**SOURCE CODE**

**All urls.py**

*"""RetinopathyofPrematurity URL Configuration  
  
The `urlpatterns` list routes URLs to views. For more information please see:  
 https://docs.djangoproject.com/en/2.0/topics/http/urls/  
Examples:  
Function views  
 1. Add an import: from my\_app import views  
 2. Add a URL to urlpatterns: path('', views.home, name='home')  
Class-based views  
 1. Add an import: from other\_app.views import Home  
 2. Add a URL to urlpatterns: path('', Home.as\_view(), name='home')  
Including another URLconf  
 1. Import the include() function: from django.urls import include, path  
 2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))  
"""***from** django.contrib **import** admin  
**from** django.urls **import** path  
**from** RetinopathyofPrematurity **import** views **as** mainView  
**from** users **import** views **as** usr  
**from** admins **import** views **as** admins  
**from** Maternity **import** views **as** rop  
  
urlpatterns = [  
 path(**'admin/'**, admin.site.urls),  
 path(**""**,mainView.index, name=**"index"**),  
 path(**"index/"**, mainView.index, name=**"index"**),  
 path(**"logout/"**, mainView.logout, name=**"logout"**),  
 path(**"UserLogin/"**, mainView.UserLogin, name=**"UserLogin"**),  
 path(**"AdminLogin/"**, mainView.AdminLogin, name=**"AdminLogin"**),  
 path(**"UserRegister/"**, mainView.UserRegister, name=**"UserRegister"**),  
 path(**"MaternityLogin/"**, mainView.MaternityLogin, name=**"MaternityLogin"**),  
  
 *### User Side Views* path(**"UserRegisterActions/"**, usr.UserRegisterActions, name=**"UserRegisterActions"**),  
 path(**"UserLoginCheck/"**, usr.UserLoginCheck, name=**"UserLoginCheck"**),  
 path(**"UserHome/"**, usr.UserHome, name=**"UserHome"**),  
 path(**"UserViewData/"**, usr.UserViewData, name=**"UserViewData"**),  
 path(**"UserLinearModel/"**, usr.UserLinearModel, name=**"UserLinearModel"**),  
 path(**"Userga2m/"**, usr.Userga2m, name=**"Userga2m"**),  
  
 *#### Admin Side views* path(**"AdminLoginCheck/"**, admins.AdminLoginCheck, name=**"AdminLoginCheck"**),  
 path(**"AdminHome/"**, admins.AdminHome, name=**"AdminHome"**),  
 path(**"ViewAllUsers/"**, admins.ViewAllUsers, name=**"ViewAllUsers"**),  
 path(**"AdminActivaUsers/"**, admins.AdminActivaUsers, name=**"AdminActivaUsers"**),  
 path(**"AdminLinearResults/"**, admins.AdminLinearResults, name=**"AdminLinearResults"**),  
 path(**"AdminGA2Results/"**, admins.AdminGA2Results, name=**"AdminGA2Results"**),  
  
 *#Maternity URLS* path(**"MaternityLoginCheck/"**, rop.MaternityLoginCheck, name=**"MaternityLoginCheck"**),  
 path(**"MaternityHome/"**, rop.MaternityHome, name=**"MaternityHome"**),  
 path(**"MaternityAddDataForm/"**, rop.MaternityAddDataForm, name=**'MaternityAddDataForm'**),  
 path(**"MaternityAddDataAction/"**, rop.MaternityAddDataAction, name=**"MaternityAddDataAction"**),  
 path(**"MaternityGa2m/"**, rop.MaternityGa2m, name=**"MaternityGa2m"**),  
 path(**"MaternityViewGA2mResults/"**, rop.MaternityViewGA2mResults, name=**"MaternityViewGA2mResults"**),  
  
  
  
]

Main views.py

**from** django.shortcuts **import** render  
**from** users.forms **import** UserRegistrationForm  
  
  
**def** index(request):  
 **return** render(request, **'index.html'**, {})  
  
**def** logout(request):  
 **return** render(request, **'index.html'**, {})  
  
**def** UserLogin(request):  
 **return** render(request, **'UserLogin.html'**, {})  
  
**def** AdminLogin(request):  
 **return** render(request, **'AdminLogin.html'**, {})  
**def** MaternityLogin(request):  
 **return** render(request,**"MaternityLogin.html"**,{})  
  
**def** UserRegister(request):  
 form = UserRegistrationForm()  
 **return** render(request, **'UserRegistrations.html'**, {**'form'**: form})

user side **views.py**

**from** django.shortcuts **import** render, HttpResponse  
**from** .forms **import** UserRegistrationForm  
**from** .models **import** UserRegistrationModel  
**from** django.contrib **import** messages  
**from** Maternity.models **import** RetinopathyofPrematureModel  
**from** .models **import** UserLinearResultModel,UserGA2MResultModel  
**from** django\_pandas.io **import** read\_frame  
  
*# Create your views here.***def** UserRegisterActions(request):  
 **if** request.method == **'POST'**:  
 form = UserRegistrationForm(request.POST)  
 **if** form.is\_valid():  
 print(**'Data is Valid'**)  
 form.save()  
 messages.success(request, **'You have been successfully registered'**)  
 form = UserRegistrationForm()  
 **return** render(request, **'UserRegistrations.html'**, {**'form'**: form})  
 **else**:  
 messages.success(request, **'Email or Mobile Already Existed'**)  
 print(**"Invalid form"**)  
 **else**:  
 form = UserRegistrationForm()  
 **return** render(request, **'UserRegistrations.html'**, {**'form'**: form})  
  
  
**def** UserLoginCheck(request):  
 **if** request.method == **"POST"**:  
 loginid = request.POST.get(**'loginname'**)  
 pswd = request.POST.get(**'pswd'**)  
 print(**"Login ID = "**, loginid, **' Password = '**, pswd)  
 **try**:  
 check = UserRegistrationModel.objects.get(loginid=loginid, password=pswd)  
 status = check.status  
 print(**'Status is = '**, status)  
 **if** status == **"activated"**:  
 request.session[**'id'**] = check.id  
 request.session[**'loggeduser'**] = check.name  
 request.session[**'loginid'**] = loginid  
 request.session[**'email'**] = check.email  
 print(**"User id At"**, check.id, status)  
 **return** render(request, **'users/UserHome.html'**, {})  
 **else**:  
 messages.success(request, **'Your Account Not at activated'**)  
 **return** render(request, **'UserLogin.html'**)  
 **except** Exception **as** e:  
 print(**'Exception is '**, str(e))  
 **pass** messages.success(request, **'Invalid Login id and password'**)  
 **return** render(request, **'UserLogin.html'**, {})  
  
  
**def** UserHome(request):  
 **return** render(request, **'users/UserHome.html'**, {})  
  
**def** UserViewData(request):  
 data = RetinopathyofPrematureModel.objects.all()  
 **return** render(request, **'users/UserViewAllData.html'**, {**'data'**: data})  
  
**def** UserLinearModel(request):  
 data = RetinopathyofPrematureModel.objects.all()  
 df = read\_frame(data)  
 print(df.head())  
 *# import statsmodels.api as sm  
 # x = df['gestationalweek','mechanicalventilation']  
 # y = df['gender']  
 # exog, endog = sm.add\_constant(x), y  
 # mod = sm.GLM(endog, exog, family=sm.families.Poisson(link=sm.families.links.log))  
 # res = mod.fit()* **from** sklearn.model\_selection **import** train\_test\_split  
 **from** sklearn.linear\_model **import** LogisticRegression  
 **from** sklearn.metrics **import** accuracy\_score,precision\_score, recall\_score  
 X = df.iloc[:, 1:10].values  
 y = df.iloc[:, 3].values  
 X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=1 / 3, random\_state=0)  
  
 model = LogisticRegression()  
 model.fit(X\_train, y\_train)  
 y\_pred = model.predict(X\_test)  
 accuracy = accuracy\_score(y\_pred,y\_test)  
 precesion = precision\_score(y\_pred, y\_test, pos\_label = **'positive'**, average = **'micro'**)  
 recall = recall\_score(y\_test, y\_pred, pos\_label = **'positive'**, average = **'micro'**)  
 print(**"Precesion = "**, precesion)  
 **from** .ChiSquaretest **import** TestChiSquare  
 obj = TestChiSquare()  
 chitest = obj.findTestResult(df)  
 print(**"Chisquare Test "**,chitest)  
 loginid =request.session[**'loginid'**]  
 UserLinearResultModel.objects.create(name=loginid,accuracy=accuracy, precesion=precesion,recall=recall,chitest=chitest)  
 **return** render(request, **"users/LinearResult.html"**,{**'accuracy'**:accuracy,**'precesion'**:precesion,**'recall'**:recall, **'chitest'**:chitest})  
  
  
**def** Userga2m(request):  
 data = RetinopathyofPrematureModel.objects.all()  
 df = read\_frame(data)  
 X = df.iloc[:, 1].values.astype(int)  
 y = df.iloc[:, 3].values.astype(int)  
 *# Poisson regression code* **import** statsmodels.api **as** sm  
 exog, endog = sm.add\_constant(X), y  
 mod = sm.GLM(endog, exog,  
 family=sm.families.Poisson(link=sm.families.links.log))  
 res = mod.fit()  
 print(res.params)  
 scale = res.scale  
 deviance = res.deviance  
 pearson\_chi2 = res.pearson\_chi2  
 llf = res.llf  
 loginid = request.session[**'loginid'**]  
 print(**"Scale ="**,scale,**" Devince = "**,deviance, **" pearsonc = "**,pearson\_chi2,**" llf = "**,llf)  
 UserGA2MResultModel.objects.create(name=loginid, scale=scale, deviance=deviance, pearson\_chi2=pearson\_chi2,llf=llf)  
 **return** render(request, **"users/LinearGa2mResult.html"**,{**'scale'**:scale,**'deviance'**:deviance,**'pearson\_chi2'**:pearson\_chi2, **'llf'**:llf})

ChiSquareTest.py

**from** scipy.stats **import** chi2\_contingency  
**class** TestChiSquare:  
 **def** findTestResult(self,df):  
 dfd = df  
 X = df.iloc[:, 1:3].values.astype(int)  
 y = df.iloc[:, 4:6].values.astype(int)  
 df = [[X], [y]]  
 table = [X,y]  
 *#print(table)* stat, p, dof, expected = chi2\_contingency(table)  
  
 *# interpret p-value* alpha = 0.05  
 print(**"p value is "** + str(p))  
 **if** p <= alpha:  
 print(**'Dependent (reject H0)'**)  
 **else**:  
 print(**'Independent (H0 holds true)'**)  
  
 **pass** *# import scipy.stats as stats  
 # x = dfd[:, :0].values.astype(int)  
 # y = dfd[:, :1].values.astype(int)  
 # table = [x, y]  
 # oddsratio, pvalue = stats.fisher\_exact()  
 # print("OddsR: ", oddsratio, "p-Value:", pvalue)  
 # import pandas as pd  
 # dfd = pd.DataFrame([list(range(5))], columns=["a{}".format(i) for i in range(5)])  
 # print(dfd.iloc[0])  
 # print(type(dfd))  
 # sr = pd.Series(dfd)  
 # print(dfd.pow(sr, axis=1))* **return** str(p)

user side **models.py**

**from** django.db **import** models  
  
*# Create your models here.***class** UserRegistrationModel(models.Model):  
 name = models.CharField(max\_length=100)  
 loginid = models.CharField(unique=**True**, max\_length=100)  
 password = models.CharField(max\_length=100)  
 mobile = models.CharField(unique=**True**, max\_length=100)  
 email = models.CharField(unique=**True**, max\_length=100)  
 locality = models.CharField(max\_length=100)  
 address = models.CharField(max\_length=1000)  
 city = models.CharField(max\_length=100)  
 state = models.CharField(max\_length=100)  
 status = models.CharField(max\_length=100)  
  
 **def** \_\_str\_\_(self):  
 **return** self.loginid  
  
 **class** Meta:  
 db\_table = **'UserRegistrations'  
  
class** UserLinearResultModel(models.Model):  
 name = models.CharField(max\_length=100)  
 accuracy = models.CharField(max\_length=100)  
 precesion = models.CharField(max\_length=100)  
 recall = models.CharField(max\_length=100)  
 chitest = models.CharField(max\_length=100)  
  
  
 **def** \_\_str\_\_(self):  
 **return** self.id  
  
 **class** Meta:  
 db\_table = **'LinearResults'  
  
class** UserGA2MResultModel(models.Model):  
 name = models.CharField(max\_length=100)  
 scale = models.CharField(max\_length=100)  
 deviance = models.CharField(max\_length=100)  
 pearson\_chi2 = models.CharField(max\_length=100)  
 llf = models.CharField(max\_length=100)  
  
  
 **def** \_\_str\_\_(self):  
 **return** self.id  
  
 **class** Meta:  
 db\_table = **'GA2MResults'**

user forms.py

**from** django **import** forms  
**from** .models **import** UserRegistrationModel  
  
  
**class** UserRegistrationForm(forms.ModelForm):  
 name = forms.CharField(widget=forms.TextInput(attrs={**'pattern'**: **'[a-zA-Z]+'**}), required=**True**, max\_length=100)  
 loginid = forms.CharField(widget=forms.TextInput(attrs={**'pattern'**: **'[a-zA-Z]+'**}), required=**True**, max\_length=100)  
 password = forms.CharField(widget=forms.PasswordInput(attrs={**'pattern'**: **'(?=.\*\d)(?=.\*[a-z])(?=.\*[A-Z]).{8,}'**,  
 **'title'**: **'Must contain at least one number and one uppercase and lowercase letter, and at least 8 or more characters'**}),  
 required=**True**, max\_length=100)  
 mobile = forms.CharField(widget=forms.TextInput(attrs={**'pattern'**: **'[56789][0-9]{9}'**}), required=**True**,  
 max\_length=100)  
 email = forms.CharField(widget=forms.TextInput(attrs={**'pattern'**: **'[a-z0-9.\_%+-]+@[a-z0-9.-]+\.[a-z]{2,}$'**}),  
 required=**True**, max\_length=100)  
 locality = forms.CharField(widget=forms.TextInput(), required=**True**, max\_length=100)  
 address = forms.CharField(widget=forms.Textarea(attrs={**'rows'**: 4, **'cols'**: 22}), required=**True**, max\_length=250)  
 city = forms.CharField(widget=forms.TextInput(  
 attrs={**'autocomplete'**: **'off'**, **'pattern'**: **'[A-Za-z ]+'**, **'title'**: **'Enter Characters Only '**}), required=**True**,  
 max\_length=100)  
 state = forms.CharField(widget=forms.TextInput(  
 attrs={**'autocomplete'**: **'off'**, **'pattern'**: **'[A-Za-z ]+'**, **'title'**: **'Enter Characters Only '**}), required=**True**,  
 max\_length=100)  
 status = forms.CharField(widget=forms.HiddenInput(), initial=**'waiting'**, max\_length=100)  
  
 **class** Meta():  
 model = UserRegistrationModel  
 fields = **'\_\_all\_\_'**

user side **views.py**

**from** django.shortcuts **import** render  
**from** django.contrib **import** messages  
**from** users.models **import** UserRegistrationModel,UserLinearResultModel,UserGA2MResultModel  
  
*# Create your views here.  
  
# Create your views here.***def** AdminLoginCheck(request):  
 **if** request.method == **'POST'**:  
 usrid = request.POST.get(**'loginid'**)  
 pswd = request.POST.get(**'pswd'**)  
 print(**"User ID is = "**, usrid)  
 **if** usrid == **'admin' and** pswd == **'admin'**:  
 **return** render(request, **'admins/AdminHome.html'**)  
 **elif** usrid == **'Admin' and** pswd == **'Admin'**:  
 **return** render(request, **'admins/AdminHome.html'**)  
 **else**:  
 messages.success(request, **'Please Check Your Login Details'**)  
 **return** render(request, **'AdminLogin.html'**, {})  
  
  
**def** AdminHome(request):  
 **return** render(request, **'admins/AdminHome.html'**)  
  
  
**def** ViewAllUsers(request):  
 data = UserRegistrationModel.objects.all()  
 **return** render(request, **'admins/ViewallUsers.html'**, {**'data'**: data})  
  
  
**def** AdminActivaUsers(request):  
 **if** request.method == **'GET'**:  
 id = request.GET.get(**'uid'**)  
 status = **'activated'** print(**"PID = "**, id, status)  
 UserRegistrationModel.objects.filter(id=id).update(status=status)  
 data = UserRegistrationModel.objects.all()  
 **return** render(request, **'admins/ViewallUsers.html'**, {**'data'**: data})  
  
**def** AdminLinearResults(request):  
 data = UserLinearResultModel.objects.all()  
 **return** render(request, **'admins/adminLinearResults.html'**,{**'data'**:data})  
  
**def** AdminGA2Results(request):  
 data = UserGA2MResultModel.objects.all()  
 **return** render(request, **'admins/AdminGa2mResults.html'**,{**'data'**:data})

maternity models.py

**from** django.db **import** models  
  
*# Create your models here.***class** RetinopathyofPrematureModel(models.Model):  
 gestationalweek = models.CharField(max\_length=100)  
 mechanicalventilation = models.CharField(max\_length=100)  
 bloodtransfusion = models.CharField(max\_length=100)  
 gender = models.CharField(max\_length=100)  
 lateonsetsepsis = models.CharField(max\_length=100)  
 chorioamnionitis = models.CharField(max\_length=100)  
 pretermprematureruptureofmembranes = models.CharField(max\_length=100)  
 antenatalsteroidtherapy = models.CharField(max\_length=100)  
 respiratorydistresssyndrome = models.CharField(max\_length=100)  
 dopamindobutamin = models.CharField(max\_length=100)  
 necrotizingenterocolitis = models.CharField(max\_length=100)  
 intraventricularhemorrhage = models.CharField(max\_length=100)  
 constant = models.CharField(max\_length=100)  
 weight = models.CharField(max\_length=100)  
  
 **def** \_\_str\_\_(self):  
 **return** self.id  
 **class** Meta:  
 db\_table = **"ROPTable"**

maternity forms.py

**from** django **import** forms  
**from** .models **import** RetinopathyofPrematureModel  
  
  
**class** RetinopathyofPrematureForm(forms.ModelForm):  
 gestationalweek = forms.IntegerField()  
 mechanicalventilation = forms.IntegerField()  
 bloodtransfusion = forms.IntegerField()  
 gender = forms.IntegerField()  
 lateonsetsepsis = forms.IntegerField()  
 chorioamnionitis = forms.IntegerField()  
 pretermprematureruptureofmembranes = forms.IntegerField()  
 antenatalsteroidtherapy = forms.IntegerField()  
 respiratorydistresssyndrome = forms.IntegerField()  
 dopamindobutamin = forms.IntegerField()  
 necrotizingenterocolitis = forms.IntegerField()  
 intraventricularhemorrhage = forms.IntegerField()  
 constant = forms.IntegerField()  
 weight = forms.IntegerField()  
  
 **class** Meta:  
 model = RetinopathyofPrematureModel  
 fields = **'\_\_all\_\_'**

Maternity side views.py

from django.shortcuts import render

from django.contrib import messages

from .forms import RetinopathyofPrematureForm

from django\_pandas.io import read\_frame

from .models import RetinopathyofPrematureModel

from users.models import UserGA2MResultModel

# Create your views here.

def MaternityLoginCheck(request):

if request.method == 'POST':

usrid = request.POST.get('loginname')

pswd = request.POST.get('pswd')

print("User ID is = ", usrid)

if usrid == 'Maternity' and pswd == 'Maternity':

return render(request, 'maternity/MaternityHome.html')

elif usrid == 'maternity' and pswd == 'maternity':

return render(request, 'maternity/MaternityHome.html')

else:

messages.success(request, 'Please Check Your Login Details')

return render(request, 'MaternityLogin.html', {})

def MaternityHome(request):

return render(request, 'maternity/MaternityHome.html')

def MaternityAddDataForm(request):

form = RetinopathyofPrematureForm()

return render(request, 'maternity/AddMaternityNewbaby.html',{'form':form})

def MaternityAddDataAction(request):

if request.method == 'POST':

form = RetinopathyofPrematureForm(request.POST)

if form.is\_valid():

print('Data is Valid')

form.save()

messages.success(request, 'Data Stored Create one More')

form = RetinopathyofPrematureForm()

return render(request, 'maternity/AddMaternityNewbaby.html', {'form': form})

else:

messages.success(request, 'Invalid Entry')

print("Invalid form")

else:

form = RetinopathyofPrematureForm()

return render(request, 'maternity/AddMaternityNewbaby.html', {'form': form})

def MaternityGa2m(request):

data = RetinopathyofPrematureModel.objects.all()

df = read\_frame(data)

X = df.iloc[:, 1].values.astype(int)

y = df.iloc[:, 3].values.astype(int)

# Poisson regression code

import statsmodels.api as sm

exog, endog = sm.add\_constant(X), y

mod = sm.GLM(endog, exog,

family=sm.families.Poisson(link=sm.families.links.log))

res = mod.fit()

print(res.params)

scale = res.scale

deviance = res.deviance

pearson\_chi2 = res.pearson\_chi2

llf = res.llf

loginid = request.session['loginid']

print("Scale =", scale, " Devince = ", deviance, " pearsonc = ", pearson\_chi2, " llf = ", llf)

return render(request, "maternity/MaternityGa2mResult.html",

{'scale': scale, 'deviance': deviance, 'pearson\_chi2': pearson\_chi2, 'llf': llf})

def MaternityViewGA2mResults(request):

data = UserGA2MResultModel.objects.all()

return render(request,'maternity/ViewGa2mResults.html',{'data':data})

Base.html

{%load static%}

<!doctype html>

<html lang="en">

<head>

<title>RoP &mdash; Retinopathy of Premature</title>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

<link href="https://fonts.googleapis.com/css?family=DM+Sans:300,400,700|Indie+Flower" rel="stylesheet">

<link rel="stylesheet" href="{%static 'fonts/icomoon/style.css'%}">

<link rel="stylesheet" href="{%static 'css/bootstrap.min.css'%}">

<link rel="stylesheet" href="{%static 'css/bootstrap-datepicker.css'%}">

<link rel="stylesheet" href="{%static 'css/jquery.fancybox.min.css'%}">

<link rel="stylesheet" href="{%static 'css/owl.carousel.min.css'%}">

<link rel="stylesheet" href="{%static 'css/owl.theme.default.min.css'%}">

<link rel="stylesheet" href="{%static 'fonts/flaticon/font/flaticon.css'%}">

<link rel="stylesheet" href="{%static 'css/aos.css'%}">

<!-- MAIN CSS -->

<link rel="stylesheet" href="{%static 'css/style.css'%}">

</head>

<body data-spy="scroll" data-target=".site-navbar-target" data-offset="300">

<div class="site-wrap" id="home-section">

<div class="site-mobile-menu site-navbar-target">

<div class="site-mobile-menu-header">

<div class="site-mobile-menu-close mt-3">

<span class="icon-close2 js-menu-toggle"></span>

</div>

</div>

<div class="site-mobile-menu-body"></div>

</div>

<header class="site-navbar site-navbar-target" role="banner">

<div class="container mb-3">

<div class="d-flex align-items-center">

<div class="site-logo mr-auto">

<a href="{%url 'index'%}"><span class="text-primary">Retinopathy of Premature</span></a>

</div>

</div>

</div>

<div class="container">

<div class="menu-wrap d-flex align-items-center">

<span class="d-inline-block d-lg-none"><a href="#" class="text-black site-menu-toggle js-menu-toggle py-5"><span class="icon-menu h3 text-black"></span></a></span>

<nav class="site-navigation text-left mr-auto d-none d-lg-block" role="navigation">

<ul class="site-menu main-menu js-clone-nav mr-auto ">

<li><a href="{%url 'index'%}">Home</a></li>

<li><a href="{%url 'UserLogin'%}" class="nav-link">User</a></li>

<li><a href="{%url 'AdminLogin'%}" class="nav-link">Admins</a></li>

<li><a href="{%url 'MaternityLogin'%}" class="nav-link">Maternity</a></li>

<li><a href="{%url 'UserRegister'%}" class="nav-link">Registrations</a></li>

</ul>

</nav>

</div>

</div>

</header>

{%block contents%}

{%endblock%}

<div class="site-section bg-info">

<div class="container">

<div class="row mb-5">

<div class="col-12 text-center">

<span class="text-cursive h5 text-red d-block">Classification You Like</span>

<h2 class="text-white">Generalized Additive Model</h2>

</div>

</div>

<div class="row">

<div class="col-lg-4 mb-4 mb-lg-0">

<div class="package text-center bg-white">

<span class="img-wrap"><img src="{%static 'images/flaticon/svg/001-jigsaw.svg'%}" alt="Image" class="img-fluid"></span>

<h3 class="text-teal">World Health Organization</h3>

<p>WHO estimates that, out of the 130 million babies born annually, around 15 million babies are born prematurely, before 37 completed weeks of gestation. Approximately 1 million children die each year due to complications of preterm birth</p>

</div>

</div>

<div class="col-lg-4 mb-4 mb-lg-0">

<div class="package text-center bg-white">

<span class="img-wrap"><img src="{%static 'images/flaticon/svg/002-target.svg'%}" alt="Image" class="img-fluid"></span>

<h3 class="text-success">Retinopathy of Prematurity</h3>

<p>Retinopathy of Prematurity (RoP) was first reported by Terry in 1942 as a developmental, vascular, and proliferative retinal disorder occurring in the premature newborns’ retinas that have not completed vascularization</p>

</div>

</div>

<div class="col-lg-4 mb-4 mb-lg-0">

<div class="package text-center bg-white">

<span class="img-wrap"><img src="{%static 'images/flaticon/svg/003-mission.svg'%}" alt="Image" class="img-fluid"></span>

<h3 class="text-danger">Clinical data</h3>

<p>The clinical data for our analysis was collected by the Newborn Clinic meterenity hospitals. we will enter data as manual</p>

</div>

</div>

</div>

</div>

</div>

<footer class="site-footer">

<div class="container">

<div class="row pt-5 mt-5 text-center">

<div class="col-md-12">

<div class="border-top pt-5">

<p>

<!-- Link back to Colorlib can't be removed. Template is licensed under CC BY 3.0. -->

Copyright &copy;<script>document.write(new Date().getFullYear());</script> All rights reserved | This template is made with <i class="icon-heart text-danger" aria-hidden="true"></i> by <a href="#" target="\_blank" >Alex Corporation</a>

<!-- Link back to Colorlib can't be removed. Template is licensed under CC BY 3.0. -->

</p>

</div>

</div>

</div>

</div>

</footer>

</div>

<script src="{%static 'js/jquery-3.3.1.min.js'%}"></script>

<script src="{%static 'js/jquery-migrate-3.0.0.js'%}"></script>

<script src="{%static 'js/popper.min.js'%}"></script>

<script src="{%static 'js/bootstrap.min.js'%}"></script>

<script src="{%static 'js/owl.carousel.min.js'%}"></script>

<script src="{%static 'js/jquery.sticky.js'%}"></script>

<script src="{%static 'js/jquery.waypoints.min.js'%}"></script>

<script src="{%static 'js/jquery.animateNumber.min.js'%}"></script>

<script src="{%static 'js/jquery.fancybox.min.js'%}"></script>

<script src="{%static 'js/jquery.stellar.min.js'%}"></script>

<script src="{%static 'js/jquery.easing.1.3.js'%}"></script>

<script src="{%static 'js/bootstrap-datepicker.min.js'%}"></script>

<script src="{%static 'js/aos.js'%}"></script>

<script src="{%static 'js/main.js'%}"></script>

</body>

</html>