Django

1.Create new folder. Then open cmd.

- python –m venv venv
- venv\scripts\activate
- pip install django
- django-admin startproject myproject .
- django-admin startapp myApp
- python manage.py runserver

2. Register App in project’s settings.py

```
INSTALLED_APPS = [..., 'appName',]
```

3.Create Model in apps models.py

Example:

```
from django.db import models

class Book(models.Model):

   title = models.CharField(max_length=200)

   author = models.CharField(max_length=100)

   published_date = models.DateField()


  def __str__(self):

      return self.title
```

4.Migrate Commands:

```
python manage.py makemigrations

python manage.py migrate
python manage.py createsuperuser
```

5. Admin Panel Registration

Example:

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

admin.site.register(Book)
```

6. Create forms.py file in app

Example:

```
from django import forms
from .models import Book

class BookForm(forms.ModelForm):
    class Meta:
        model = Book
        fields = ['title', 'author', 'published_date']
```

7. Create Basic CRUD Views in apps views.py

Example:

```
from django.shortcuts import render, redirect, get_object_or_404
from .models import Book
from .forms import BookForm

def book_list(request):
    books = Book.objects.all()
    return render(request, 'books/book_list.html', {'books': books})

def book_create(request):
    if request.method == "POST":
        form = BookForm(request.POST)
        if form.is_valid():
            form.save()
            return redirect('book_list')
    else:
        form = BookForm()
    return render(request, 'books/book_form.html', {'form': form})

def book_update(request, id):
    book = get_object_or_404(Book, id=id)
    if request.method == "POST":
        form = BookForm(request.POST, instance=book)
        if form.is_valid():
            form.save()
            return redirect('book_list')
    else:
        form = BookForm(instance=book)
    return render(request, 'books/book_form.html', {'form': form})

def book_delete(request, id):
    book = get_object_or_404(Book, id=id)
    if request.method == "POST":
        book.delete()
        return redirect('book_list')
    return render(request, 'books/book_confirm_delete.html', {'book': book})
```

8. Create a file urls.py in app & edit

Example:

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

urlpatterns = [
    path('', views.book_list, name='book_list'),
    path('create/', views.book_create, name='book_create'),
    path('update/<int:id>/', views.book_update, name='book_update'),
    path('delete/<int:id>/', views.book_delete, name='book_delete'),
]
```

9. Include app url in projects urls.py

```
from django.contrib import admin
from django.urls import path, include
urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('books.urls')),
  ]
```

10.Create templates folder & html files in it

11.Add templates in settings.py templates

Line 65:'DIRS': [BASE_DIR / 'templates'],

12.If there are pictures in frontend & backend….create static & media folder..

Add CSS,,JS files & images for frontend in static folder..The media folder is for the uploaded images in backend…

13. In settings.py

Line 14. import os

Line 124.

```
STATIC_URL = 'static/'

STATICFILES_DIRS = [BASE_DIR/'static',]

MEDIA_URL = 'media/'

MEDIA_ROOT = os.path.join(BASE_DIR,'media')
```

In projects url.py

```
from . import settings
from django.conf.urls.static import static

urlpatterns + static(settings.MEDIA_URL, document_root = settings.MEDIA_ROOT)
```

14.Command : python manage.py runserver

A model with choices:

```
from django.db import models


class Book(models.Model):

    # Choices define korbo ekta list of tuples diye

    BOOK_TYPES = [
        ('Literature', 'Literature'),
        ('SciFi', 'SciFi'),
        ('Romantic', 'Romantic'),
        ('Comedy', 'Comedy'),
        ('Story', 'Story'),
    ]

    title = models.CharField(max_length=200)
    author = models.CharField(max_length=200)
    published_date = models.DateField()

    # New field with choices
    book_type = models.CharField(max_length=50, choices=BOOK_TYPES)

    def __str__(self):
        return self.title
```

Simple html files to use if gpt made UI prohbited:

book_list.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Book List</title>
</head>
<body>
    <h1>Book List</h1>

    <a href="{% url 'book_create' %}">Add New Book</a>

    <table border="1">
        <thead>
            <tr>
                <th>Title</th>
                <th>Author</th>
                <th>Published Date</th>
                <th>Actions</th>
            </tr>
        </thead>
        <tbody>
            {% for book in books %}
                <tr>
                    <td>{{ book.title }}</td>
                    <td>{{ book.author }}</td>
                    <td>{{ book.published_date }}</td>
                    <td>
                        <a href="{% url 'book_update' book.id %}">Update</a> |
                        <a href="{% url 'book_delete' book.id %}">Delete</a>
                    </td>
                </tr>
            {% empty %}
                <tr>
                    <td colspan="4">No books available.</td>
                </tr>
            {% endfor %}
        </tbody>
    </table>

</body>
</html>
```

book_form.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Book Form</title>
</head>
<body>
    <h1>{% if form.instance.pk %}Update Book{% else %}Add New Book{% endif %}</h1>

 <form method="POST">
        {% csrf_token %}
        {{ form.as_p }}
        <button type="submit">Submit</button>
    </form>
 <br>
    <a href="{% url 'book_list' %}">Back to Book List</a>
</body>
</html>
```

book_confirm_delete.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Confirm Deletion</title>
</head>
<body>
    <h1>Are you sure you want to delete "{{ book.title }}" by {{ book.author }}?</h1>

    <form method="POST">
        {% csrf_token %}
        <button type="submit">Yes, Delete</button>
    </form>

    <br>
    <a href="{% url 'book_list' %}">Cancel</a>
</body>
</html>
```

Jodi models er vitor model bole tahole ekta extra line for foreign key…

```
class Book(models.Model):
    semester = models.ForeignKey(Semester, on_delete=models.CASCADE)
```