Searching Hotel Room Project in Django

Last Updated : 18 Mar, 2024

In this article, we'll explore a technique for efficiently organizing and displaying images or information based on specific parameters. Our goal is to make it easy for users to sort images according to their preferences, particularly by price, with a single click. This project simplifies the process of arranging images or data in both ascending and descending order, offering a seamless user experience. By implementing a dynamic sorting feature, users can instantly arrange the images or information either in decreasing order (highest to lowest price) or increasing order (lowest to highest price) with a single click. This dynamic functionality enhances the user interface and improves the overall usability of your project. we understand this by creating one app Hotel search app so let's start

Hotel Search App in Django

First, create the virtual environment by running the following command :

```
virtualenv env
```

and then enter in virtual env

```
.\env\Scripts\activate.ps1
```

and then install Django and create the **Django** project by using the **Django** command

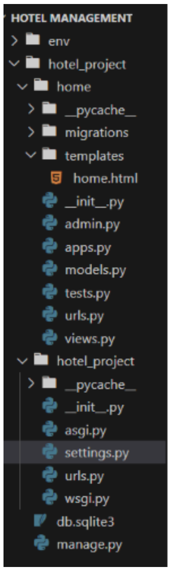
```
pip install django
django-admin startproject hotel_search
```

```
cd hotel_search
```

then we create the app

```
python manage.py startapp home
```

File Structure



```
HOTEL MANAGEMENT
> env
v hotel_project
  v home
    > __pycache__
    > migrations
    v templates
        home.html
      __init__.py
      admin.py
      apps.py
      models.py
      tests.py
      urls.py
      views.py
  v hotel_project
    > __pycache__
      __init__.py
      asgi.py
      settings.py
      urls.py
      wsgi.py
    db.sqlite3
    manage.py
```

File Structure

Then we need to register app in setttings.

```
INSTALLED_APPS = [
    "django.contrib.admin",
    "django.contrib.auth",
    "django.contrib.contenttypes",
    "django.contrib.sessions",
    "django.contrib.messages",
    "django.contrib.staticfiles",
    "home",
]
```

model.py: This code defines a Django model named 'Hotel' with fields for the hotel's name, price, description, and timestamps for creation and updates. The __str__ method returns the hotel's name for a human-readable representation.

Python3

```
from django.db import models
```

```python
class GFG(models.Model):
    hotel_name = models.CharField(max_length=100)
    hotel_price = models.IntegerField()
    hotel_description = models.TextField()
    created_at = models.DateField(auto_now_add =True)
    updated_at = models.DateField(auto_now  =True)

    def __str__(self) -> str:
        return self.hotel_name
```

views.py: This code defines two view functions:

home page is accessed.

- **get_hotel(request):** This function handles requests for hotel data. It retrieves hotel objects from the database, allows sorting by price (ascending or descending), and filters by price based on the 'amount' parameter in the GET request. It then converts the hotel information into JSON format and returns it as a JSON response.
- Comments have been added to explain each part of the code and its functionality.

Python3

```python
# views.py
from django.shortcuts import render
from .models import *
from django.http import JsonResponse

# Create your views here.
def home(request):
    return render(request, 'home.html')


def get_hotel(request):
    try:
        Ans_objs = GFG.objects.all()
        if request.GET.get('sort_by'):
            sort_by_value = request.GET.get('sort_by')
            if sort_by_value == 'asc':
                Ans_objs = Ans_objs.order_by('hotel_price')
            elif sort_by_value == 'dsc':
                Ans_objs = Ans_objs.order_by('-hotel_price')

            if request.GET.get('amount'):
                amount = request.GET.get('amount')
                Ans_objs = Ans_objs.filter(hotel_price__lte=amount)

        payload = []
        for Ans_obj in Ans_objs:
            payload.append({
                'name': Ans_obj.hotel_name,
                'price': Ans_obj.hotel_price,
                'description': Ans_obj.hotel_description,
            })
        return JsonResponse(payload, safe=False)

    except Exception as e:
        print(e)

    return JsonResponse({'message': 'Something went wrong !'})
```

admin.py: Then we register out app in admin.py file.

- It imports the necessary modules.
- It defines a custom admin class, HotelAdmin, to specify how the Hotel model is displayed in the admin interface. It chooses which fields to display in the list view.
- It registers the Hotel model with the custom admin class, connecting it to the admin interface so that you can manage hotel records easily.

Python3

```
from django.contrib import admin
from .models import *

# Register your models here.
class GFGAdmin(admin.ModelAdmin):
    list_display = ['hotel_name', 'hotel_price',
                    'hotel_description']

admin.site.register(GFG, GFGAdmin)
```

Project/urls.py: Then we map the project url with app url by adding include method for map the urls.py file of our app

Python

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

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

App/urls.py: Then we write a connect all files by creating the urls in app's urls.py file.

Python3

```
from .views import *
from django.urls import path, include

urlpatterns = [
    path('', home),
    path('api/get_GFG/', get_hotel),
]
```

Templates Folder

In template folder we create the **home.html** file through which we create the representation of our project so let's start

- This is an HTML page that uses Vue.js for interactivity and Bootstrap for styling.

- It displays a list of hotels and provides controls for sorting and filtering hotels by price.
- The JavaScript code within the <script> tags initializes a Vue.js app (var app = new Vue) that fetches hotel data from an API based on sorting and price filtering.
- The data-binding syntax **[[hotel.hotel_name]]** and event listeners like **@change="sortBy"** are used to connect Vue.js to the HTML elements.
- The page loads Bootstrap CSS and JavaScript for styling and functionality.

HTML

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Bootstrap demo</title>
    <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet"
        integrity="sha384-T3c6CoIi6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwwykc2MPK8M2HN" crossorigin="anonymous">
</head>
<style>
    .text-center {
        margin-top: -6%;
        font-weight: bold;
        color: green;
    }

    h4 {
        color: red;
    }
</style>

```

```
<body>
    <div id="app" class="container mt-5 pt-5">
        <h1 class="text-center">GeeksforGeeks </h1>
        <h4>Search/Sort Hotel By Price :- </h4>
        <br>
        <div class="row">
            <div class="col-md-4">
                <label for=""><strong>Sort By</strong></label>
                <select class="form-control" v-model="selectedSort" @change="sortBy">
                    <option value="asc">ASC</option>
                    <option value="dsc">DSC</option>
                </select>
            </div>
            <div class="col-md-4">
                <label for="customRange3" class="form-label"><strong>Select Price</strong></label>
                <input type="range" class="form-range" @change="priceChange($event)" min="1000" max="10000" step="100"
                    id="customRange3">
            </div>
        </div>
        <div class="row mt-5 pt-5">
            <div class="col-md-4" v-for="hotel in hotels">
                <div class="card shadow-lg" style="width: 18rem;">
                    <div class="card-body">
                        <h5 class="card-title">[[hotel.name]]</h5>
                        <p class="card-text">[[(hotel.description).substr(0, 30)]]</p>
                        <a href="#" class="btn btn-primary">Rs. [[hotel.price]]</a>
                    </div>

                </div>
                <br>
            </div>

        </div>
    </div>

    <script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
    <script>
        var app = new Vue({
            delimiters: ['[[', ']]'],
            el: '#app',
            data: {
                message: 'Hello Django Hotel App!',
                hotels: [],
                selectedSort: 'asc', // Default sorting option
            },
            methods: {
                getGFG(sort = '', amount = '') {
                    console.log(sort, amount);
                    fetch(`/api/get_GFG/?sort_by=${sort}&amount=${amount}`)
                        .then(response => response.json())
                        .then(result => {
                            console.log(result);
                            this.hotels = result;
                        });
                },
                sortBy(e) {
                    this.getGFG(sort = this.selectedSort);
                },
                priceChange(e) {
                    this.getGFG(' ', amount = e.target.value, ' ');
                },

            },
            created() {
                this.getGFG();
                console.log("Page loaded");
            }
        });
    </script>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"
        integrity="sha384-C6RzsynM9kWDrMNeT87bh95OGNyZPhcTNXj1NW7RuBCsyN/o0jlpcV8Qyq46cDfL"
```

```
                crossorigin="anonymous"></script>
</body>

</html>
```

Then we need to do migrations for which we run the following commands

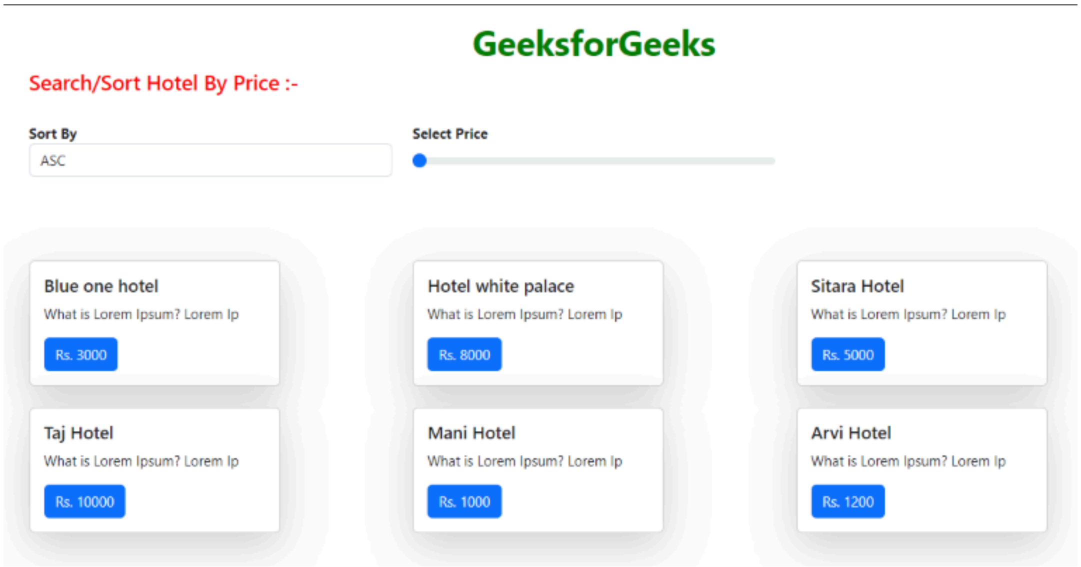
```
python manage.py makemigrations
python manage.py migrate
```

Then we need to create superuser by run the command

```
python manage.py createsuperuser
```

Process of adding data in Backend, and then go to 127.0.0.1:8000/admin and create and fill the hotel details

Output



GeeksforGeeks

Search/Sort Hotel By Price :-

Sort By: ASC

Select Price

Blue one hotel
What is Lorem Ipsum? Lorem Ip
Rs. 3000

Hotel white palace
What is Lorem Ipsum? Lorem Ip
Rs. 8000

Sitara Hotel
What is Lorem Ipsum? Lorem Ip
Rs. 5000

Taj Hotel
What is Lorem Ipsum? Lorem Ip
Rs. 10000

Mani Hotel
What is Lorem Ipsum? Lorem Ip
Rs. 1000

Arvi Hotel
What is Lorem Ipsum? Lorem Ip
Rs. 1200

Home.html

Conclusion

In conclusion, the hotel search app in Django is a web application that allows users to search and sort hotels by price. It utilizes Django for the backend, Vue.js for front-end interactivity, and Bootstrap for styling. Users can filter and sort hotels based on their preferences, and the app retrieves and displays hotel data dynamically. This app demonstrates how to build a simple yet functional hotel search and sorting feature in a Django-based web application.

prath...

Next Article

Searching a list of objects in Python

Similar Reads

Adding Tags Using Django-Taggit in Django Project

Django-Taggit is a Django application which is used to add tags to blogs, articles etc. It makes very easy for us to make adding the tags functionality to our djang...

2 min read

Simple Chat Room using Python

This article demonstrates - How to set up a simple Chat Room server and allow multiple clients to connect to it using a client-side script. The code uses the...

8 min read

How to customize Django forms using Django Widget Tweaks ?

Django forms are a great feature to create usable forms with just few lines of code. But Django doesn't easily let us edit the form for good designs. Here, we...

3 min read

Styling Django Forms with django-crispy-forms

Django by default doesn't provide any Django form styling method due to which it takes a lot of effort and precious time to beautifully style a form. django-crispy...

1 min read

Integrating Django with Reactjs using Django REST Framework

In this article, we will learn the process of communicating between the Django Backend and React js frontend using the Django REST Framework. For the sake...

15+ min read

How to Fix Django "ImportError: Cannot Import Name 'six' from…

After Django 3.0, django.utils.six was removed. The "ImportError: Cannot Import Name 'six' from 'django.utils'" error occurs because of the dependency issue. Thi...

3 min read

Django Introduction | Set 2 (Creating a Project)

Note- This article is in continuation of Django introduction. Popularity of Django Django is used in many popular sites like as: Disqus, Instagram, Knight...

3 min read

How to add RSS Feed and Sitemap to Django project?

This article is in continuation of Blog CMS Project in Django. Check this out here - Building Blog CMS (Content Management System) with Django RSS (Really...

3 min read

Django Project MVT Structure

Django is based on MVT (Model-View-Template) architecture. MVT is a software design pattern for developing a web application. MVT Structure has the followin...

2 min read

How to Create a Basic Project using MVT in Django ?

Prerequisite - Django Project MVT Structure Assuming you have gone through the previous article. This article focuses on creating a basic project to render a...

2 min read

Article Tags : Geeks Premier League Python Django Geeks Premier League 2023

Practice Tags : python

GeeksforGeeks
Sanchhaya Education Private Limited

Corporate & Communications Address:-
A-143, 9th Floor, Sovereign Corporate
Tower, Sector- 136, Noida, Uttar Pradesh
(201305) | Registered Address:- K 061,
Tower K, Gulshan Vivante Apartment,
Sector 137, Noida, Gautam Buddh
Nagar, Uttar Pradesh, 201305

Company **Languages**

About Us
Legal
In Media
Contact Us
Advertise with us
GFG Corporate Solution
Placement Training Program
GeeksforGeeks Community

Python
Java
C++
PHP
GoLang
SQL
R Language
Android Tutorial
Tutorials Archive

DSA

Data Structures
Algorithms
DSA for Beginners
Basic DSA Problems
DSA Roadmap
Top 100 DSA Interview Problems
DSA Roadmap by Sandeep Jain
All Cheat Sheets

Data Science & ML

Data Science With Python
Data Science For Beginner
Machine Learning
ML Maths
Data Visualisation
Pandas
NumPy
NLP
Deep Learning

Web Technologies

HTML
CSS
JavaScript
TypeScript
ReactJS
NextJS
Bootstrap
Web Design

Python Tutorial

Python Programming Examples
Python Projects
Python Tkinter
Web Scraping
OpenCV Tutorial
Python Interview Question
Django

Computer Science

Operating Systems
Computer Network
Database Management System
Software Engineering
Digital Logic Design
Engineering Maths
Software Development
Software Testing

DevOps

Git
Linux
AWS
Docker
Kubernetes
Azure
GCP
DevOps Roadmap

System Design

High Level Design
Low Level Design
UML Diagrams
Interview Guide
Design Patterns
OOAD

Inteview Preparation

Competitive Programming
Top DS or Algo for CP
Company-Wise Recruitment Process
Company-Wise Preparation
Aptitude Preparation
Puzzles

System Design Bootcamp

Interview Questions

School Subjects

Mathematics

Physics

Chemistry

Biology

Social Science

English Grammar

Commerce

World GK

GeeksforGeeks Videos

DSA

Python

Java

C++

Web Development

Data Science

CS Subjects