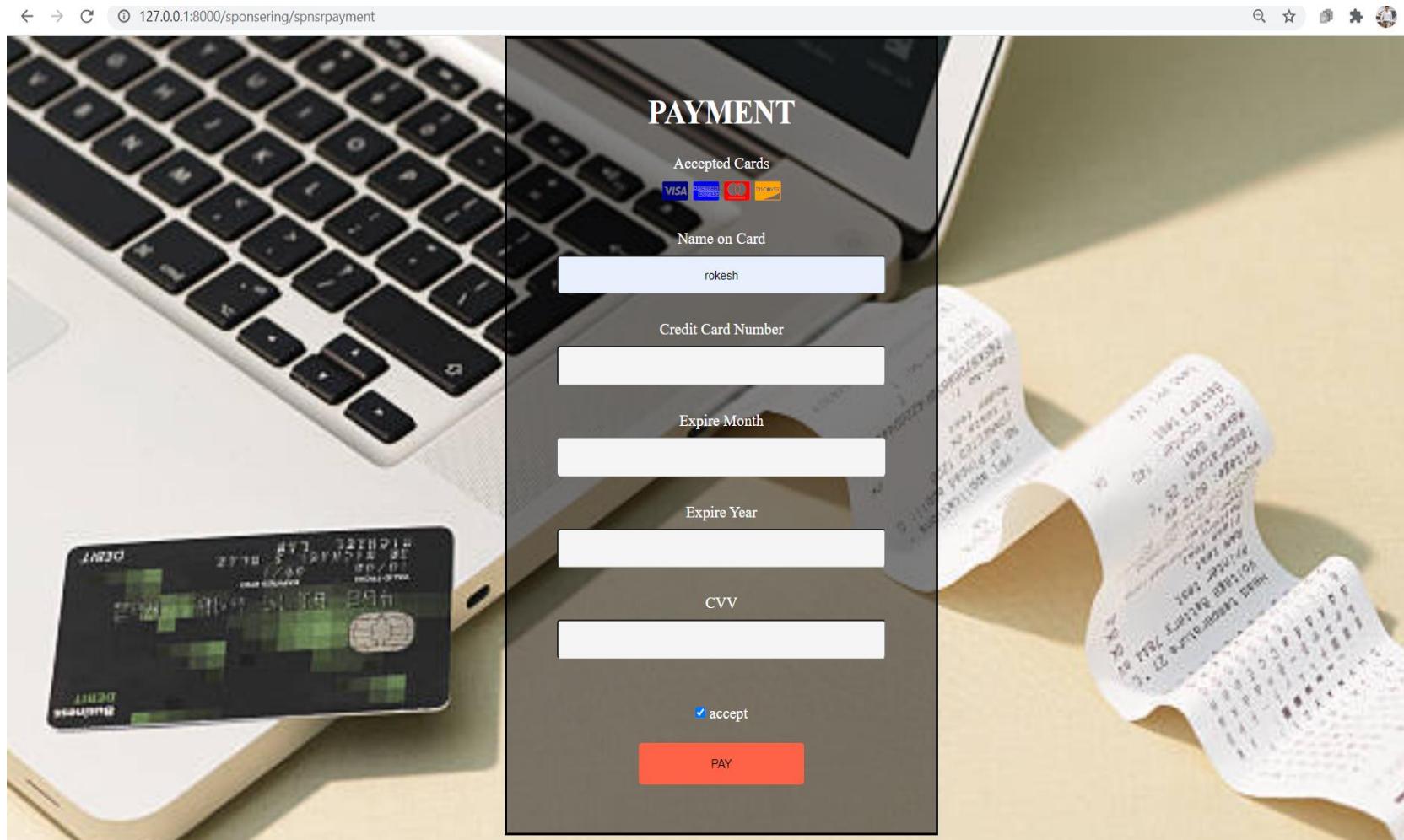
The entered data by the user is stored in the database of the website.

The screenshot shows the Django administration interface for a 'virat object (2)' model. The left sidebar contains a navigation menu with categories like AUTHENTICATION AND AUTHORIZATION, DONATE, DONATINGTHINGS, EVENTPP, FOOD, and GALLERY, each with 'Groups' and 'Users' or 'Financialreports' and 'Pays' options. The main content area is titled 'Change virat' and displays the following fields:

Fname:	sc
Lname:	s
Gender:	Male
Email:	c@gmail.com
Phone:	9381540744
Role:	lkujytfdf
Amount:	8520
Address:	KONGAMUDI rokesh, 8-12/2 , SHANTI NAGAR
Message:	iuytr

At the bottom right are three buttons: 'Delete' (red), 'Save and add another' (blue), 'Save and continue editing' (blue), and 'SAVE' (dark blue).

After filling the user details it will takes the user to the payment form for transactions.



## BACKEND OF THE PAYMENT FORM:

The Name of the cardholder and the card number given by the user is stored in the website database.

The screenshot shows the Django administration interface for a payment object. The URL in the browser is `127.0.0.1:8000/admin/sponsoring/payment/2/change/`. The page title is "Change payment". The left sidebar contains a navigation menu with sections like AUTHENTICATION AND AUTHORIZATION, DONATE, DONATINGTHINGS, EVENTP, FOOD, GALLERY, and SPONSERING. Under SPONSERING, there is a link to "Imgss". The main form has two fields: "Name:" with the value "rokesh" and "Card:" with the value "9878563". At the bottom right are three buttons: "Delete", "Save and add another", "Save and continue editing", and a large "SAVE" button.

## CHILDREN WAITING FOR THE SPONSORS:

The Orphanage children who are waiting for the sponsorship from the Sponsors, the images and the details are uploaded in this page. After clicking the SPONSOR NOW button it will take the user to the sponsor form where sponsor can fill the details and start sponsoring the child.

← → ⌂ 127.0.0.1:8000/sponsoring/img

# CHILDREN WAITING FOR THE SPONSORS

The screenshot shows a web page titled "CHILDREN WAITING FOR THE SPONSORS". The URL in the address bar is "127.0.0.1:8000/sponsoring/img". The page displays six children's portraits arranged in two rows of three. Each portrait is accompanied by a green rounded rectangular button labeled "SPONSOR NOW". The first row contains portraits of a young man, a young boy, and a young girl. The second row contains portraits of two boys. The browser interface includes standard navigation buttons (back, forward, search) and a toolbar with icons for star, refresh, and other functions.

[SPONSOR NOW](#)

[SPONSOR NOW](#)

[SPONSOR NOW](#)

[SPONSOR NOW](#)

[SPONSOR NOW](#)

[SPONSOR NOW](#)

## BACKEND PART:

The images and the details of the children in the orphanage who are waiting for the sponsors are uploaded to the website from the backend.

The screenshot shows the Django administration interface for the 'Sponsoring' model, specifically the 'Imgss' tab. The URL in the browser is `127.0.0.1:8000/admin/sponsoring/imgss/add/`. The page title is 'Django administration'.

The left sidebar contains a navigation menu with several categories and their sub-items:

- AUTHENTICATION AND AUTHORIZATION
  - Groups
  - Users
- DONATE
  - Financialreports
  - Pays
- DONATINGTHINGS
  - Contactforms
  - Regformdthingss
- EVENTP
  - Calenders
  - Eventforms
  - Eventphotos
- FOOD
  - Foodregdetailss
- GALLERY
  - Galleries

The main content area is titled 'Add imgs'. It contains the following fields:

- Img: A file input field with the value 'orphan2.jpg'.
- Name: An input field containing 'chitti'.
- Age: An input field containing '5'.

At the bottom right of the form are three buttons: 'Save and add another', 'Save and continue editing', and a large blue 'SAVE' button.

This view “sponsor” is to receive web request from user and redirect to the sponsor main page of the website(web response) .This “sponsored” is to receive web request from user and redirect to the sponsoring details page of the website(web response)

```
ORPHANAGE MANAGEMENT SYSTEM > sponsoring > 📄 views.py > 🗁 spnsrpment
 1  from django.shortcuts import render
 2  from .models import virat
 3  # this virat class is for sponsor details form
 4  from .models import payment
 5  from .models import imgs
 6  from django.shortcuts import redirect
 7  from django.http import HttpResponseRedirect
 8
 9  # Create your views here.
10 def sponsor(request):
11     return render(request,"sponsoredpage.html")
12 #this view function is to receive web request from user and redirect to the sponsor main page of the website(web response)
13 # this is for sponsor main page
14 def sponsored(request):
15     if request.method=="POST":
16
17         fn=request.POST["fname"]
18         ln=request.POST["lname"]
19         g=request.POST["gender"]
20         em=request.POST["email"]
21         ph=request.POST["phone"]
22         re=request.POST["Role"]
23         amt=request.POST["amount"]
24         add=request.POST["address"]
25         mes=request.POST["message"]
26         data=virat(fname=fn, lname=ln, gender=g, email=em, phone=ph, Role=re, amount=amt, address=add, message=mes)
27         data.save()
28         return redirect("spnsrpment")
29         return HttpResponseRedirect("<h1> Successful:</h1>")
30         return render(request,"payment.html")
31
32     return render(request,"sponsored.html")
33 #this "sponsored" function is for form in sponsor page which seeks the basics details of the sponsor
```

This view “sponsoring” function is to receive web request from user and redirect to the sponsoring details page of the website(web response)

```
> def sponsoring(request):
>
>     return render(request,'sponsorship.html')
> #"sponsoring" fuction is to recieve web request from user and redirect to the sponsoring details page of the website(web response)
> def img(request):
>
>     pics=imgs.objects.all()
>     return render(request,'images.html',{'pics':pics})
> #this "imgs" fuction is to recieve web request from user and redirect to the children waiting for sponsor of the website(web response)
> def adoption(request):
>     return render(request,"adopt.html")
> #this "adoption" fuction is to recieve web request from user and redirect to the adoption page of the website(web response)
> def spnsrpayment(request):
>     if request.method=="POST":
>
>         cardname=request.POST["name"]
>         cardnumber=request.POST["card"]
>         data=payment(name=cardname,card=cardnumber)
>         data.save()
>         return HttpResponse("<h1> Payment Successful:</h1>")
> #this "spnsrpayment" function is for payment page which is for payment
>
>         return redirect("sponsor")
>
>     return render(request,"payment.html")
```

Function “img” for the page children for the sponsor, "adoption" function is to receive web request from user and redirect to the adoption page of the website(web response)  
"spnsrpayment" function is for payment page which is for payment

ORPHANAGE MANAGEMENT SYSTEM > sponsoring >  models.py > ...

```
1 from django.db import models
2
3 # Create your models here.
4 # this virat class is for sponsor details form
5 class virat(models.Model):
6     fname=models.CharField(max_length=50)
7     lname=models.CharField(max_length=50)
8     GENDER_CHOICES = (('male', 'Male'), ('female', 'Female'), ('other', 'Other'))
9     gender = models.CharField(max_length=7, choices=GENDER_CHOICES,null=True)
10    email=models.EmailField(max_length=50)
11    phone=models.BigIntegerField()
12    Role=models.CharField(max_length=200)
13    amount=models.IntegerField(default=0)
14    address=models.CharField(max_length=500)
15    message=models.CharField(max_length=200)
16 #this "virat" class in models.py represents tables or collection in our database for the form in sponsor page of the website
17
18 class imgs(models.Model):
19
20
21     img=models.ImageField(upload_to="pictures",default="")
22     name=models.CharField(max_length=50,default="")
23     age=models.CharField(max_length=50,default="")
24 #this "imgs" class in models.py represents tables or collection in our database for children waiting for sponsor of the website
25
26 class payment(models.Model):
27
28     name=models.CharField(max_length=200)
29     card=models.BigIntegerField()
30 #this "payment" class in models.py represents tables or collection in our database for the payment page in sponsor page of the website
```

"payment", "virat" and "imgs" classes in models.py represents tables or collection in our database for the form in sponsor page,children waiting for sponsor and payment page in sponsor page of the website

## DONATING THINGS:

By clicking the DONATING THINGS button in the navigation bar the website will take the user to the donating things page. The detailed information of which things can be donated to the orphans is displayed on the website. The users who want to donate things through the courier can donate by finding the postal address on the website.

The screenshot shows a web browser window with the URL 127.0.0.1:8000/donatingthings/dthings. The main title is "Donate Things To Change Lives Of Orphans" with a subtitle "A Really Great Way To Help Us Out Is To Donate Goods To Our Orphanage." A green "Donate now" button is visible. Below the main title, there are three sections: "Donate Toys" (with an icon of a teddy bear in a box), "Donate Clothes" (with an icon of a hand holding a shirt), and "Donate Books" (with an icon of a book and a lightbulb). Each section has a brief description. At the bottom, a green bar contains the text "Postal Address For Packages".

**Donate Things To Change Lives Of Orphans**  
A Really Great Way To Help Us Out Is To Donate Goods To Our Orphanage.

**Donate now**

**Donate Toys**



Everyone have their most funtimes is playing with toys. if your child doesn't play with some toys anymore,donate them.Toy donations brings happiness and smiles on orphanage kids.

**Donate Clothes**



you're looking for a way to help the needy, donating clothes could be a super-easy start. A small gesture of yours will brings lots of happiness in orphanage homes. Donate them to orphanage kids.

**Donate Books**



Books have been your friends. They helped you find solutions to you. Reading books can improve creativity and imagination power. Share with the less fortunate orphans, books are not affordable.

**Postal Address For Packages**

By clicking the DONATE NOW on the donating things page it will take the user to the donating things form. By giving the basic details of the donating things to the form the user can successfully donate the things to the orphanage.

### Register Form



user details    donation details    location details

1 ————— 2 ————— 3

#### User Details

First Name  
rokes

Last Name  
reddy

Email  
rokes@gmail.com

Gender  
male

age  
20

phone no  
9381540744++

**NEXT**

### Register Form



user details    donation details    location details

1 ✓ ————— 2 ————— 3

#### Donation information

toys  
54

books  
5

blankets  
9

clothes  
boy

dress type  
shirt

others  
cycle

**PREVIOUS**    **NEXT**

### Register Form



user details    donation details    location details

1 ✓ ————— 2 ✓ ————— 3

#### Location

address  
shanti nagar

landmark  
m.t.palli

city  
TIRUPATI

District  
chittoor

Pincode  
517502

State  
Andhra Pradesh

**PREVIOUS**    **SUBMIT**

## BACKEND PART OF THE DONATING THINGS FORM:

All the details given in the form by the user will be stored in the website database.

The screenshot shows the Django Admin interface for a 'regformdthings' object. The left sidebar contains a navigation menu with categories like AUTHENTICATION AND AUTHORIZATION, DONATE, DONATINGTHINGS, EVENTP, FOOD, GALLERY, and SPONSERING. Under the 'DONATINGTHINGS' category, 'Regformdthingss' is selected. The main content area displays a form with the following fields and values:

Field	Value
Fname:	dfghj
Lname:	4rtyj
Email:	c@gmail.com
Gender:	male
Age:	20
Pnumber:	9381540744
Ntoys:	54
Nbooks:	5
Nblankets:	0
Dress:	werg
Cgender:	boy
Others:	cycle
Adress:	fgbn
Landmark:	ertg

"dthings" function is to receive web request from user and redirect to the donating things page of the website(web response). "dtreform" function is for form in donating things form page which seeks the basics details for donating things.

ORPHANAGE MANAGEMENT SYSTEM > donatingthings > views.py > dtreform

```
21
22 def dthings(request):
23     return render(request, "donatingthings.html")
24 #this fuction is to recieve web request from user and redirect to the donating things page of the website(web response)
25 def dtreform(request):
26     if request.method == 'POST':
27         f1 = request.POST['fname']
28         l1 = request.POST['lname']
29         e1 = request.POST['email']
30         ge = request.POST['gender']
31         a1 = request.POST['age']
32         p1 = request.POST['pnumber']
33         nt = request.POST['ntoys']
34         b1 = request.POST['nbooks']
35         nb = request.POST['nblankets']
36         dr = request.POST['dress']
37         cg = request.POST['cgender']
38         ot = request.POST['others']
39         ad = request.POST['adress']
40         la = request.POST['landmark']
41         ci = request.POST['city']
42         di = request.POST['district']
43         data = regformdthings(fname=f1, lname=l1, email=e1, gender=ge, age=a1, pnumber=p1, ntoys=nt, nbooks=b1, nblankets=nb, dress=dr, cgender=cg)
44         data.save()
45         return HttpResponse("<h1>submitted successfully</h1>")
46         return render(request, "dtregisterform.html")
47 #this "dtreform" function is for form in donating things form page which seeks the basics details for donating things
48
```

"regformdthing" class in models.py represents tables or collection in our database for the form in donating things page of the website

```
ORPHANAGE MANAGEMENT SYSTEM > donatingthings > ✎ models.py > 📁 contactform
 1  from django.db import models
 2
 3  # Create your models here.
 4  class regformdthings(models.Model):
 5      fname = models.CharField(max_length=50)
 6      lname = models.CharField(max_length=50)
 7      email = models.EmailField(max_length=254)
 8      gender_choices = (('m', 'male'), ('f', 'female'), ('u', 'other'))
 9      gender = models.CharField(max_length=1, choices=gender_choices,null=True)
10      age = models.IntegerField()
11      pnumber = models.BigIntegerField()
12      ntoys = models.CharField(max_length=50)
13      nbooks = models.CharField(max_length=50)
14      nblankets = models.CharField(max_length=50)
15      dress = models.CharField(max_length=50)
16      clothes_choices = (('M', 'boy'),('F','girl'))
17      cgender = models.CharField(max_length=1, choices=clothes_choices,null=True)
18      others = models.CharField(max_length=50)
19      adress = models.CharField(max_length=5000)
20      landmark = models.CharField(max_length=50)
21      city = models.CharField(max_length=50)
22      district = models.CharField(max_length=50)
23  #this "regformdthing" class in models.py represents tables or collection in our database for the form in donating things page of the website
24  class contactform(models.Model):
25      name = models.CharField(max_length=50)
26      phone = models.BigIntegerField()
27      email = models.EmailField(max_length=254)
28      subject = models.CharField(max_length=50)
29      message = models.CharField(max_length=100)
30  #this "contactform" class in models.py represents tables or collection in our database for the form in contact page of the website
```

## ADOPTION:

By clicking the ADOPTION button in the navigation bar it take the user to the adoption page . In this page the user can find the guidelines for the adoption of an orphanage child. Website provides the contact details, those who interested can contact.



ADOPTION

DON'T SHOP..ADOPT..

We have an overwhelming amount of people asking us about adoption. As a result we have made this page. Please note:

We are by no means experts in the matter but in our experience and understanding in-country and inter-country adoption in India has been very difficult in the past but new laws are being implemented to streamline this process. Estimates for the number of orphans in India varies from 16 million to 30 million, but only about 2,500 orphans were adopted last year, down from 5,700 four years ago. The Ministry of Women & Child Development's autonomous body Central Adoption Resource Authority (CARA) now handles all adoption inquiries, applications and adoptions within India. Prospective parents are should register with the authority, whose automated system will match them with children

For those people wishing to adopt we suggest the following:

- 1.Only adopt from registered Specialized Adoption Agencies (SAA's) recognized by State Governments.
- 2.Read the guidelines on CARA's website carefully.
- 3.Do not try to adopt directly from nursing homes, hospitals, maternity homes, unauthorized institutions or individuals.

While we believe that adoption can be a beautiful thing it is not something that our organization is or at this stage plans to be involved with in the future. We wish anyone wanting to adopt the very best and we hope that you find what you are looking for.

Note: The current Indian system allows adoption to single/unmarried people but does not allow adoption by gay/lesbian prospective parents

## CONTACT PAGE:

By clicking the CONTACT button on the navigation bar it take the user to the contact page. In this page the user can ask any queries related to the orphanage details and any further queries.

The screenshot shows a contact form on a web page. At the top left, there is a search bar with the URL "127.0.0.1:8000/donatingthings/contact". To the right of the search bar are several small icons: a magnifying glass, a star, a folder, a puzzle piece, a person icon, and a three-dot menu. The main content area has a light gray background. On the left side, there is a white rectangular form with the title "contact us" in bold. Below the title, a message says "we are here to help you and respond to your querries as early as possible. you can also get our friendly customer care service". The form contains four input fields: "name" (KONGAMUDI), "phone" (09381540744), "email" (kmrokeshreddy@gmail.com), and "subject" (SHARED WEB HOSTING). Below these fields is a larger text area labeled "message" with the placeholder "GOOD TO SEND DETAILED MESSAGE |". At the bottom of the form is a blue "send" button. To the right of the form is a dark blue sidebar with the title "reach us" in bold. It lists the following contact information: email (rokiando@gmail.com), phone (9381540744), and address (8-12,santhi nagar,m.t palli, tirupati, chittoor, andhrapradesh).

contact us

we are here to help you and respond to your querries as early as possible.  
you can also get our friendly customer care service

name  
KONGAMUDI

phone  
09381540744

email  
kmrokeshreddy@gmail.com

subject  
SHARED WEB HOSTING

message  
GOOD TO SEND DETAILED MESSAGE |

send

reach us

email rokiando@gmail.com

phone 9381540744

adress 8-12,santhi nagar,m.t palli  
tirupati,chittoor, andhrapradesh

## BACKEND PART OF THE CONTACT PAGE:

The queries asked by the user in the contactus page are in the database of the website.

The screenshot shows the Django administration interface for a 'contactform object (2)'. The left sidebar contains a navigation menu with categories like AUTHENTICATION AND AUTHORIZATION, DONATE, DONATINGTHINGS, EVENTP, FOOD, GALLERY, SPONSERING, and VOLUNTEER. Under the DONATINGTHINGS category, 'Contactforms' is selected, and its detail view is shown. The contact form fields include Name (KONGAMUDI), Phone (9381540744), Email (kmrokereddy@gmail.com), Subject (SHARED WEB HOSTING), and Message (GOOD TO SEND DETAILED MESSAGE). At the bottom right are buttons for Delete, Save and add another, Save and continue editing, and SAVE.

Django administration

Home · Donatingthings · Contactforms · contactform object (2)

WELCOME, ORPHANAGE · VIEW SITE / CHANGE PASSWORD / LOG OUT

AUTHENTICATION AND AUTHORIZATION

Groups + Add

Users + Add

DONATE

Financialreports + Add

Pays + Add

DONATINGTHINGS

Contactforms + Add

Regformdthingss + Add

EVENTP

Calenders + Add

Eventforms + Add

« Eventphotos + Add

FOOD

Foodregdetailss + Add

GALLERY

Galleries + Add

SPONSERING

Imgss + Add

Payments + Add

Virats + Add

VOLUNTEER

Change contactform

contactform object (2)

Name: KONGAMUDI

Phone: 9381540744

Email: kmrokereddy@gmail.com

Subject: SHARED WEB HOSTING

Message: GOOD TO SEND DETAILED MESSAGE

Delete

Save and add another

Save and continue editing

SAVE

"contactform" class in models.py represents tables or collection in our database for the form in contact page of the website

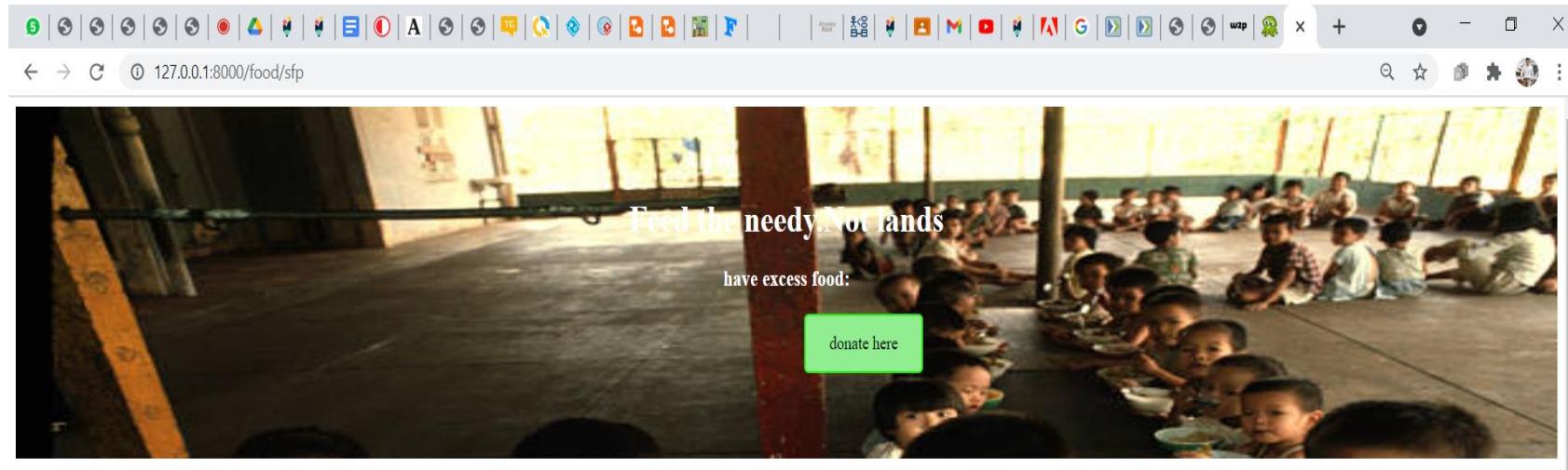
```
class contactform(models.Model):
    name = models.CharField(max_length=50)
    phone = models.BigIntegerField()
    email = models.EmailField(max_length=254)
    subject = models.CharField(max_length=50)
    message = models.CharField(max_length=100)
#this "contactform" class in models.py represents tables or collection in our database for the form in contact page of the website
```

"contact" function is for contact form page which seeks the queries of users about the website function is to receive web request from user and redirect to the donating things page of the website(web response)

```
# Create your views here.
def contact(request):
    if request.method=='POST':
        name=request.POST['name']
        ph=request.POST['phone']
        emailid=request.POST['email']
        sub=request.POST['subject']
        mes=request.POST['message']
        dataac = contactform(name=name,phone=ph,email=emailid,subject=sub,message=mes)
        dataac.save()
        return HttpResponse("<h1>submitted succesfully</h1>")
#this "contact" function is for contact form page which seeks the queries of users about the website
```

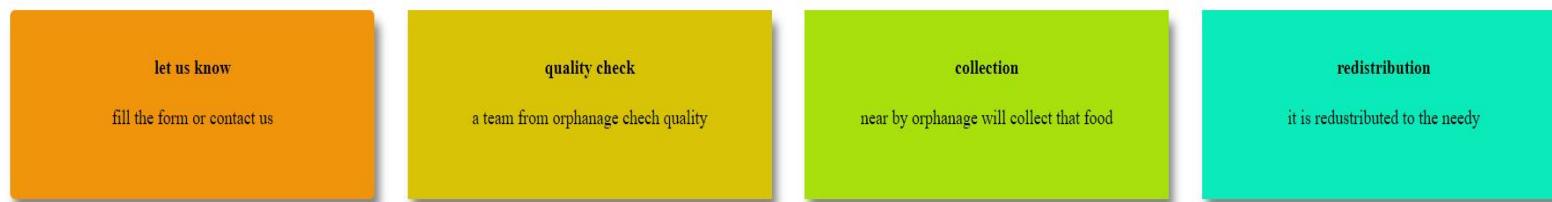
## SURPLUS FOOD:

By clicking on the SURPLUS FOOD button on the navigation bar it take the user to the surplus food page.



### How We Work

We aim to redistribute surplus food to those who are in need of it. If you have excess food from an event, party, wedding etc, please call our helpline, our volunteers will collect the excess food from you and distribute it among people who need it. The excess food will be checked for quality because lack of proper refrigeration and storage tend to spoil food.



### Food Wastage in India

We aim to redistribute surplus food to those who are in need of it. If any user has excess food from an event, party, wedding etc.. They can donate to the orphanages nearer to them by clicking the DONATE HERE button. After clicking the donate here button it will take the user to the page where the user is asked to fill the basic details of the form. In this the user's location (latitudes and longitudes) are traced and distance is calculated to the nearby orphanage with which we are collaborated. The food is collected by the nearby orphanage team if and only if the food distance is less than 100km.

The nearby orphanage details is shown like an alert message after clicking the button PRESS HERE TO GET THE NEARBY ORPHANAGE. If the distance is greater than 100km then website will show an alert message that "you are out the server". If anyone skips the some details it show an alert msg that the data must be filled.



← → C ① 127.0.0.1:8000/food/registerform

## GIVE FOOD DETAILS HERE

First Name  
BHARATH

Last Name  
REDDY

Email  
pdblreddy@gmail.com

Phone Number  
09381540744

Type Of Food  
Non-Veg Food

State Of Food  
Cooked

Food If Cooked  
30-05-2021 11:46

Food Quantity  
150



← → C ① 127.0.0.1:8000/food/registerform

127.0.0.1:8000 says  
you are out of server (away of the 100 kms)

OK

locality  
tirupati

city  
TIRUPATI

district  
chittoor

state  
Andhra Pradesh

pin  
517502

press here to get nearby  
orphanage

submit

The form consists of several input fields and a button. The background is a vibrant fruit collage. A modal window at the top displays an error message about being out of server range. Below the modal are fields for locality, city, district, state, and pin code. A green button with a link to nearby orphanages is present, along with a submit button.

## BACKEND PART OF SURPLUS FOOD:

The entered data of the surplus food by the donor is stored in the database of the website.

The screenshot shows the Django admin interface for a food registration model. The left sidebar contains a navigation menu with categories like AUTHENTICATION AND AUTHORIZATION, DONATE, DONATING THINGS, EVENTP, FOOD, GALLERY, and SPONSERING. Under the FOOD category, 'Foodregdetailss' is selected and highlighted in yellow. The main content area displays a form titled 'foodregdetails object (2)'. The form fields are as follows:

- Fname: BHARATH
- Lname: REDDY
- Email: pdbreddy@gmail.com
- Phone: 9381540744
- Food1: nonveg
- Food2: cooked
- Fooddate: Date: 2021-05-30 | Today |  Time: 11:46:00 | Now |
- Note: You are 5.5 hours ahead of server time.
- FoodQuantity: 150
- Foodpack: packed
- Locality: tirupati
- City: TIRUPATI
- District: chittoor
- State: Andhra Pradesh
- Pin: 517502

At the bottom of the form, there are three buttons: 'Delete', 'Save and add another', 'Save and continue editing', and a large blue 'SAVE' button.

These are the collaborated orphanage homes that we are working for. By clicking the Don't waste donate instead it will take the user to the page which the guidelines of the food donation is displayed.

• India reported the most number of deaths of children below five years in 2018, with 8,82,000 deaths,

Dont waste the food donate instead

Our partners

CHENNAI orphanage

HYDERABAD orphanage

THIRUPATI ORPHANAGE

GUNTUR

VIJAYAWADA orphanage

**ABOUT US**

Smile Foundations ,On our platform, you can donate to a nonprofit with confidence. Our robust due diligence process covers legal, compliance, financial and impact checks for each donation made to any non-profit, including in-person visits to beneficiaries and actual feedback reports. We do all this so that our donors can trust that their giving is indeed doing what it's supposed to do - change lives.

**Address**

Tirupati  
+91 6309716107  
smilefounations@gmail.com

f t i v

FUNCTION “sfp” is for surplus food page

FUNCTION “moreinfo” is for more information on food page .(guidelines of food donation )

FUNCTION “operations” is for operations page,”registerform” is for food details form.

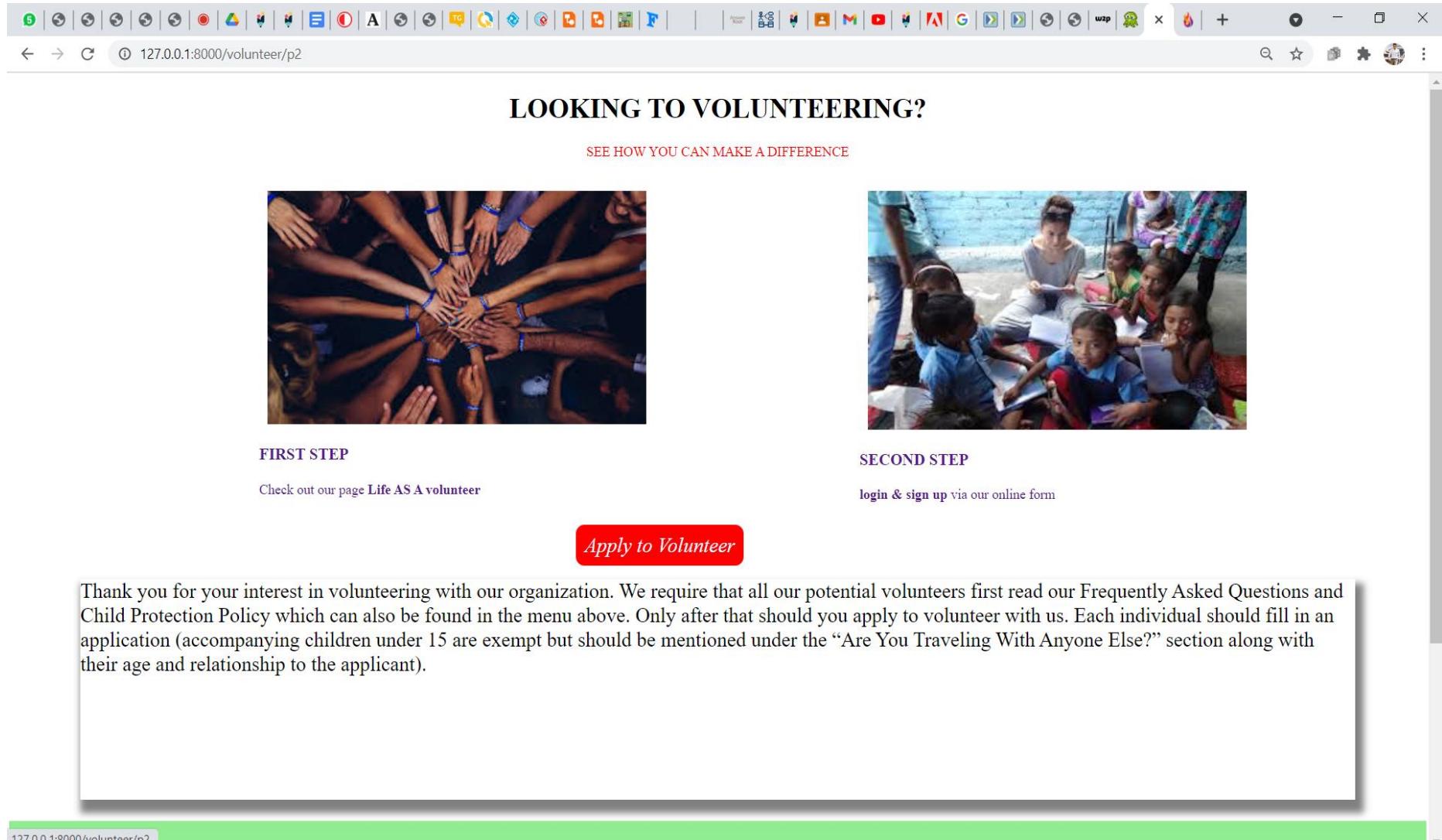
```
ORPHANAGE MANAGEMENT SYSTEM > food > views.py > registerform
1  from django.shortcuts import render
2  from django.shortcuts import redirect
3  from . models import foodregdetails
4  from django.http import HttpResponseRedirect
5  # Create your views here.
6  def sfp(request):
7      return render(request, "surplusfoodpage.html")
8  #this fuction is to recieve web request from user and redirect to the surplus food page of the website(web response)
9  def moreinfo(request):
10     return render(request, "moreinfo.html")
11 #this fuction is to recieve web request from user and redirect to the more info(guide lines for donating food) page of the website(web response)
12 def operations(request):
13     return render(request, "operations.html")
14 #this fuction is to recieve web request from user and redirect to the operations page(how food is circulated) of the website(web response)
15 def registerform(request):
16     if request.method == "POST":
17         fn = request.POST["fname"]
18         ln = request.POST["lname"]
19         em = request.POST["email"]
20         ph = request.POST["phone"]
21         f1 = request.POST["food1"]
22         f2 = request.POST["food2"]
23         fd = request.POST["fooddate"]
24         fq = request.POST["foodQuantity"]
25         fp = request.POST["foodpack"]
26         lc = request.POST["locality"]
27         ct = request.POST["city"]
28         dt = request.POST["district"]
29         st = request.POST["state"]
30         pn = request.POST["pin"]
31         data = foodregdetails(fname=fn, lname=ln, email=em, phone=ph, food1=f1, food2=f2, fooddate=fd, foodQuantity=fq, foodpack=fp, locality=lc, city=ct, district=dt, state=st, pin=pn)
32         data.save()
33     #this "registerform" function is for form in surplus food form page which seeks the basics details for donating surplus food
34     return HttpResponseRedirect("<h1>data successfully saved</h1>")
35     return redirect("sfp")
36     return render(request, "foodregister.html")
```

“foodregdetails” model class is for food details form Database.

```
ORPHANAGE_MANAGEMENT_SYSTEM > food > models.py > ...
1  from django.db import models
2
3  # Create your models here.
4
5  class foodregdetails(models.Model):
6
7      fname = models.CharField(max_length=50)
8      lname = models.CharField(max_length=50)
9      email = models.EmailField(max_length=254)
10     phone =models.BigIntegerField()
11     FOOD_CHOICES = (('M', 'veg'), ('F', 'nonveg'), ('U', 'both'))
12     food1 = models.CharField(max_length=1, choices=FOOD_CHOICES,null=True)
13     FOODSTATE_CHOICES = (('FRO', 'frozen'), ('FRE', 'fresh'), ('COOK', 'cooked'))
14     food2 = models.CharField(max_length=4, choices=FOODSTATE_CHOICES,null=True)
15     fooddate = models.DateTimeField(auto_now=False, auto_now_add=False)
16     foodQuantity = models.IntegerField(null=True)
17     FOOD_PACK = (('L', 'loose'), ('P', 'packed'))
18     foodpack = models.CharField(max_length=1, choices=FOOD_PACK,null=True)
19     locality = models.CharField(max_length=50)
20     city = models.CharField(max_length=50)
21     district = models.CharField(max_length=50)
22     state = models.CharField(max_length=50)
23     pin= models.CharField(max_length=10)
24 #this "foodregdetails" class in models.py represents tables or collection in our database for the form in the surplus food page of the website
25
26
```

## VOLUNTEERING:

By clicking on the VOLUNTEER button on the navigation bar it will take the user to the volunteer page of the website. In the volunteer page the user will find the article “LIFE AS A VOLUNTEER IN THE ORPHANAGES HOMES”.



The screenshot shows a web browser window with a toolbar at the top containing various icons. The address bar displays the URL "127.0.0.1:8000/volunteer/p2". The main content area has a title "LOOKING TO VOLUNTEERING?" and a subtitle "SEE HOW YOU CAN MAKE A DIFFERENCE". It features two images: one of hands joined in a circle and another of children sitting on the ground. Below each image is a step description and a link. A central red button says "Apply to Volunteer".

**SEE HOW YOU CAN MAKE A DIFFERENCE**

**FIRST STEP**  
Check out our page [Life AS A volunteer](#)

**SECOND STEP**  
[login & sign up](#) via our online form

**Apply to Volunteer**

Thank you for your interest in volunteering with our organization. We require that all our potential volunteers first read our Frequently Asked Questions and Child Protection Policy which can also be found in the menu above. Only after that should you apply to volunteer with us. Each individual should fill in an application (accompanying children under 15 are exempt but should be mentioned under the “Are You Traveling With Anyone Else?” section along with their age and relationship to the applicant).

By clicking the APPLY TO VOLUNTEER button it will take the user to volunteer registration form. By filling the details the user can register as a volunteer and work for the orphanage home.

Register Form

First Name  
govardhan

Last Name  
haik

Gender  
Male

your availability  
Weekends

Email  
rokesh@gmail.com

Phone Number  
09381540744

Role  
student

Address  
shanti nagar

GOOD TO HELP!

Send

## BACKEND PART OF VOLUNTEER:

The entered details of the user in the volunteer registration form are stored in the database of the website.

The screenshot shows the Django administration interface for a 'volunteer object (2)'. The left sidebar contains a navigation menu with sections like AUTHENTICATION AND AUTHORIZATION, DONATE, DONATINGTHINGS, EVENTP, FOOD, GALLERY, SPONSERING, and VOLUNTEER. Under the VOLUNTEER section, there are links for 'Groups' (+ Add), 'Users' (+ Add), 'Financialreports' (+ Add), 'Pays' (+ Add), 'Contactforms' (+ Add), 'Regformdthingss' (+ Add), 'Calenders' (+ Add), 'Eventforms' (+ Add), 'Eventphotos' (+ Add), 'Foodregdetailss' (+ Add), and 'Galleries' (+ Add). The main content area is titled 'Change volunteer' and shows the following fields for the volunteer object (2):

- Fname: govardhan
- Lname: naik
- Gender: Male
- Availability: weekends
- Email: rokesh@gmail.com
- Phone: 9381540744
- Role: student
- Address: shanti nagar
- Message: GOOD TO HELP

At the bottom right, there are three buttons: 'Delete' (red), 'Save and add another' (blue), 'Save and continue editing' (blue), and 'SAVE' (blue).

Function “p2” is for Volunteer Page.

Function “vl” is for Volunteer registration form.

Function “ragava” is for Life as Volunteer Page.

```
ORPHANAGE MANAGEMENT SYSTEM > volunteer > views.py > ...
1  from django.shortcuts import render
2  from .models import volunteer
3  from django.http import HttpResponseRedirect
4  # Create your views here.
5  def p2(request):
6      return render(request,"p2.html")
7  #this "p2" fuction is to recieve web request from user and redirect to the volunteer page of the website(web response)
8
9  def vl(request):
10     if request.method == "POST":
11         fn = request.POST["fname"]
12         ln = request.POST["lname"]
13         gender = request.POST["gender"]
14         avl = request.POST["availability"]
15         email = request.POST["email"]
16         ph = request.POST["phone"]
17         role = request.POST["role"]
18         adr = request.POST["address"]
19         msg = request.POST["message"]
20         volunteer_data=volunteer(fname=fn, lname=ln, gender=gender, availability=avl, email=email, phone=ph, role=role, address=adr, message=msg)
21         volunteer_data.save()
22         return HttpResponseRedirect("<h1>data successfully saved</h1>")
23     return render(request,"volregistration.html")
24 #this "vl" function is for volunteer form page which seeks the basic details of users about the website
25 def ragava(request):
26     return render(request,"p3.html")
27 #this "ragava" fuction is to recieve web request from user and redirect to the life as a volunteer page of the website(web response)
```

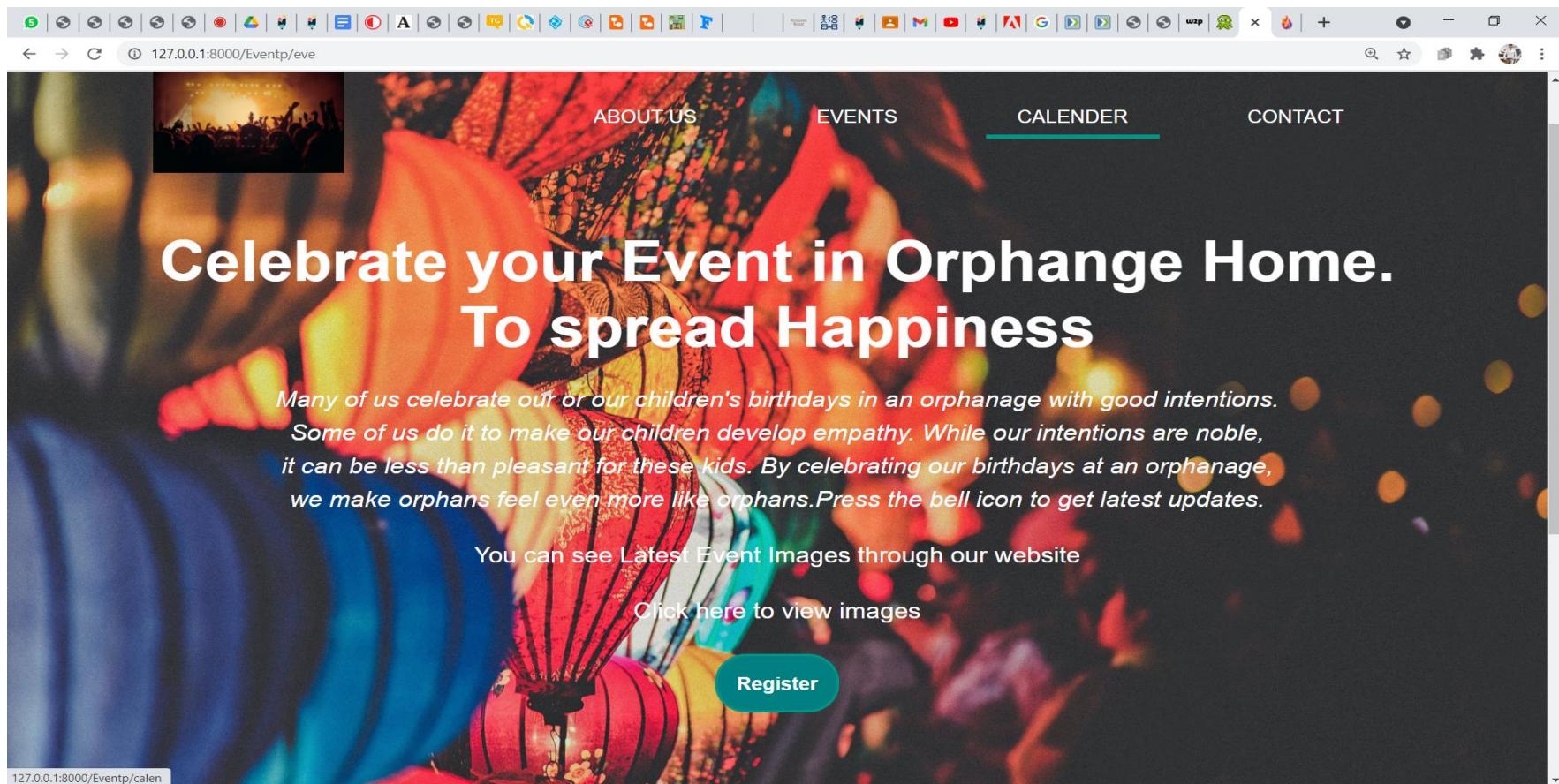
“volunteer” model class is for Volunteer form Database.

ORPHANAGE MANAGEMENT SYSTEM > volunteer >  models.py > ...

```
1  from django.db import models
2
3
4  # Create your models here.
5  class volunteer(models.Model):
6      fname = models.CharField(max_length=50)
7      lname = models.CharField(max_length=50)
8      Gender_CHOICES = (('male', 'Male'), ('female', 'Female'), ('other', 'Other'))
9      gender = models.CharField(max_length=7, choices=Gender_CHOICES)
10     Availability_CHOICES = (('WD', 'weekdays'), ('WE', 'weekends'), ('B', 'both'),('V','varies'))
11     availability = models.CharField(max_length=2, choices=Availability_CHOICES)
12     email = models.EmailField(max_length=200)
13     phone = models.BigIntegerField()
14     role = models.CharField(max_length=400)
15     address = models.CharField(max_length=50)
16     message = models.CharField(max_length=50)
17 #this "volunteer" class in models.py represents tables or collection in our database for the form in volunteer page of the website
```

## EVENT CELEBRATIONS:

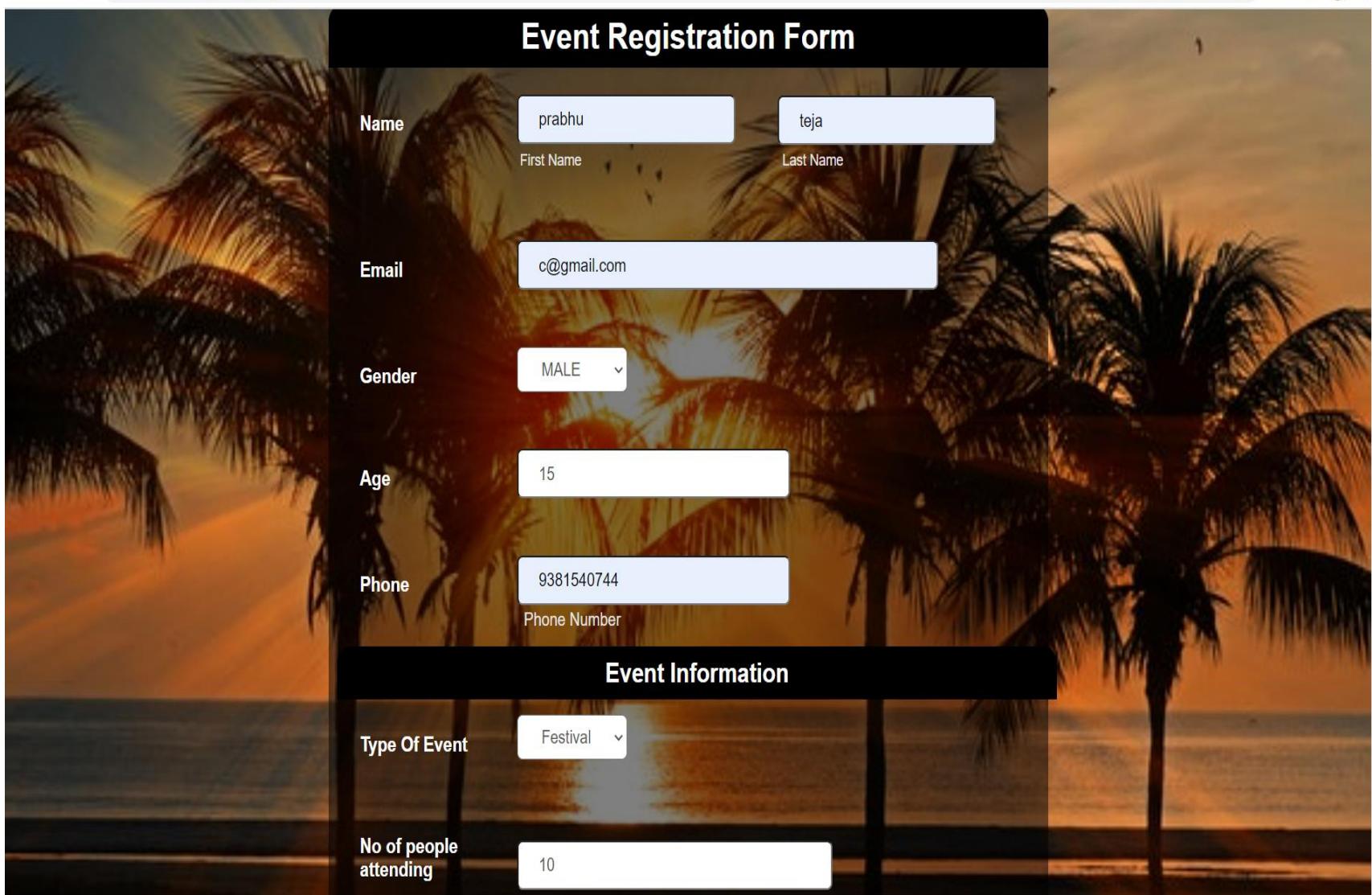
By clicking the EVENT button on the navigation bar it will take the user to the EVENT CELEBRATIONS page. People who are willing to celebrate their events in the orphanage they can register through our website. By clicking the REGISTER button it will take the user to the registration form .By filling the all the mandatory details in the registration form they can register their time slot by mentioning time slot.





← → C ① 127.0.0.1:8000/Eventp/reform

# Event Registration Form



Name

First Name  Last Name

Email

Gender

Age

Phone

Phone Number

## Event Information

Type Of Event

No of people attending

## BACKEND PART OF EVENTS:

The entered details of the register form by the user is stored in the database of the website.

Change eventform

Eventform object (1)

First name: KONGAMUDI

Last name: rokesh

Email: kmrokeshreddy@gmail.com

Gender: Male

Age: 20

Phone: 9381540744

Category: Birthday

People: 8

Event Date: Date: 2021-05-28 Today |

Time: 17:11:00 Now |

Note: You are 5.5 hours ahead of server time.

Event Celebrations: Sri Alluri Yuvajana Seva Sangam

Delete

Save and add another

Save and continue editing

**SAVE**

## EVENT CALENDAR:

By clicking the CALENDAR button on the navigation bar it will take the user to the calendar page.

The page contains the events celebrated in the orphanages on that respective day.



2021

**27**  
OCTOBER

**ROKESH BDAY**  
A great day on which the whole Indian people celebrate  
07.04 PM

**02**  
OCTOBER

**MAHATMA GANDHI JAYANTHI**  
A great day on which the whole Indian people celebrate  
07.04 AM

## BACKEND PART OF THE CALENDAR:

The admin can update the calendar for the users to celebrate the important events.

The screenshot shows the Django administration interface for adding a new calendar entry. The left sidebar has a 'EVENTP' section selected, which contains a 'Calenders' item. The main content area is titled 'Add calender' and includes fields for Date (02), Month (OCTOBER), Specialday (MAHATMA GANDHI JAYANTHI), Description (A great day on which the whole Indian people), and Time (07.04 AM). At the bottom right are three buttons: 'Save and add another', 'Save and continue editing', and a large blue 'SAVE' button.

Django administration

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

Home > Eventp > Calendars > Add calendar

AUTHENTICATION AND AUTHORIZATION

- Groups [+ Add](#)
- Users [+ Add](#)

DONATE

- Financialreports [+ Add](#)
- Pays [+ Add](#)

DONATINGTHINGS

- Contactforms [+ Add](#)
- Regformdthingss [+ Add](#)

« EVENTP

- Calenders [+ Add](#)
- Eventforms [+ Add](#)
- Eventphotos [+ Add](#)

FOOD

- Foodregdetailss [+ Add](#)

GALLERY

- Galleries [+ Add](#)

Add calender

Date: 02

Month: OCTOBER

Specialday: MAHATMA GANDHI JAYANTHI

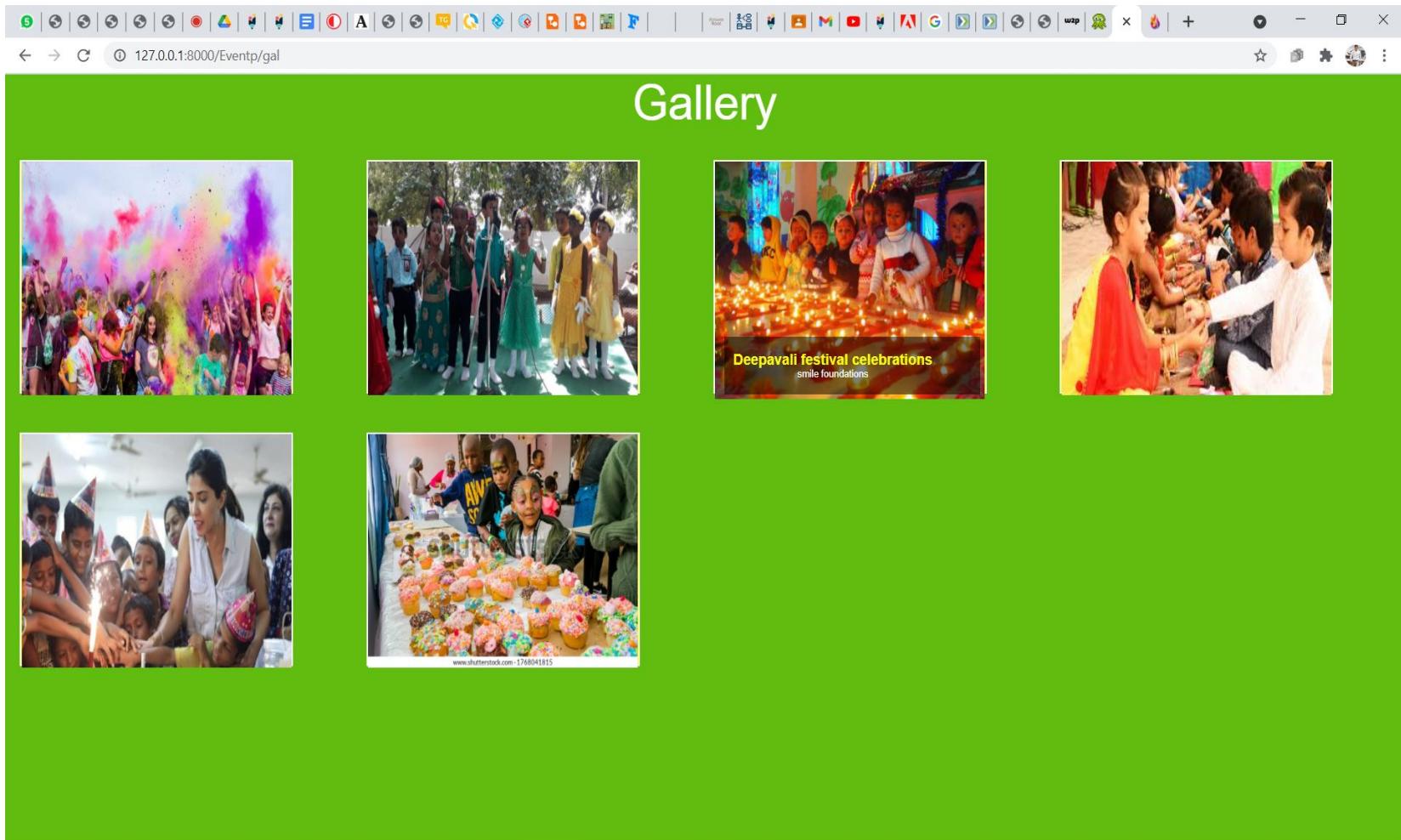
Description: A great day on which the whole Indian people

Time: 07.04 AM

Save and add another Save and continue editing **SAVE**

## EVENTS GALLERY:

By clicking the gallery button on the events page it takes the user to the event gallery page where it shows the previous events celebrated in the orphanage.



## BACKEND PART OF EVENTS GALLERY:

The admin can update the events celebration photos to the event page gallery.

The screenshot shows the Django administration interface for the 'eventphoto' model. The URL in the browser is `127.0.0.1:8000/admin/Eventp/eventphoto/1/change/`. The left sidebar has a 'EVENTP' section with 'Eventphotos' highlighted in yellow, indicating it's the current model being edited. The main content area shows the 'Change eventphoto' page for an 'eventphoto object (1)'. It displays two fields: 'Img:' with a preview of a photo named 'holi1\_kDMxgzi.jpg' and a file input field 'Choose File' containing 'holi.jpg'; and 'Info:' with the value 'holi celebrations'. At the bottom right are buttons for 'Delete', 'Save and add another', 'Save and continue editing', and a large 'SAVE' button.

"eve" function is to receive web request from user and redirect to the event page of the website(web response), this "calen" fuction is to recieve web request from user and redirect to the event calender of the website(web response), "regform" function is for event form page which seeks the basic details of users regarding event celebration.

```
ORPHANAGE MANAGEMENT SYSTEM > Eventp > views.py > regform
 2  from . models import Eventform
 3  from . models import eventphoto
 4  from . models import calender
 5  from django.http import HttpResponseRedirect
 6  # Create your views here.
 7  def eve(request):
 8      return render(request, "Event.html")
 9  #this "eve" fuction is to recieve web request from user and redirect to the event page of the website(web response)
10 # this is for event event page
11 def calen(request):
12     cal=calender.objects.all()
13     return render(request, "Calender.html",{'cal':cal})
14 #this "calen" fuction is to recieve web request from user and redirect to the event calender of the website(web response)
15 # this is for event calender page
16 def regform(request):
17     if request.method == "POST":
18         f1=request.POST["first_name"]
19         l1=request.POST["last_name"]
20         e=request.POST["email"]
21         ge=request.POST["Gender"]
22         a=request.POST["Age"]
23         ph=request.POST["Phone"]
24         CC=request.POST["Category"]
25         Pe=request.POST["People"]
26         EvD=request.POST["Event_Date"]
27         OH=request.POST["Event_Celebrations"]
28         data=Eventform(first_name=f1,last_name=l1,email=e,Gender=ge,Age=a,Phone=ph,Category=CC,People=Pe,Event_Date=EvD,Event_Celebrations=OH)
29         data.save()
30 #this "regform" function is for event form page which seeks the basic details of users regarding event celebrations
31         return HttpResponseRedirect("<h1>datasaved successfully</h1>")
32     return render(request, "HELP.html")
33 #this fuction is to recieve web request from user and redirect to the event form page of the website(web response)
34 def gal(request):
35     photo=eventphoto.objects.all()
36     return render(request,'newone.html',{'photo':photo})
37 #this "gal" fuction is to recieve web request from user and redirect to the event gallery of the website(web response)
38 # this is for event gallery page
```

"eventphoto","Eventform" and "calender" class in models.py represents tables or collection in our database for gallery page,form in event register page and in details in calender page of our website.

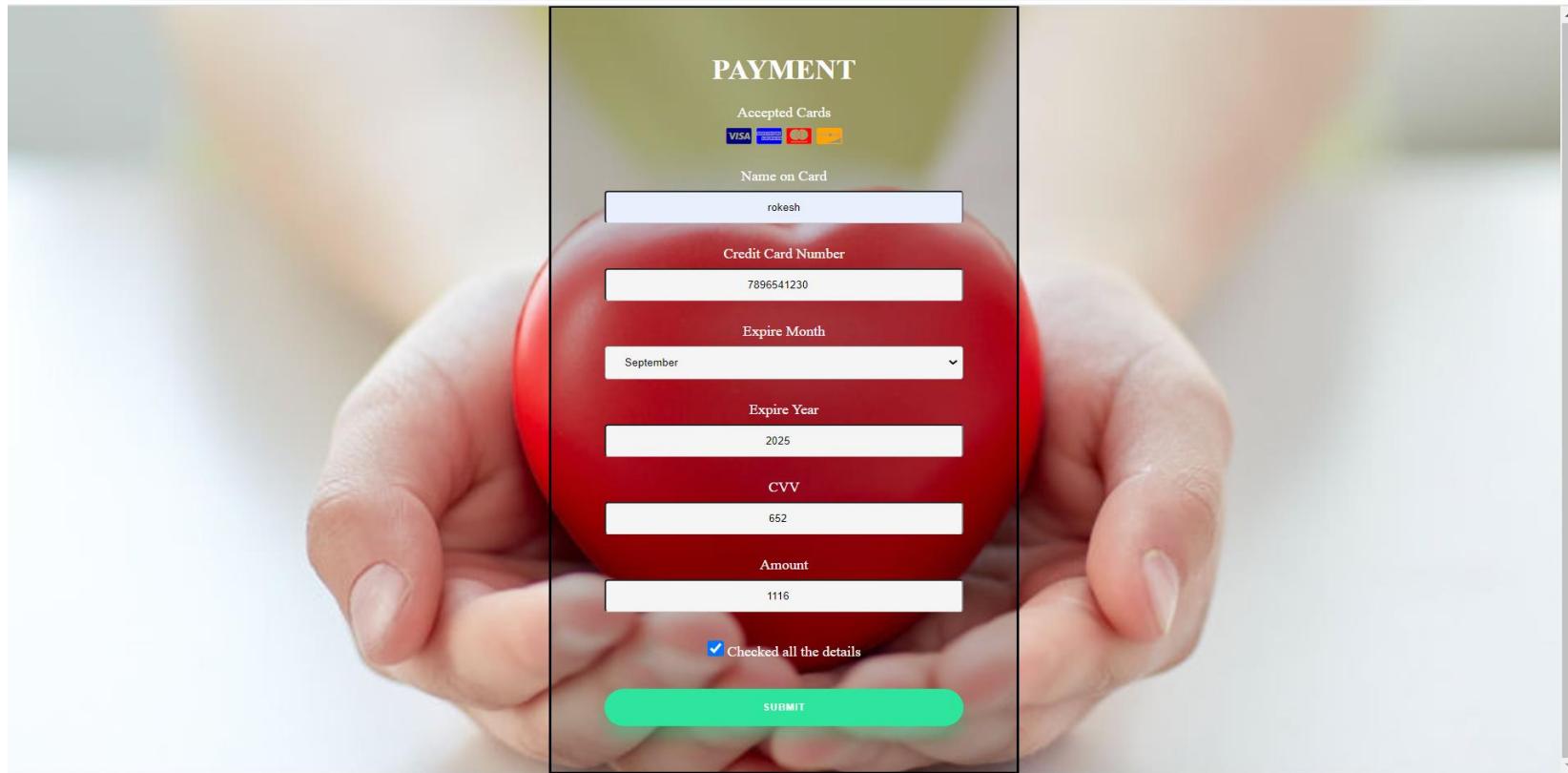
```
ORPHANAGE MANAGEMENT SYSTEM > Eventp > models.py > ...
1 # Create your models here.
2 from django.db import models
3 from django.core.files.storage import FileSystemStorage
4 # Create your models here.
5 class eventphoto(models.Model):
6     img=models.ImageField(upload_to="pictures",default="")
7     info=models.CharField(max_length=50,default="")
8 #this "eventphoto" class in models.py represents tables or collection in our database for gallery page of our website
9 #ImageField is used for storing valid image files into the database. One can any type of image file in ImageField.
10
11 class Eventform(models.Model):
12
13     first_name=models.CharField(max_length=50)
14     last_name=models.CharField(max_length=50)
15     email=models.EmailField(max_length=50)
16     GChoices = (('M', 'Male'), ('F', 'Female'), ('O', 'Others'))
17     Gender = models.CharField(max_length=1, choices=GChoices,null=True)
18     Age=models.IntegerField()
19     Phone=models.BigIntegerField()
20     CCategory = ((('F', 'Festival'), ('B', 'Birthday'), ('A', 'Anniversary')))
21     Category = models.CharField(max_length=1, choices=CCategory,null=True)
22     People=models.IntegerField()
23     Event_Date = models.DateTimeField()
24     Orphange_Home = ((('2','Sri Alluri Yuvajana Seva Sangam'),('3','Mr Mani'),('4','Childrens Home')))
25     Event_Celebrations=models.CharField(max_length=1,choices=Orphange_Home,null=True)
26 #this "Eventform" class in models.py represents tables or collection in our database for the form in event register page of the website
27 class calender(models.Model):
28     date=models.CharField(max_length=3)
29     month=models.CharField(max_length=15)
30     specialday=models.CharField(max_length=50)
31     description=models.CharField(max_length=250)
32     time=models.CharField(max_length=10)
33 #this "calender" class in models.py represents tables or collection in our database for the details in calender page of the website|
```

## DONATE:

By clicking the DONATE button on the navigation bar of the website it will take the user to the donate page. The donation page showcase the respective years FINANCIAL reports and also AUDIT reports. And the page shows the user how the donation money used and where it goes.

## PAYMENT FORM:

By clicking the DONATE NOW button on the donate page The website will take the user to the donation payment form page. Here the user should fill the payment related details in the provided payment form.



**PAYMENT**

Accepted Cards  
   

Name on Card

Credit Card Number

Expire Month

Expire Year

CVV

Amount

Checked all the details

**SUBMIT**

## BACKEND PART OF THE PAYMENT FORM:

The Name of the cardholder and amount given by the user is stored in the website database.

The screenshot shows the Django administration interface at the URL `127.0.0.1:8000/admin/DONATE/pay/2/change/`. The left sidebar contains a navigation menu with categories like AUTHENTICATION AND AUTHORIZATION, DONATE, DONATING THINGS, EVENTP, FOOD, GALLERY, SPONSERING, and VOLUNTEER. Under the DONATE category, the 'Pays' model is selected. The main content area is titled 'Change pay' and shows a 'pay object (2)'. It has two fields: 'Name:' with the value 'rokesh' and 'Amount:' with the value '1116'. At the bottom right are buttons for 'Delete', 'Save and add another', 'Save and continue editing', and 'SAVE'.

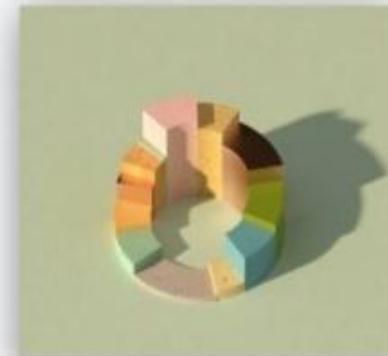
## FINANCIAL & AUDIT REPORTS:

The donation page showcase the respective years FINANCIAL reports and also AUDIT reports. And the page shows the user how the donation money used and where it goes.

### Where the money goes...

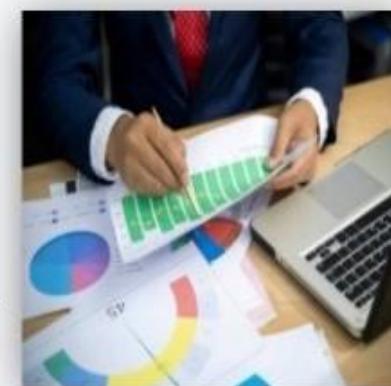
We believe that transparency is extremely important and want to be clear that there are absolutely zero administration fees deducted by us from donations independent of the banking/platform charges.

We are so fortunate to be able to pay all of our own administration fee's, salaries and similar costs through independent funding programs such as our school, workshop, farm and, occasionally, by donations specifically given by family and friends specifically for these expenses.



### Financial Reports & Audit Reports

Transparency is important for us and a huge part of what we do is ensuring that supporters know where their money is going. We have always worked hard to accurately account for and report the financial position of our organization. We have a culture of accountability and integrity and are committed to doing the right thing. Many organizations choose to only present their Annual Reports to the public for viewing but in the spirit of full disclosure we have chosen to make available our Audited Financial Reports as well as regular declaration of our Foreign Contribution intake. For those who do not know the Financial Year in India is from April 1st to March 31st each year.



After clicking the buttons FINANCIAL and AUDIT reports it will take the user to FINANCIAL and AUDIT reports files page.

The screenshot shows a web browser window with the URL [127.0.0.1:8000/DONATE/finpdfs](http://127.0.0.1:8000/DONATE/finpdfs). The main content is a banner image featuring a person in a suit working on a laptop, with the text "FINANCIAL & AUDIT REPORTS" overlaid. Below the banner, there are three sections, each representing a financial year:

- 2020-21**: Contains a green button labeled "2020-21 FINANCIAL REPORT".
- 2019-20**: Contains a white button labeled "2019-20 FINANCIAL REPORT" and a white button labeled "2019-20 ANNUAL REPORT".
- 2018-19**: Contains a white button labeled "2018-19 FINANCIAL REPORT" and a white button labeled "2018-19 ANNUAL REPORT".

A small address bar at the bottom left shows the file path [127.0.0.1:8000/media/pdfreports/LAB01\\_WBW35aa.pdf](http://127.0.0.1:8000/media/pdfreports/LAB01_WBW35aa.pdf).

## BACKEND OF THE FINANCIAL & AUDIT reports:

The Admin can update the FINANCIAL & AUDIT reports of the respective years from the admin panel of the website.

The screenshot shows the Django Admin interface at the URL `127.0.0.1:8000/admin/DONATE/financialreport/add/`. The page title is "Django administration". The left sidebar contains a navigation menu with sections: AUTHENTICATION AND AUTHORIZATION (Groups, Users), DONATE (Financialreports, Pays), DONATINGTHINGS (Contactforms, Regformdthingss), EVENTP (Calenders, Eventforms, Eventphotos), FOOD (Foodregdetailss), and GALLERY (Galleries). The main content area is titled "Add financialreport". It has five input fields: "Fin:" with a file input showing "10-Kongamu...on -2020.pdf", "Fina:" with a file input showing "ac83c8d5-3...6306177e.pdf", "Year:" with the value "2020-21", "Yearf:" with the value "2020-21 FINANCIAL REPORT", and "Yeara:" with the value "2020-21 ANNUAL REPORT". At the bottom right are three buttons: "Save and add another", "Save and continue editing", and a large blue "SAVE" button.

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

Home › Donate › Financialreports › Add financialreport

AUTHENTICATION AND AUTHORIZATION

Groups + Add

Users + Add

DONATE

Financialreports + Add

Pays + Add

DONATINGTHINGS

Contactforms + Add

Regformdthingss + Add

EVENTP

Calenders + Add

Eventforms + Add

Eventphotos + Add

FOOD

Foodregdetailss + Add

GALLERY

Galleries + Add

Add financialreport

Fin: Choose File 10-Kongamu...on -2020.pdf

Fina: Choose File ac83c8d5-3...6306177e.pdf

Year: 2020-21

Yearf: 2020-21 FINANCIAL REPORT

Yeara: 2020-21 ANNUAL REPORT

Save and add another Save and continue editing SAVE

"dont" function is to receive web request from user and redirect to the donate page of the website(web response), "dtreform" function is for form in donating things form page which seeks the payment details of sponsors.

```
ORPHANAGE MANAGEMENT SYSTEM > DONATE > views.py > ...
1  from django.shortcuts import render
2  from django.http import HttpResponse
3  from .models import pay
4  from .models import financialreport
5  # Create your views here.
6  def dont(request):
7      return render(request, "donate.html")
8
9  #this fuction is to recieve web request from user and redirect to the donate page of the website(web response)
10 def finreprt(request):
11
12     return render(request,'finrep.html')
13
14 #this fuction is to recieve web request from user and redirect to the financial report page of the website(web response)
15 def payment(request):
16     if request.method=="POST":
17
18         cardname=request.POST["name"]
19         damount=request.POST["amount"]
20         data=pay(name=cardname,amount=damount)
21         data.save()
22         return HttpResponse("<h1>submitted succesfully</h1>")
23
24
25
26 #this "dtreform" function is for form in donating things form page which seeks the payment details of sponsors
27     return render(request,"payments.html")
28 def finpdfs(request):
29
30     finr=financialreport.objects.all()
31     return render(request,'finrepdf.html',{'finr':finr})
32
33 #this fuction is to recieve web request from user and redirect to the financial report page of the website(web response)
```

"pay" class in models.py represents tables or collection in our database for the payment of the website

"financialreport" class in models.py represents tables or collection in our database for the pdf files in financial report page of the website #FileField is used for storing valid pdf files into the database. One can any type of image file in FileField.

```
ORPHANAGE MANAGEMENT SYSTEM > DONATE > 📄 models.py > ...
1  from django.db import models
2  from django.core.files.storage import FileSystemStorage
3  # Create your models here.
4  class pay(models.Model):
5
6      id = models.AutoField(primary_key=True)
7      name=models.CharField(max_length=200)
8      amount=models.BigIntegerField()
9  #this "pay" class in models.py represents tables or collection in our database for the payment of the website
10
11 class financialreport(models.Model):
12     fin=models.FileField(upload_to="pdfreports")
13     fina=models.FileField(upload_to="pdfreports")
14     year=models.CharField(max_length=50)
15     yearf=models.CharField(max_length=50)
16     yeara=models.CharField(max_length=50)
17 #this "financialreport" class in models.py represents tables or collection in our database for the pdf files in financial report page of the webs
18 #FileField is used for storing valid pdf files into the database. One can any type of image file in FileField.|
```

## DATABASE QUERIES

CREATE TABLE "food\_foodregdetails" ("id" serial NOT NULL PRIMARY KEY, "fname" varchar(50) NOT NULL, "lname" varchar(50) NOT NULL, "email" varchar(254) NOT NULL, "phone" bigint NOT NULL, "food1" varchar(1) NULL, "food2" varchar(4) NULL, "fooddate" timestamp with time zone NOT NULL, "foodQuantity" integer NULL, "foodpack" varchar(1) NULL, "locality" varchar(50) NOT NULL, "city" varchar(50) NOT NULL, "district" varchar(50) NOT NULL, "state" varchar(50) NOT NULL, "pin" varchar(10) NOT NULL); => THIS STORES SURPLUS FOOD DETAILS ENTER BY THE USER

CREATE TABLE "DONATE\_pay" ("id" serial NOT NULL PRIMARY KEY, "name" varchar(200) NOT NULL, "amount" bigint NOT NULL); => THIS STORES DONOR NAME & AMOUNT.

CREATE TABLE "DONATE\_financialreport" ("id" bigserial NOT NULL PRIMARY KEY, "fin" varchar(100) NOT NULL, "fina" varchar(100) NOT NULL, "year" varchar(50) NOT NULL, "yearf" varchar(50) NOT NULL, "yeara" varchar(50) NOT NULL); => THIS STORES FINANCIAL & AUDIT reports pdf files.

CREATE TABLE "donatingthings\_regformdthings" ("id" bigserial NOT NULL PRIMARY KEY, "fname" varchar(50) NOT NULL, "lname" varchar(50) NOT NULL, "email" varchar(254) NOT NULL, "gender" varchar(1) NULL, "age" integer NOT NULL, "pnumber" bigint NOT NULL,

"ntoys" varchar(50) NOT NULL, "nbooks" varchar(50) NOT NULL, "nblankets" varchar(50) NOT NULL, "dress" varchar(50) NOT NULL, "cgender" varchar(1) NULL, "others" varchar(50) NOT NULL, "adress" varchar(5000) NOT NULL, "landmark" varchar(50) NOT NULL, => THIS STORES THINGS DONATION FORM DETAILS.

CREATE TABLE "donatingthings\_contactform" ("id" bigserial NOT NULL PRIMARY KEY, "name" varchar(50) NOT NULL, "phone" bigint NOT NULL, "email" varchar(254) NOT NULL, "subject" varchar(50) NOT NULL, "message" varchar(100) NOT NULL); => THIS STORES USER QUERIES PROVIDED IN THE CONTACT US PAGE.

CREATE TABLE "Eventp\_calender" ("id" bigserial NOT NULL PRIMARY KEY, "date" varchar(3) NOT NULL, "month" varchar(15) NOT NULL, "specialday" varchar(50) NOT NULL, "description" varchar(250) NOT NULL, "time" varchar(10) NOT NULL); => THIS STORES EVENT CALENDAR DATA

CREATE TABLE "Eventp\_eventform" ("id" bigserial NOT NULL PRIMARY KEY, "first\_name" varchar(50) NOT NULL, "last\_name" varchar(50) NOT NULL, "email" varchar(50) NOT NULL, "Gender" varchar(1) NULL, "Age" integer NOT NULL, "Phone" bigint NOT NULL, "Category" varchar(1) NULL, "People" integer NOT NULL, "Event\_Date" timestamp with time zone NOT NULL, "Event\_Celebrations" varchar(1) NULL); => THIS STORES EVENT DETAILS ENTERED BY USER.

CREATE TABLE "Eventp\_eventphoto" ("id" bigserial NOT NULL PRIMARY KEY, "img" varchar(100) NOT NULL, "info" varchar(50) NOT NULL); => THIS STORES PHOTOS IN EVENT GALLERY PAGE .

CREATE TABLE "gallery\_gallery" ("id" bigserial NOT NULL PRIMARY KEY, "img" varchar(100) NOT NULL, "info" varchar(50) NOT NULL); => THIS STORES PHOTOS IN MAIN GALLERY PAGE.

CREATE TABLE "sponsoring\_imgs" ("id" serial NOT NULL PRIMARY KEY, "img" varchar(100) NOT NULL, "name" varchar(50) NOT NULL, "age" varchar(50) NOT NULL); => THIS STORES THE PHOTOS, DETAILS OF ORPHANS WAITING FOR ORPHANS.

CREATE TABLE "sponsoring\_payment" ("id" serial NOT NULL PRIMARY KEY, "name" varchar(200) NOT NULL, "card" bigint NOT NULL); => THIS STORES THE DETAILS ENTERED BY SPONSOR.

CREATE TABLE "sponsoring\_virat" ("id" serial NOT NULL PRIMARY KEY, "fname" varchar(50) NOT NULL, "lname" varchar(50) NOT NULL, "gender" varchar(7) NULL, "email" varchar(50) NOT NULL, "phone" bigint NOT NULL, "Role" varchar(200) NOT NULL, "amount" integer NOT NULL, "address" varchar(500) NOT NULL, "message" varchar(200) NOT NULL); => THIS

STORES THE DETAILS ENTERED BY THE SPONSOR.

CREATE TABLE "volunteer\_volunteer" ("id" serial NOT NULL PRIMARY KEY, "fname" varchar(50) NOT NULL, "lname" varchar(50) NOT NULL, "gender" varchar(7) NOT NULL, "availability" varchar(2) NOT NULL, "email" varchar(200) NOT NULL, "phone" bigint NOT NULL, "role" varchar(400) NOT NULL, "address" varchar(50) NOT NULL, "message" varchar(50) NOT NULL); => THIS STORES THE DETAILS ENTERED BY THE USER FOR VOLUNTEER REGISTRATION.