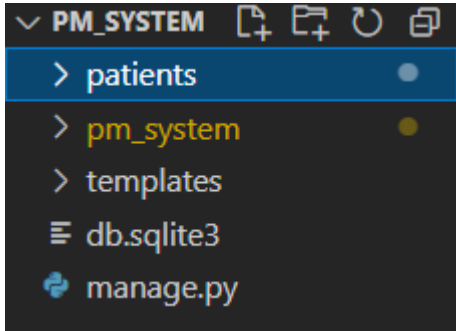


Patient Management System

Directory Structure:



- templates: HTML files for the project
- patients: Contain models, forms and views for patient

Note : Unique ID is generated using uuid package. It can be modified if needed.

Output:

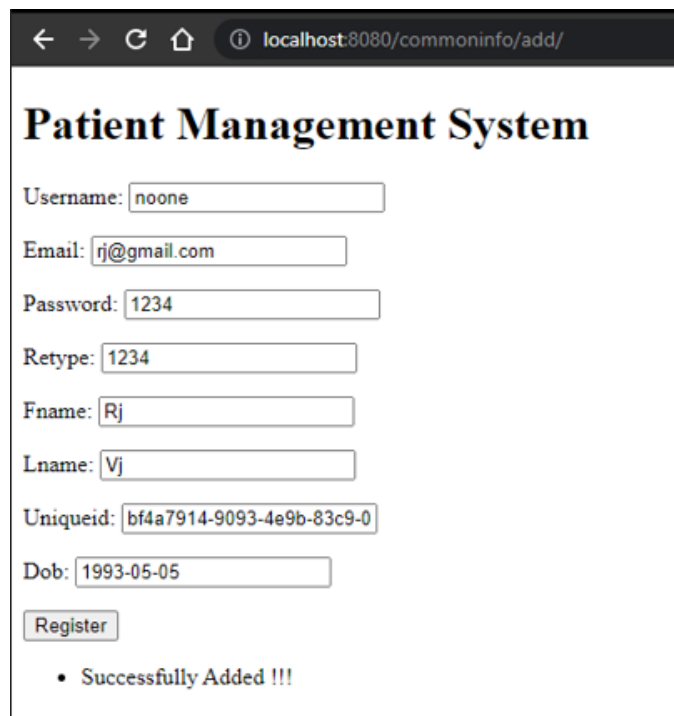
1. /commoninfo/add/

A screenshot of a web browser showing the registration page of the Patient Management System. The browser address bar shows 'localhost:8080/commoninfo/add/'. The page title is 'Patient Management System'. The form includes fields for Username, Email, Password, Retype, Fname, Lname, Uniqueid (pre-filled with '62213a73-2f68-45f1-b7dc-2f'), and Dob (pre-filled with 'yyyy-mm-dd'). A 'Register' button is at the bottom.

Home Page for Registration. Enter details

A screenshot of the same registration page, but with input data. The browser address bar shows 'localhost:8080/commoninfo/add/'. The form fields are filled with: Username: 'noone', Email: 'rj@gm', Password: '1234', Retype: '1234', Fname: 'Rj', Lname: 'Vj', Uniqueid: 'bf4a7914-9093-4e9b-83c9-0', and Dob: '1993'. There are two error messages: 'Enter a valid email address.' and 'Enter a valid date.' The 'Register' button is at the bottom.

Registration of patient with inbuilt pattern check. Once filled, click 'Register' button



A screenshot of a web browser showing the 'Patient Management System' registration page. The browser's address bar displays 'localhost:8080/commoninfo/add/'. The page has a title 'Patient Management System' in a large, bold, black serif font. Below the title, there are several input fields: 'Username' with 'noone', 'Email' with 'rj@gmail.com', 'Password' with '1234', 'Retype' with '1234', 'Fname' with 'Rj', 'Lname' with 'Vj', 'Uniqueid' with 'bf4a7914-9093-4e9b-83c9-0', and 'Dob' with '1993-05-05'. Each field is preceded by its label. At the bottom of the form is a 'Register' button. Below the button, a green message '• Successfully Added !!!' is displayed.

← → ↻ 🏠 ⓘ localhost:8080/commoninfo/add/

Patient Management System

Username:

Email:

Password:

Retype:

Fname:

Lname:

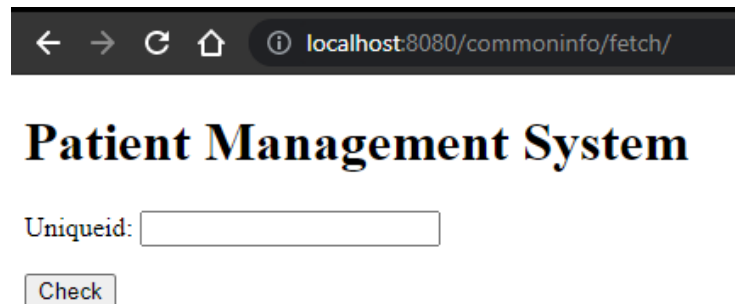
Uniqueid:

Dob:

• Successfully Added !!!

Successful Registration

2. /commoninfo/fetch/



A screenshot of a web browser showing the 'Patient Management System' fetch page. The browser's address bar displays 'localhost:8080/commoninfo/fetch/'. The page has a title 'Patient Management System' in a large, bold, black serif font. Below the title, there is a single input field labeled 'Uniqueid:'. Below the input field is a 'Check' button.

← → ↻ 🏠 ⓘ localhost:8080/commoninfo/fetch/

Patient Management System

Uniqueid:

Home Page to check the patient info. Enter patient Unique ID



A screenshot of the 'Patient Management System' fetch page, similar to the one above, but with the 'Uniqueid' input field containing the value 'bf4a7914-9093-4e9b-83c9-0'.

Patient Management System

Uniqueid:

Enter unique id of the patient and press 'Check' button

Patient Management System

Uniqueid:

User Info

User Name : Rj Vj

Unique ID : bf4a7914-9093-4e9b-83c9-0fa0ee19f629

Date of Birth : May 5, 1993

Result Obtained

Important Codes:

1. `pm_system/settings.py` (Major code snippet)

```
# Application definition

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

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 = 'pm_system.urls'

TEMPLATES = [
    {
```

```

        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [os.path.join(BASE_DIR, 'templates')],
        '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 = 'pm_system.wsgi.application'

# Database
# https://docs.djangoproject.com/en/3.2/ref/settings/#databases

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': BASE_DIR / 'db.sqlite3',
    }
}

```

2. pm_system/urls.py

```

from django.contrib import admin
from django.urls import path
from patients.views import patient_view_form, patient_check_form

urlpatterns = [
    path('admin/', admin.site.urls),
    path('commoninfo/fetch/', patient_check_form, name='commoninfo/fetch/'),
    path('commoninfo/add/', patient_view_form, name='commoninfo/add/'),
]

```

3. patients/admin.py

```

from django.contrib import admin
from .models import Patient

# Register your models here.
admin.site.register(Patient)

```

4. patients/forms.py

```

from django import forms
from django.forms import fields
from django.forms import widgets

```

```

from django.forms.widgets import TextInput
from .models import Patient, PatientCheck

class PatientForm(forms.ModelForm):

    class Meta:
        model = Patient
        fields = [
            'username',
            'email',
            'password',
            'retype',
            'fname',
            'lname',
            'uniqueid',
            'dob'
        ]

        widgets= {
            'dob': forms.TextInput(attrs = {'placeholder' : 'yyyy-mm-dd'}),
        }

class PatientIDForm(forms.ModelForm):
    class Meta:
        model = PatientCheck
        fields = {
            'uniqueid'
        }

```

5. patients/models.py

```

from django.db import models
import uuid

# Create your models here.

class Patient(models.Model):
    uniqueid = models.UUIDField(primary_key=True ,editable=True)
    username = models.CharField(max_length=50)
    email = models.EmailField()
    password = models.CharField(max_length=50)
    retype = models.CharField(max_length=50,null=True)
    fname = models.CharField(max_length=50)
    lname = models.CharField(max_length=50)
    dob = models.DateField(null=True)

class PatientCheck(models.Model):
    uniqueid = models.CharField(max_length=150)
    username = models.CharField(max_length=50)
    dob = models.DateField(null=True)

```

6. patients/views.py

```
from django.shortcuts import render
from .forms import PatientForm, PatientIDForm
from .models import Patient, PatientCheck
from django.contrib import messages

import uuid

# Create your views here.

def patient_view_form(request):
    form = PatientForm(request.POST or None, initial={'uniqueid' : uuid.uuid4})
    if form.is_valid():
        form.save()
        messages.success(request, 'Successfully Added !!!')

    context = {'form':form}

    return render(request,"patient_details.html",context)
## patient_details.html: html file for /commoninfo/add/

def patient_check_form(request):
    form = PatientIDForm(request.POST or None)
    context = {'form':form}
    flag = False
    if form.is_valid():
        # form.save()
        key = form.cleaned_data['uniqueid']

        try:
            patient = Patient.objects.get(uniqueid = key)
            fname = patient.fname
            lname = patient.lname
            uniqid = key
            dob = patient.dob
            context['username'] = fname + ' ' + lname
            context['uniqueid'] = uniqid
            context['dob'] = dob
            context['fform'] = fform

            flag = True
            context['flag'] = flag
        except:
            print('User not in the list')

    return render(request,"final.html",context)
## final.html: html file for /commoninfo/fetch/
```