***COMSATS UNIVERSITY***

Logo, company name

Description automatically generated

**WAH CAMPUS**

***Submitted By***

**Project Name: IMDB**

**Entity: Movies**

***Name:*** *Shayan Babar*

***Registration No:***  *Fa19-Bcs-051*

***Class/Section:*** *BSCS/6D*

***Submitted To***

***Teacher Name: Dr Majid Mumtaz***

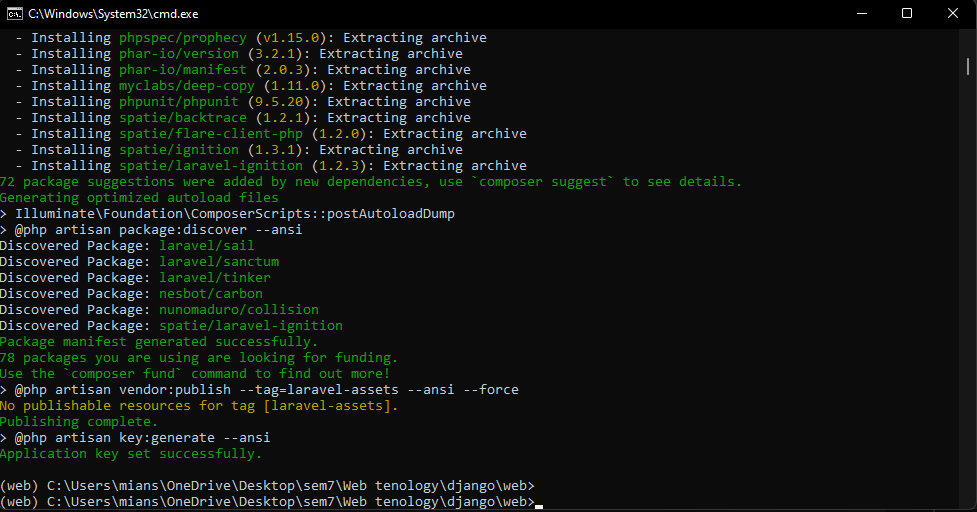
***Date of Submission: 20-05-2022***

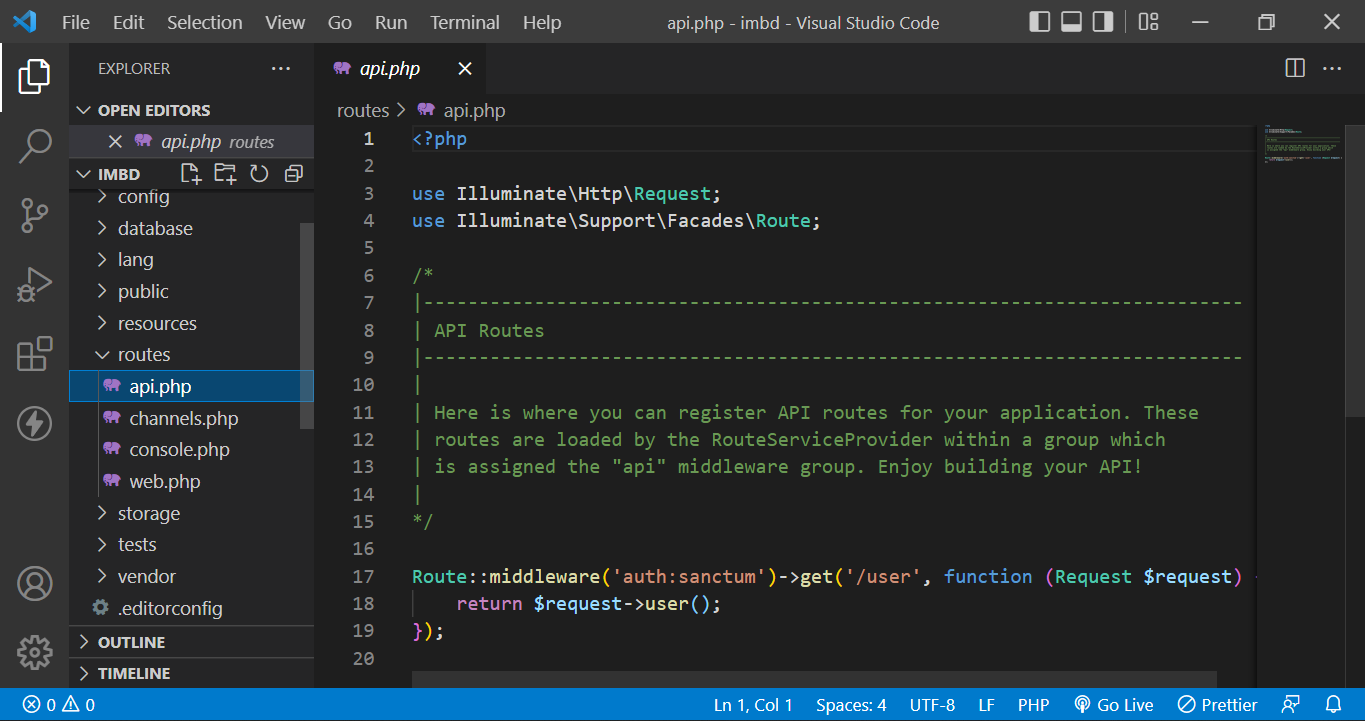
# What is Composer? Create a Laravel project….?

Composer is an application-level dependency manager for the PHP programming language that provides a standard format for managing dependencies of PHP software and required libraries. It was developed by Nils Adermin and Jordi Bogging, who continue to manage the project.

**First in cmd line**

composer create-project laravel/laravel imbd\



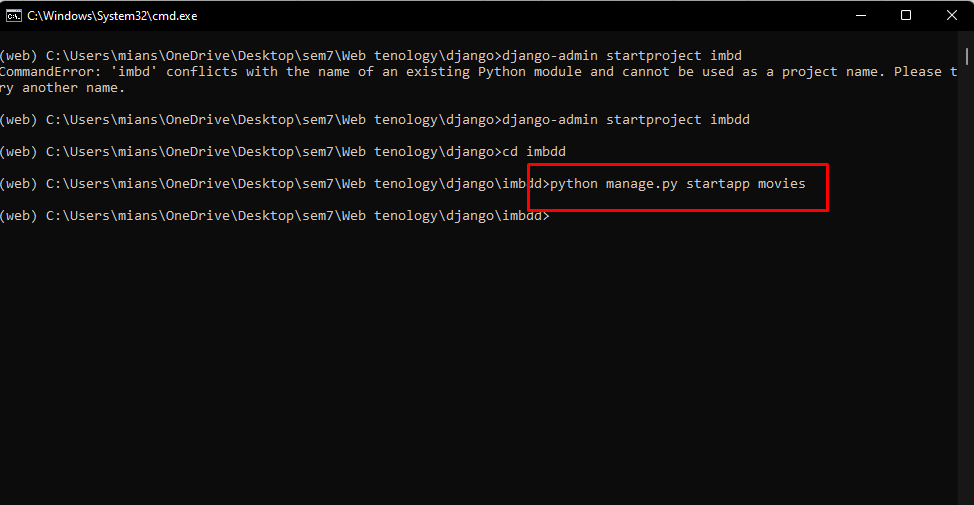


# Question2 crud operation on your semester project

**Step#1 django-admin startproject imbdd**:Text

Description automatically generated

**Step#2:** **python manage.py startapp movies**



**Now open the moelds.py file in movies app and add following code**

from platform import release

from django.db import models

from django.urls import reverse # Used to generate URLs by reversing the

import uuid

# Create your models here.

class movies(models.Model):

    id = models.UUIDField(primary\_key=True, default=uuid.uuid4, help\_text='Unique ID for this particular book across whole library')

    movie\_name = models.CharField(max\_length=40, help\_text='Enter Movie')

    summary = models.TextField(max\_length=1000, help\_text='Enter a brief description of the book')

    Actorname=models.CharField(max\_length=25,help\_text='Enter Actors name')

    release\_date=models.DateField(null=True, blank=True)

    # Methods

    def get\_absolute\_url(self):

        """Returns the URL to access a particular instance of MyModelName."""

        return reverse('model-detail-view', args=[str(self.id)])

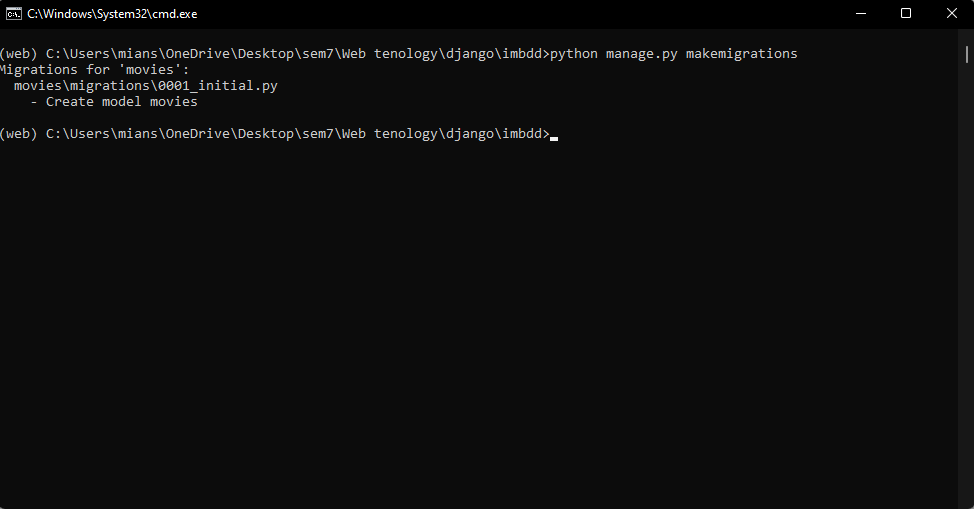
    def \_\_str\_\_(self):

        """String for representing the MyModelName object (in Admin site etc.)."""

        return self.movie\_name

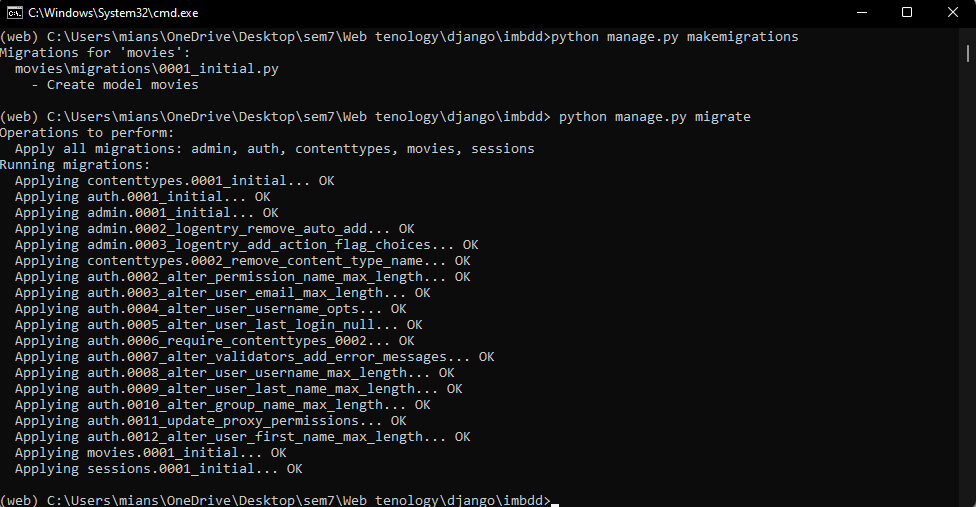
**Step#4**

python manage.py makemigrations



**Step #5**

python manage.py migrate



**Step#6:** python manage.py migrate and superuser

Text

Description automatically generated

**Step#7:Python manage.py runserver (to server the file)**

**Open in local host**

**Step#8 adding the routes**

**Open the urls.py file in imbdd folder an following**

Text

Description automatically generated

**Now create a urls.py file in movies app folder**

**Leave it empty for now**

**Step#8 change admin.py file as below**

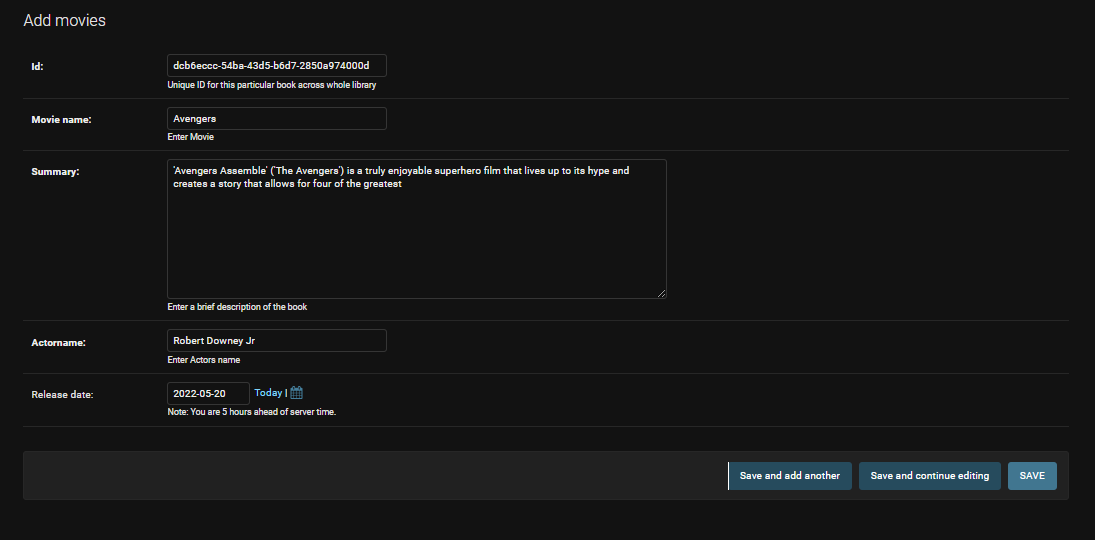
A screenshot of a computer

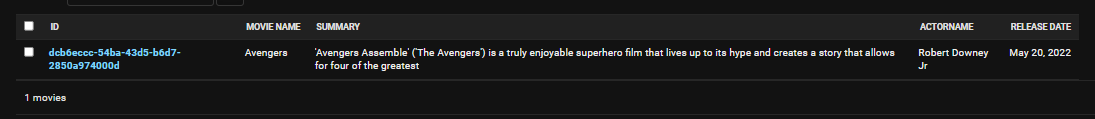
Description automatically generated with medium confidence

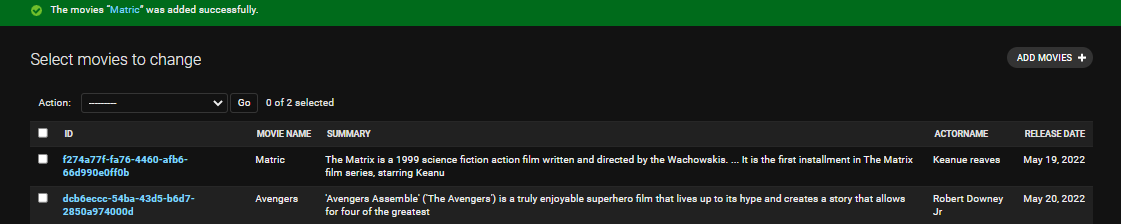
Graphical user interface, application

Description automatically generated

## Step#9 time to add some movies







A screenshot of a computer

Description automatically generated with medium confidence

Graphical user interface, text

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

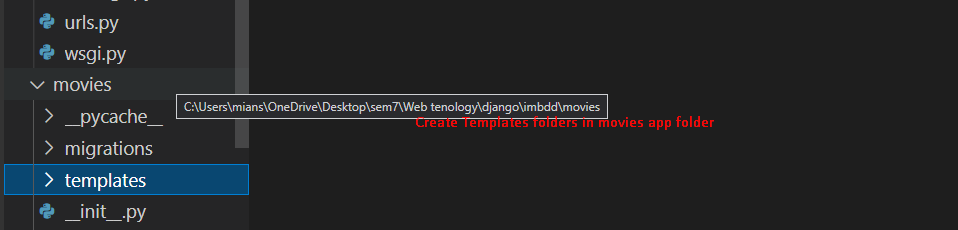
A screenshot of a computer

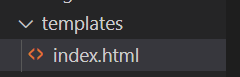
Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

## Step#9 lets create a view template





## To create a simple html view add following code

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

    <style>

        td,th{

            border: 2px solid black;

        }

    </style>

</head>

<body>

    <h1>Articles</h1>

    <ul>

    <table>

        <tr>

            <th>id</th>

            <th>movie\_name </th>

            <th>summary</th>

            <th>actorname</th>

            <th>release\_date</th>

        </tr>

        {% for article in object\_list %}

        <tr>

        <td>{{ article.id }}</td>

        <td> {{article.movie\_name}}</td>

        <td>{{article.summary}}</td>

        <td>{{article.actorname}}</td>

        <td>{{article.release\_date}}</td>

        </tr>

    {% empty %}

        <li>No articles yet.</li>

    {% endfor %}

    </table>

    </ul>

</body>

</html>

**Now in views.py add following code**

Text

Description automatically generated

**Now adding urls for this**

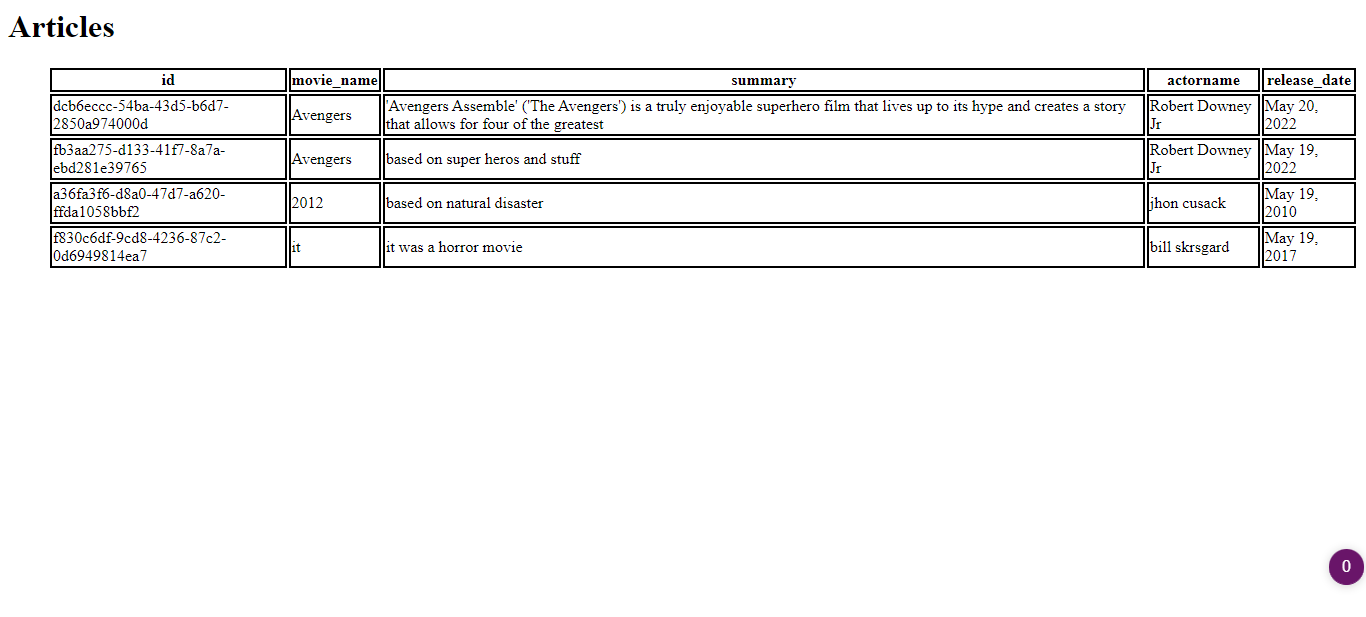
Text

Description automatically generated

**Open**



## So a generic list view is also creates

****

**Now lets add a del icon**

Table

Description automatically generated

Now add delete url

Text

Description automatically generated with medium confidence

Now lets create a view for delete

Graphical user interface

Description automatically generated

## Update functionality=>first add update.html file in templates folder



## Make urls:



## Make views

Text

Description automatically generated

## Finally:

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

# Conclusion:

Hence crud implemented on both using custom and build in methods.

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidenceA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

Table

Description automatically generated with medium confidence

Graphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generated

🡨------------------------------------------------🡪