

APPOINTMENT MASTER

Project Group 20

Pareshkumar Sharma	185223693104
Raj Trambadiya	185223693117

Project Definition:

Appointment master is a system that provide better way for client & service provider to communicate. In this system we can manage appointments, services, payments, feedbacks etc. It is an interface between clients and service providers such as doctor and patient.

In any normal organization or department you can see there are lot of people come on same time, but at a time only one or limited people get their service, on that time other all peoples are waiting. The ultimate goal of this system is to reduce this waiting and wasting time of clients.

Requirement Determination & Analysis

This is an appointment management system so there will be strong requirement of time management of appointments. To make this system successful it needs to get as more and more service providers and services as possible.

To give the structure to this system the following will be main blocks of this system.

- **Users**
 - **DBA**
 - **Client**
 - **Service Provider (SP)**
- **Appointments**
 - **Client id**
 - **Details of Service**
 - **Time**
 - **Date**
 - **Place**
 - **Token Number**
- **Services**
 - **Created by SP**
 - **Time of Appointment**
 - **Dates are available**
 - **Place**
 - **Tags**
- **Feedbacks**
 - **Type Positive/Negative**
 - **Client description**
- **Payment**
 - **Amount**

This system has this main 3 types of users and has a main goal as to overcome the wasting and waiting time of clients to use service.

Appointments are best way to communicate between client and service provider. The communication will be like as one patient placed one appointment for a doctor and doctor see it, he will decide to whether he is free on that time or not, if he is free than he can approve the appointment of patient otherwise doctor can refuse that appointment and system will tell next available appointment time for that patient.

To manage system we need to add sign up, sign in and dashboard for client and service provider users.

As appointments, service provider and services are important part of this system. We need to make a decisions on their tasks as:

Clients:

Clients can place their request appointment by simply register them in system. Clients can explore all the services if they are not registered.

Service Provider:

The registration is 1st thing for SPs, after that they can add services to the system, but the service will be checked and authenticated first by our High level officers, if the service is genuine then it will be shown in the system. One SP can create maximum 5 services.

The SP can see the feedback from clients, based on feedbacks from client the service will be judged by DBA and our system also trying to add payment facility.

Admin:

Admin will manage the other users and other parts of the system, he have all super access to the system.

Project Modules:

- Home-Page:

In this home page user view all service that provide by service provider. User can sign-up for use feature of this system. User can also select any service that time sign-in is first, then after user can use services.

- Server Provider Dashboard:

In this module service provider manage the service such like add service, update service, delete service etc. service provider also manage clients & service provider view feedback was given by clients. They will get appointment request & also those request approve/ rejected by us.

- Client Dashboard:

In this module user can add appointment request to the service provider & also give feedback for the service. Client can also explore the services.

- Admin Dashboard:

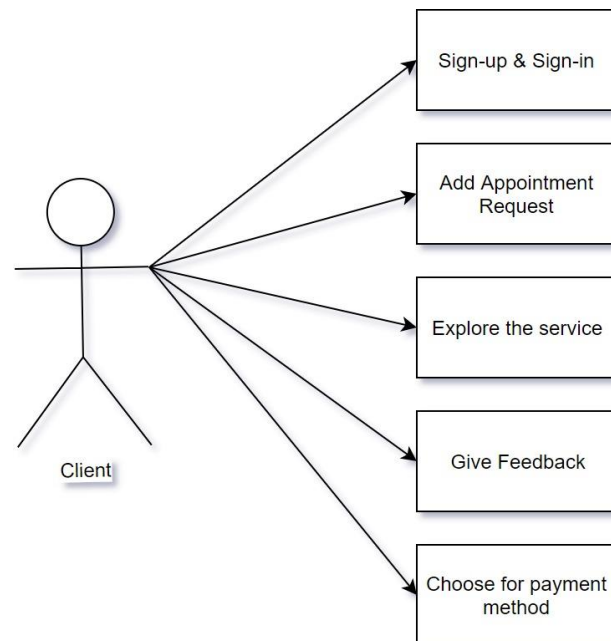
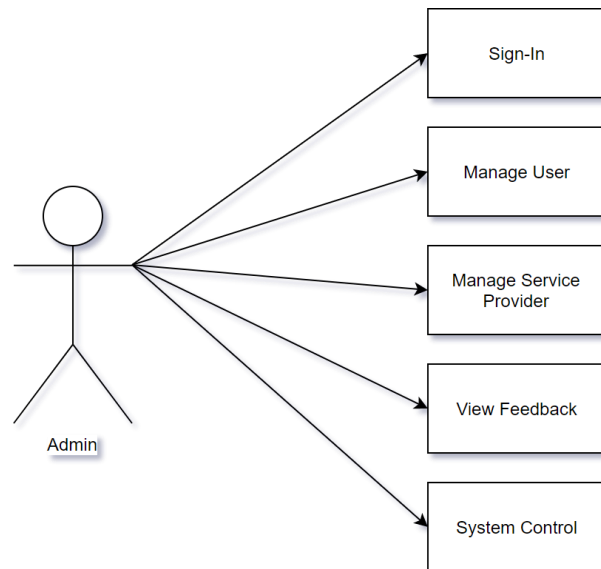
In this admin module admin can manage both clients and service provider. Admin can also view feedback who give by clients.

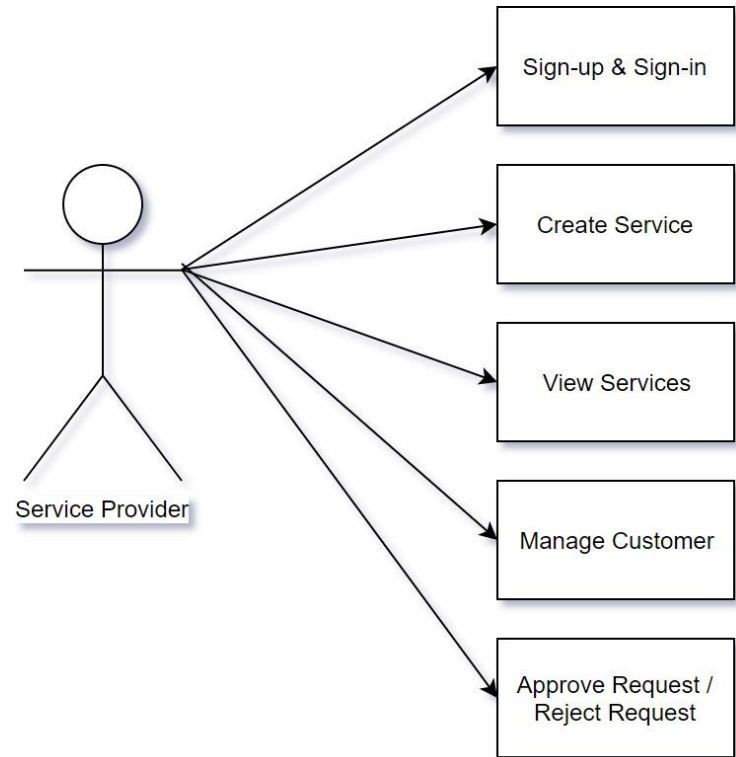
- Search Page:

In this module user will redirect to search page & get the results.

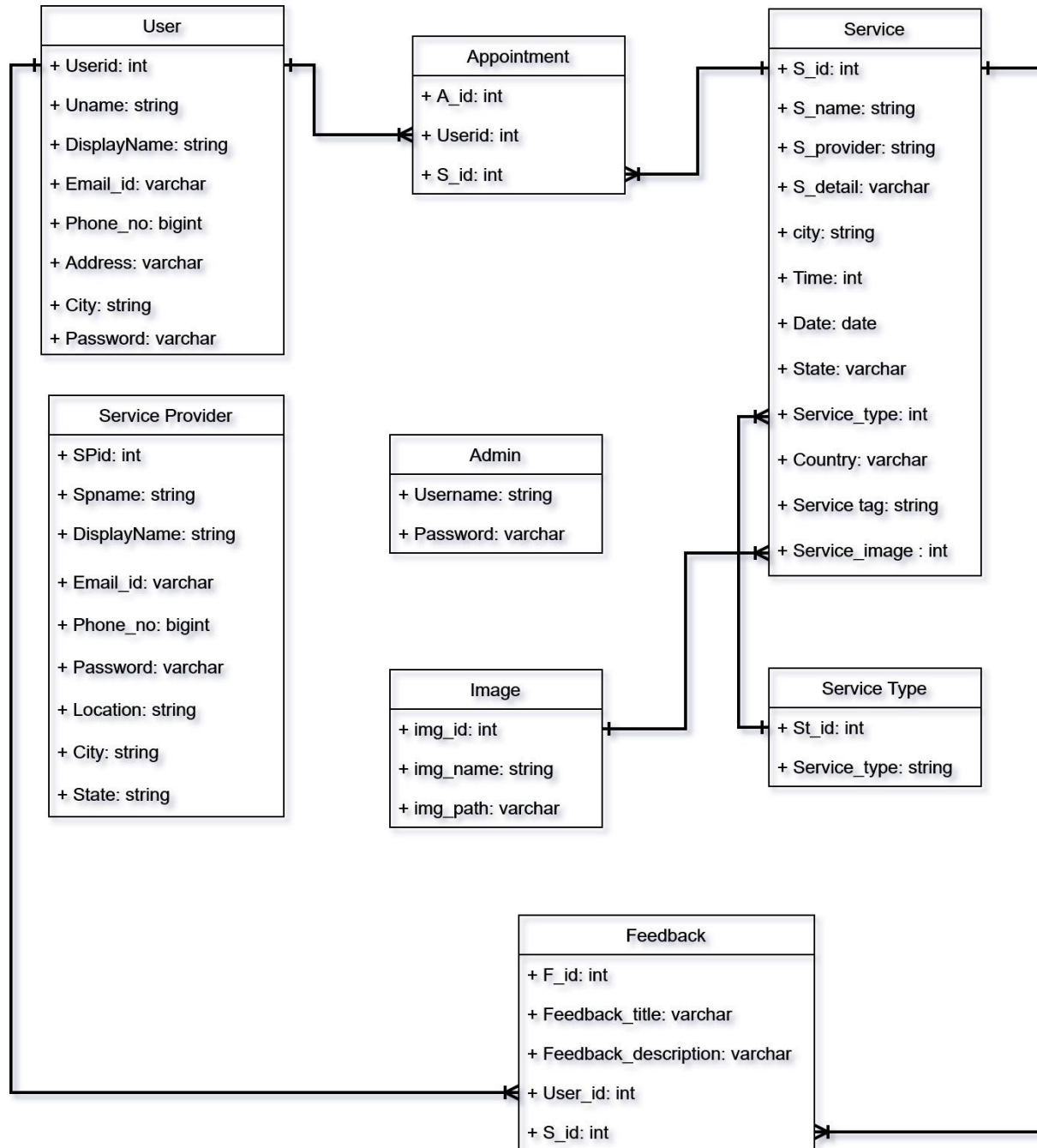
Project Diagrams

Use Cases:

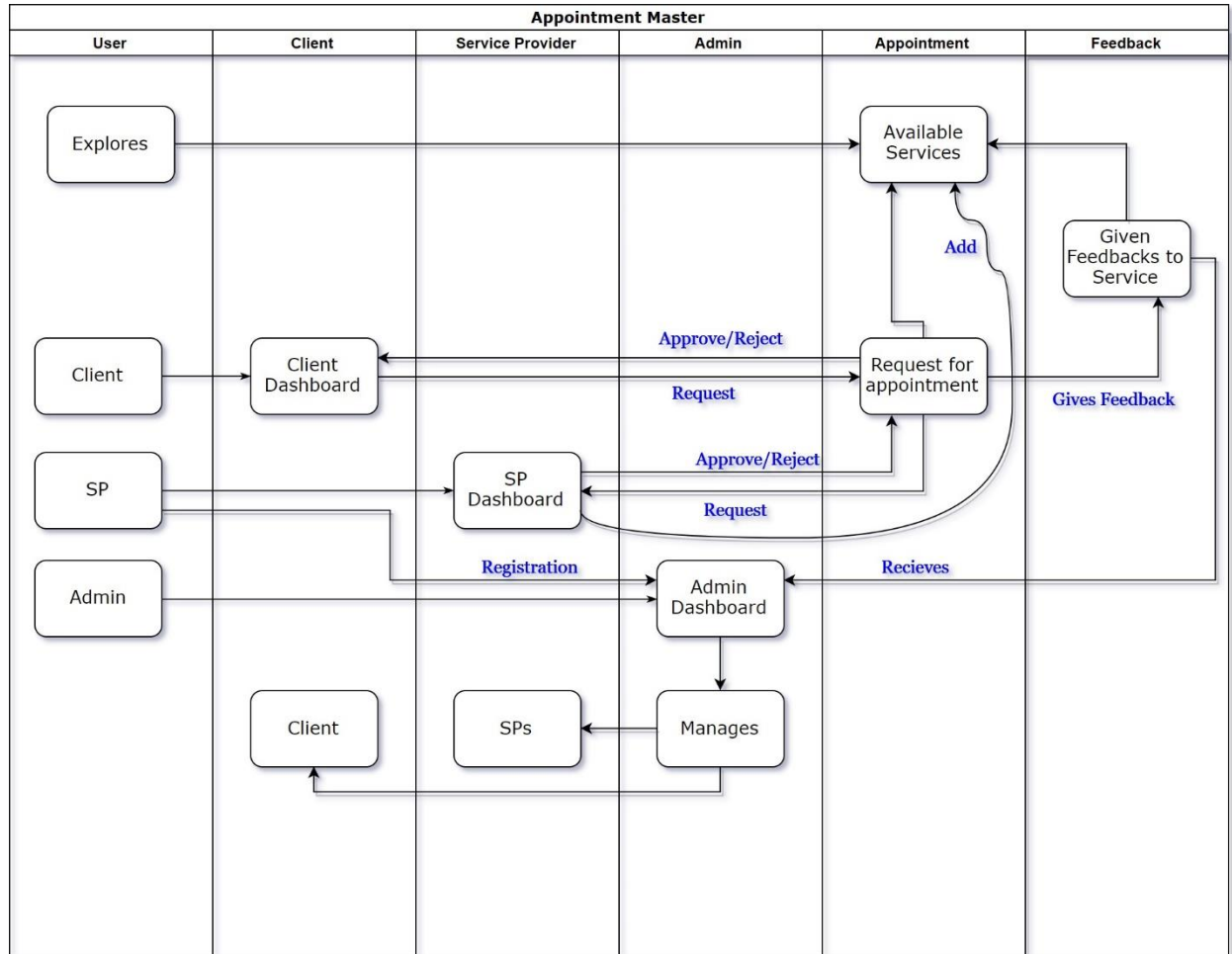




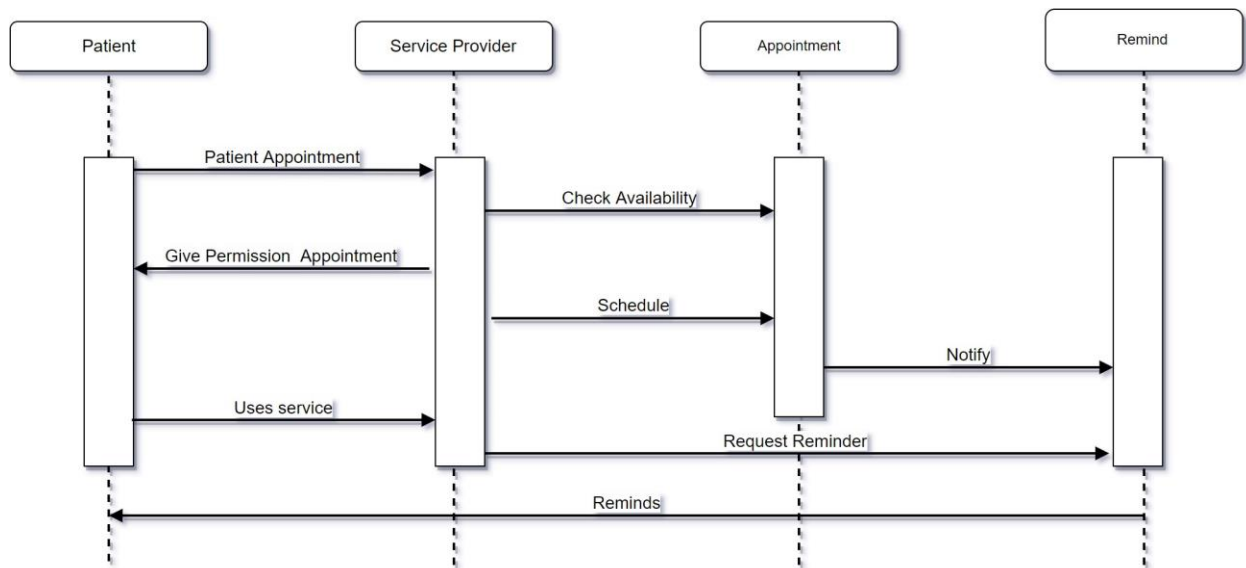
Class Diagram:



Activity Diagram



Interaction Diagram



Project Coding Phase 1

View.py

```
from django.shortcuts import render,redirect
from pages.models import client,appointment,sp,admin,service

issp = False
isadmin = False

def qry(str):
    import MySQLdb as sql
    global cur,con
    con = sql.connect(user='root',password='',database='AppointmentMaster')
    cur = con.cursor()
    return cur.execute(str),cur

def getusername(req):

    global issp,isadmin

    try:
        clientid = req.COOKIES['userid']
    except:
        clientid = None

    try:
        spid = req.COOKIES['spid']
    except:
        spid = None

    try:
        adminid = req.COOKIES['adminid']
    except:
        adminid = None

    if clientid != None and clientid != '' and clientid != 'None':
        row,c = qry("select name from pages_client where id =
{}".format(clientid))
        if row>0:
            issp = False
```

```

        isadmin = False
        return c.fetchone()[0]
    else:
        return None
elif spid != None and spid != '' and spid != 'None':

    issp = True
    isadmin = False
    row,c = qry("select name from pages_sp where id = {}".format(spid))

    if row>0:
        return c.fetchone()[0]
    else:
        return None

else:

    if adminid != None and adminid != '' and adminid != 'None':
        issp = False
        isadmin = True
        print('\n\n',"select username from pages_admin where id =
{}".format(adminid),'\n\n')
        row,c = qry("select username from pages_admin where id =
{}".format(adminid))
        if row>0:
            return c.fetchone()[0]
        else:
            return None
    else:
        return None

def usercheck_profiling(req,html,userneed=False):
    global issp,isadmin
    user = getusername(req)

    if user==None:
        if not userneed:
            return render(req,html,{'profile_visibility':
'none','register_visibility':'block','user_name':None,'issp':issp,'isadmin':isadm
in,'total_client':client.objects.count(),'total_sp':sp.objects.count(),'total_app
ointment':appointment.objects.count(),'total_service':service.objects.count()})
        else:
            return None
    else:

```

```

        return render(req,html,{'profile_visibility':
'block','register_visibility':'none','user_name':user,'issp':issp,'isadmin':isadm
in,'total_client':client.objects.count(),'total_sp':sp.objects.count(),'total_app
ointment':appointment.objects.count(),'total_service':service.objects.count()})
#
#
## Pages starts from here */
#

def logout(req):
    res = render(req,'home.html',{'profile_visibility':
'none','register_visibility':'block','user_name':None,'total_client':client.objec
ts.count(),'total_sp':sp.objects.count(),'total_appointment':appointment.objects.
count(),'total_service':service.objects.count()})
    try:
        res.set_cookie('userid','')
        res.set_cookie('spid','')
        res.set_cookie('adminid','')
    except Exception as e:
        print('\n\n',e)
    return res

def home(request):

    # print('objects----- client',client.objects.all())
    # print('objects-----service',service.objects.all())
    # print('objects-----admin',admin.objects.all())
    # print('objects-----sp',sp.objects.all())
    # print('objects-----appointment',appointment.objects.all())
    # print('Method type in homepage:',request.method)

    if request.method == 'GET':
        return usercheck_profiling(request,'home.html',False)
    else:
        return usercheck_profiling(request,'home.html',False)

def clientSignup(request):

    username = getusername(request)

    if request.method == 'GET':
        row,c = qry('select * from city')
        cities = c.fetchall()
        #print(cities)

```

```

        if username == None:
            return
    render(request, '.\\login\\usersignupform.html',{'profile_visibility':
'none','register_visibility':'block','city':cities,"user_name":None,'total_client':client.objects.count(),'total_sp':sp.objects.count(),'total_appointment':appointment.objects.count(),'total_service':service.objects.count()})
        else:
            print('\n\n username: ',username)
            return
    render(request, '.\\login\\usersignupform.html',{'profile_visibility':
'block','register_visibility':'none','city':cities,"user_name":username,'total_client':client.objects.count(),'total_sp':sp.objects.count(),'total_appointment':appointment.objects.count(),'total_service':service.objects.count()})
        else:
            c1 =
client(id=(client.objects.count()+1),name=request.POST['name'],city=request.POST['location'],email=request.POST['email_id'],phone_no=request.POST['mobile'],password=request.POST['password'],gender=request.POST['gender'])
            #print("\n\n insert: ",c1.save(),"\n\n")
            if c1.save()>0:
                response = redirect('/homepage/')
                response.set_cookie("userid",c1.id)
                return response
            return None

def signin(request):
    login_error = None
    username = getusername(request)
    if request.method == 'GET':
        if username == None:
            return render(request,r'.\\login\\userlogin.html',{'profile_visibility':
'none','register_visibility':'block',"user_name":None,'total_client':client.objects.count(),'total_sp':sp.objects.count(),'total_appointment':appointment.objects.count(),'total_service':service.objects.count()})
        else:
            print('\n\n username: ',username)
            return
    render(request,r'.\\login\\userlogin.html',{'profile_visibility':
'block','register_visibility':'none',"user_name":username,'total_client':client.objects.count(),'total_sp':sp.objects.count(),'total_appointment':appointment.objects.count(),'total_service':service.objects.count()})
        else:
            usr =
client.objects.filter(name=request.POST["uname"],password=request.POST['password'])

```

```

        username = None
        if len(usr)>0:
            username = usr[0].name
        else:
            login_error=' *username or password is wrong '
            username = None

        if username == None:
            res = render(request,r'login\userlogin.html',{'profile_visibility':
'none','register_visibility':'block',"user_name":None,"login_error":login_error})
            res.set_cookie('userid','')
            return res
        else:
            print('\n\n username: ',username,'logged in')
            res = redirect('/homepage/')
            res.set_cookie('userid',usr[0].id)
            return res

def SPSignup(request):
    username = getusername(request)

    if request.method == 'GET':
        row,c = qry('select * from city')
        cities = c.fetchall()
        #print(cities)

        if username == None:
            return
render(request,'.\login\serviceproviderssignupform.html',{'profile_visibility':
'none','register_visibility':'block','city':cities,"user_name":None,'total_client
':client.objects.count(),'total_sp':sp.objects.count(),'total_appointment':appoin
tment.objects.count(),'total_service':service.objects.count()})
        else:
            print('\n\n username: ',username)
            return
render(request,'.\login\serviceproviderssignupform.html',{'profile_visibility':
'block','register_visibility':'none','city':cities,"user_name":username,'total_cl
ient':client.objects.count(),'total_sp':sp.objects.count(),'total_appointment':ap
pointment.objects.count(),'total_service':service.objects.count()})
        else:
            sp1 =
sp(id=(sp.objects.count()+1),name=request.POST['name'],city=request.POST['locatio
n'],email=request.POST['email_id'],phone_no=request.POST['phone_no'],password=req
uest.POST['password'])
            #print("\n\n insert: ",c1.save(),"\n\n")

```

```

        if sp1.save()>0:
            response = redirect('/homepage/')
            response.set_cookie('userid','')
            response.set_cookie("spid",sp1.id)
            return response
        return None

def SPLogin(request):
    print('\n\n sp objects -----',sp.objects.all(),'\n\n')
    login_error = None
    username = getusername(request)
    if request.method == 'GET':
        if username == None:
            return
    render(request, '.\\login\\serviceproviderlogin.html',{'profile_visibility':
'none','register_visibility':'block',"user_name":None,'total_client':client.objects.
count(),'total_sp':sp.objects.count(),'total_appointment':appointment.objects.
count(),'total_service':service.objects.count()})
    else:
        print('\n\n username: ',username)
        return
    render(request, '.\\login\\serviceproviderlogin.html',{'profile_visibility':
'block','register_visibility':'none',"user_name":username,'total_client':client.o
bjects.count(),'total_sp':sp.objects.count(),'total_appointment':appointment.obje
cts.count(),'total_service':service.objects.count()})
    else:
        usr =
sp.objects.filter(name=request.POST["uname"],password=request.POST['password'])
        username = None
        if len(usr)>0:
            username = usr[0].name
        else:
            login_error=' *username or password is wrong '
            username = None

        if username == None:
            res =
render(request, '.\\login\\serviceproviderlogin.html',{'profile_visibility':
'none','register_visibility':'block',"user_name":None,"login_error":login_error,'
total_client':client.objects.count(),'total_sp':sp.objects.count(),'total_appoint
ment':appointment.objects.count(),'total_service':service.objects.count()})
            res.set_cookie('spid','')
            res.set_cookie('userid','')
            res.set_cookie('adminid','')
            return res

```



```

        else:
            print('\n\n username: ',username,'logged in')
            res = redirect('/homepage/')
            res.set_cookie('spid',usr[0].id)
            res.set_cookie('userid','')
            res.set_cookie('adminid','')
            return res

def adminLogin(request):
    login_error = None
    htmlurl = r'login\adminlogin.html'
    username = getusername(request)
    if request.method == 'GET':
        if username == None:
            return render(request,htmlurl,{ 'profile_visibility':
'none','register_visibility':'block',"user_name":None,'total_client':client.objects.count(),
'total_sp':sp.objects.count(), 'total_appointment':appointment.objects.count(),
'total_service':service.objects.count()})
        else:
            print('\n\n username: ',username)
            return render(request,htmlurl,{ 'profile_visibility':
'block','register_visibility':'none',"user_name":username,'total_client':client.objects.count(),
'total_sp':sp.objects.count(), 'total_appointment':appointment.objects.count(),
'total_service':service.objects.count()})
        else:
            usr =
admin.objects.filter(username=request.POST["name"],password=request.POST['password'])

            username = None
            if len(usr)>0:
                username = usr[0].username
            else:
                login_error = ' *username or password is wrong '
                username = None

            if username == None:
                res = render(request,htmlurl,{ 'profile_visibility':
'none','register_visibility':'block',"user_name":None,"login_error":login_error})
                res.set_cookie('adminid','')
                return res
            else:
                print('\n\n username: ',username,'logged in')
                res = redirect('/homepage/')
                res.set_cookie('adminid',usr[0].id)
                return res

```

```
def about(request):
    return render(request, 'about.html', {})

def contact(request):
    return render(request, 'contact.html', {})

def clientdashboard(request):
    if request.method == 'GET':
        return usercheck_profiling(request, 'userdashboard.html')
    else:
        return usercheck_profiling(request, 'userdashboard.html')

def spdashboard(request):
    if request.method == 'GET':
        return usercheck_profiling(request, 'spdashboard.html')
    else:
        return usercheck_profiling(request, 'spdashboard.html')

def appointments(request):
    return render(request, 'appointment.html', {})

def notification(request):
    return render(request, 'notification.html', {})

def admindashboard(request):
    if request.method == 'GET':
        return usercheck_profiling(request, 'admindashboard.html')
    else:
        return usercheck_profiling(request, 'admindashboard.html')

def clientprofile(request):
    pass

def spprofile(request):
    pass

def newservice(request):
    if request.method == 'GET':
        usr = usercheck_profiling(request, 'newservice.html', True)
        if usr != None:
            return usr
```

```
        else:
            return redirect('/homepage/')
    else:
        pass

def manageservice(request):
    pass
```

Models.py

```
from django.db import models

class client(models.Model):
    id = models.IntegerField(primary_key=True)
    name = models.CharField(max_length=30)
    city = models.CharField(max_length=30)
    email = models.CharField(max_length=30)
    phone_no = models.CharField(max_length=30)
    password = models.CharField(max_length=30)
    gender = models.CharField(max_length=4)

    def save(self):
        import MySQLdb as s

        con = s.connect(user = 'root',password='',database='appointmentmaster')
        cur = con.cursor()
        query = 'insert into pages_client
(id,name,city,email,phone_no,password,gender) values
({}, "{}", "{}", "{}", "{}", "{}", "{}')'.format(self.id,self.name,self.city,self.email
,self.phone_no,self.password,self.gender)
        print("\n\n",self.name,"\n\n")
        out = cur.execute(query)
        if out > 0:
            con.commit()
        return out

class admin(models.Model):
    id = models.IntegerField(primary_key=True)
    username = models.CharField(max_length=30)
    password = models.CharField(max_length=30)

class sp(models.Model):
    id = models.IntegerField(primary_key=True)
    name = models.CharField(max_length=30)
    email = models.CharField(max_length=30)
    city = models.CharField(max_length=30)
    phone_no = models.CharField(max_length=30)
    password = models.CharField(max_length=30)
    status = models.CharField(max_length=10)

    def save(self):
```

```

import MySQLdb as s

con = s.connect(user = 'root',password='',database='appointmentmaster')
cur = con.cursor()
query = 'insert into pages_sp (id,name,city,email,phone_no,password)
values
({}, "{}", "{}", "{}", "{}", "{}")'.format(self.id,self.name,self.city,self.email,self
.phone_no,self.password)
print("\n\n",self.name,"\n\n")
out = cur.execute(query)
if out > 0:
    con.commit()
return out

class service(models.Model):
    id = models.IntegerField(primary_key=True,)
    providerid = models.IntegerField()
    title = models.CharField(max_length=30)
    time = models.CharField(max_length=30)
    city = models.CharField(max_length=30)
    location = models.CharField(max_length=30)
    day = models.CharField(max_length=30)
    description = models.CharField(max_length=30)
    image = models.CharField(max_length=30)

class appointment(models.Model):
    id = models.IntegerField(primary_key=True)
    userid = models.IntegerField()
    serviceid = models.IntegerField()
    city = models.CharField(max_length=30)
    description = models.CharField(max_length=30)
    status = models.CharField(max_length=10)
    date = models.CharField(max_length=10)
    time = models.CharField(max_length=10)
    location = models.CharField(max_length=30)

```

Urls.py

```
from django.urls import path
from . import views

urlpatterns = [
    path('', views.home, name="home"),
    path('homepage/', views.home, name="home"),
    path('about/', views.about, name="about"),
    path('contact/', views.contact, name='contact'),
    path('client/signup/', views.clientSignup, name='clientSignup'),
    path('client/signin/', views.signin, name='signin'),
    path('sp/signup/', views.SPSignup, name='spsignup'),
    path('sp/login/', views.SPLogin, name='splogin'),
    path('admin_login/', views.adminLogin, name='adminLogin'),
    path('admin_login/dashboard', views.admindashboard, name='adminbdb'),
    path('homepage/logout/', views.logout, name='lot'),
    path('client/dashboard/', views.clientdashboard, name='clientbdb'),
    path('client/profile/', views.clientprofile, name='clientprf'),
    path('sp/dashboard/', views.spdashboard, name='spbdb'),
    path('sp/profile/', views.spprofile, name='spprf'),
    path('client/appointment/', views.appointments, name='appointment'),
    path('client/notification/', views.notification, name='notification'),
    path('sp/newservice/', views.newservice, name='newservice'),
    path('sp/manageservice/', views.manageservice, name='manageservice'),
]
```

Settings.py

```
"""
Django settings for APMT project.

Generated by 'django-admin startproject' using Django 2.2.3.

For more information on this file, see
https://docs.djangoproject.com/en/2.2/topics/settings/

For the full list of settings and their values, see
https://docs.djangoproject.com/en/2.2/ref/settings/
"""

import os

# Build paths inside the project like this: os.path.join(BASE_DIR, ...)
BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))

# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/2.2/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = '^vfag$sx_e&d_g!k!y(2xb=h+9)gw%z^+8pyq538p(u222lk@7'

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = ['localhost']

# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'pages',
]
```

```
MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'APMT.urls'

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [],
        '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',
            ],
        },
    ],
]

WSGI_APPLICATION = 'APMT.wsgi.application'

# Database
# https://docs.djangoproject.com/en/2.2/ref/settings/#databases

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        #'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
        'NAME': 'AppointmentMaster',
        'USER': 'root',
        'PASSWORD': '',
    }
}
```



```
# Password validation
# https://docs.djangoproject.com/en/2.2/ref/settings/#auth-password-validators

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME':
'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]

# Internationalization
# https://docs.djangoproject.com/en/2.2/topics/i18n/

LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'UTC'

USE_I18N = True

USE_L10N = True

USE_TZ = True

# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/2.2/howto/static-files/

STATIC_URL = '/static/'
STATIC_ROOT = os.path.join(BASE_DIR, 'static')
```

APMT – Urls.py

```
from django.contrib import admin
from django.urls import path,include

urlpatterns = [
    path('admin/', admin.site.urls),
    path('',include('pages.urls')),
]
```

Base.html

```
{% load static %}

<html>

    <head>

        <link rel='stylesheet' type='text/css' href="{% static 'pages/header.css'
%}" />
        <link rel='stylesheet' type='text/css' href="{% static 'pages/footer.css'
%}" />

        <script>

            function search(){

                document.getElementById('rr').href='search.html?searchtext='+docu
ment.getElementById('searchtext').value;
                return 0;
            }

            function logout(){
                document.getElementById('signed').style.display= 'none';
                document.getElementById('unsigned').style.display= 'block';
            }

            function load(){
                document.getElementById('searchtext').focus();
            }
        </script>

    </head>

    <body onload="load();">

        <header>

            <div class='containerheader'>

                <div class="title" style="width: 30%; ">
                    <h1 style="margin-left:30px; float:left; min-width: 290px"><a
href="/homepage" style="text-decoration: none; "> Appointment Master </a> </h1>
```

```

        </div>

        <div class="profile" style="float:right; width: 100px; width:
12%; min-width: 150px;">

            <span id="signed" style='display:{{ profile_visibility }}'>

                <h5 style="width:50%; text-align:center; margin:0;
padding:0;">

                    <img width="50px" height="50px" src='{% static
"/images/user.png" %}' >

                        <span style="display:block; padding: 0px 5px 5px 5px;
border-bottom-left-radius:5px; border-bottom-right-radius:5px; font-weight:bold;
font-size: 16px;">{{ user_name }}</span>
                    </h5>

                    <div class="pmenu">
                    <div id="profileicon"></div>
                    <div id="menus">

                        <ul>

                            {% if issp %}

                                <li><a href="{% url 'spprf' %}">Profile</a></li>
                                <li><a href="{% url 'spdbd' %}">
Dashboard</a></li>
                                <li><a href="{% url 'lot' %}"
onclick="logout();">Log Out</a></li>

                                {% elif isadmin %}
                                <li><a href="{% url 'admindbd' %}">
Dashboard</a></li>
                                <li><a href="{% url 'lot' %}"
onclick="logout();">Log Out</a></li>

                                {% else %}

                                    <li><a href="{% url 'clientprf'
%}">Profile</a></li>
                                    <li><a href="{% url 'clientdbd' %}">
Dashboard</a></li>
                                    <li><a href="{% url 'appointment'
%}">Appointment</a></li>

```

```

        <li><a href="{% url 'notification'
{%}">Notification</a></li>
        <li><a href="{% url 'lot' %}"
onclick="logout();">Log Out</a></li>

        {% endif %}

    </ul>
</div>
</div>
</span>

<span id="unsigned" style="display:{% register_visibility %}">

    <h3> <a href="{% url 'signin' %}"> Sign In </a> | <a href="{%
url 'clientSignup' %}" > Sign Up </a> </h3>

</span>
</div>

<div class="search" style="float:left; width: 40%; padding-left:
16%;">

    <form method="get" action="search.html">
        <input type="text" placeholder="search doctor |
appointment" id="searchtext" name="searchtext" onkeydown="search();" style="font-
size: 20px; width: 70%; padding: 5px; padding-left:9px; border-radius: 10px;
padding-right: 40px;"/>
        <a id="rr" style="text-decoration:none; " href=""><span
id="redirect">Search</span> </a>
    </form>
</div>
</div>

</header>

<div class="navigation" >
    <nav>
        <ul>
            <li><a href="{% url 'home' %}">Homepage</a> </li>
            <li style="color: black; font-weight: normal;">Take
Appointment For :-</li>
            <li><a href="">Doctors</a> | </li>

```

```

        <li><a href="">Business</a> | </li>
        <li><a href="">Celebrities</a> | </li>
        <li>
            <span><a id="abcd" href="#"
onblur="document.getElementById('.diffservices').style.display='none';">Servi
ces</a>

                <div class="diffservices">
                    <div class="sub-menu">
                        <div style="float:left;">
                            <ul>
                                <li><a href="">Service 1</a></li>
                                <li><a href="">Service 1</a></li>
                                <li><a href="">Service 1</a></li>
                                <li><a href="">Service 1</a></li>
                            </ul>
                        </div>

                    </div>
                </div>
            </span>
            |
        </li>

        <li><a href="">Join Us</a></li>
        <li style="float:right; margin-top: 8px;"><a href="">About
Us</a></li>
        <li style="float:right; margin-top: 8px;"><a href="">Contact
Us</a></li>
    </ul>

</nav>

</div>

<div class='otherpages'>

    {% block otherpage %}

    {% endblock %}

    {% block otherpage1 %}

    {% endblock %}

</div>

```

```

<footer>
  <p id="title"> Appointment Master </p>
  <div>

    <div style="float:left; overflow: auto; ">
      <ul>
        <li><a href="{% url 'home' %}">Homepage</a></li>
        <li><a href="">Contact Us</a></li>
        <li><a href="">About Us</a></li>
        <li><a href="">Services</a></li>
      </ul>
    </div>
    <div style="float:left;">
      <ul>
        <li><a href="{% url 'signin' %}">Client Log In</a></li>
        <li><a href="{% url 'clientSignup' %}">Client Sign
Up</a></li>

      </ul>
    </div>
    <div style="float:left;">
      <ul>
        <li><a href="{% url 'splogin' %}">Service Provider Log
In</a></li>
        <li><a href="{% url 'spsignup' %}">Service Provider Sign
Up</a></li>

      </ul>
    </div>

    <div style="float:left;">
      <ul>

        <li>Appointments : - {{ total_appointment }}</li>
        <li>Services : - {{ total_service }}</li>
        <li>Clients : - {{ total_client }}</li>
        <li>Services Providers : - {{ total_sp }}</li>

      </ul>
    </div>
  </div>

  <div style="float:right;">
    <ul>

```

```

                <li><a href="{% url 'adminLogin' %}">Admin Log
In</a></li>

            </ul>
        </div>
    </div>

    <p> Raj Trambadiya & Pareshkumar &copy; All Right Reserved</p>

</footer>

</body>

</html>

```

AdminLogin.html

```
{% extends 'base.html' %}
{% load static %}
{% block otherpage %}
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>

body {font-family: Arial, Helvetica, sans-serif;}
form.admin {border: 3px solid #f1f1f1;}

input[type=text], input[type=password] {
    width: 100%;
    padding: 12px 20px;
    margin: 8px 0;
    display: inline-block;
    border: 1px solid #ccc;
    box-sizing: border-box;
}

button {
    background-color: #4CAF50;
    color: white;
    padding: 14px 20px;
    margin: 8px 0;
    border: none;
    cursor: pointer;
    width: 100%;
}

button:hover {
    opacity: 0.8;
}

.cancelbtn {
    width: auto;
    padding: 10px 18px;
    background-color: #f44336;
}

.imgcontainer {
    text-align: center;
    margin: 24px 0 12px 0;
```

```

}

img.avatar {
  width: 40%;
  border-radius: 50%;
}

.container {
  padding: 16px;
}

span.psw {
  float: right;
  padding-top: 16px;
}

/* Change styles for span and cancel button on extra small screens */
@media screen and (max-width: 300px) {
  span.psw {
    display: block;
    float: none;
  }
  .cancelbtn {
    width: 100%;
  }
}
</style>

<h2>Admin Login Form</h2>

<form action="{% url 'adminLogin' %}" method="post" class='admin'>
  {% csrf_token %}
  <div class="imgcontainer">
    
  </div>

  <div class="container">
    <label for="uname"><b>Username:</b></label>
    <input type="text" placeholder="Enter Username" name="name" required>

    <label for="psw"><b>Password:</b></label>
    <input type="password" placeholder="Enter Password" name="password" required>

```

```
<button type="submit">Login</button>

</div>

<div class="container" style="background-color:#f1f1f1">
  <button type="button" class="cancelbtn">Cancel</button>
  <span class="psw">Forgot <a href="#">password?</a></span>
</div>
</form>

{% endblock %}
```