Step-by-Step Tutorial: Creating a Django Project with a Table and Three Pages on windows.

In this tutorial, we'll walk through the process of creating a Django project from scratch, setting up a database table, and creating three pages to query and view data from the browser. We'll assume you have basic knowledge of Python and web development concepts.

Before we begin, make sure you have the following installed on your system:

1. Python (https://www.python.org/downloads/)
2. Virtual environment by setting it in the command line(cmd)by typing virtualenv then the name of the environment(venv)

3. Django (Install using pip: `pip install django`)

Step 1: Create a Django Project

First of after installing the virtual environment then you go ahead and activate it by running a command: **venv\scripts\activate or django_env\scripts\activate** to see that your environment is activate you see this (venv) at the beginning of your line but to do this you to be connected on internet.

Open your terminal or command prompt and create a new directory for the project:

Mkdir (is used to create a new folder) my project

Cd my project (you go into the project which u have created above)

Next, create a new Django project using the `django-admin` command: django-admin start project my project. (Then the name of the project).

The above command creates a new Django project with the name `my project` and makes sure that you create it in the current directory.

Step 2: Create a Django App

A Django project consists of one or more apps. Let's create a new app for our project: **python manage.py startapp (my_app)** the name of your app

To see that you successfully installed django go your web browser and type(localhost:8000) you see something like a message which shows that we have installed django successfully.

The next step is to create the admin user this admin will be the one register the user of the app

This command generates a new app named `my_app: **python manage.py create superuser**

Step 3: Define the Model

In this step, we'll define the model for our database table. Open the `models.py` file inside the `my_app` folder and add the following code:

```
from django.db import models


class Item(models.Model):
    name = models.CharField(max_length=100)
    description = models.TextField()
    price = models.IntergeralField(max_length=50, null=false, blank=false)
    def __str__(self):
        return self.name
```

This model represents an `Item` table with three fields: `name`, `description`, and `price`.

Step 4: Create the Database Table

To create the database table based on the model, run the following commands:

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

This will generate and apply the migrations needed to create the table in the database.

Step 5: Create Views

Now, let's create views to handle the requests and render the templates for our three pages. Open the `views.py` file inside the `my_app` folder and add the following code:

```
from django.shortcuts import render

from .models import Item


def home(request):
    items = Item.objects.all()
    return render(request, 'home.html', {'items': items})


def item_detail(request, item_id):
    item = Item.objects.get(pk=item_id)
    return render(request, 'item_detail.html', {'item': item})


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

In the above code, we have defined three views: `home`, `item_detail`, and `about`.

Step 6: Create Templates

Next, we need to create HTML templates for our three pages. Inside the `my_app` folder, create a new folder named `templates`. Inside the `templates` folder, create three HTML files: `home.html`, `item_detail.html`, and `about.html`.

In each of these files, you can add the necessary HTML code to structure the pages and display the data accordingly.

For example, in `home.html`:

```html
```

```
<!DOCTYPE html>
<html>
<head>
  <title>Home</title>
</head>
<body>
  <h1>Items</h1>
  <ul>
    {% for item in items %}
      <li><a href="{% url 'item_detail' item.id %}">{{ item.name }}</a></li>
    {% endfor %}
  </ul>
</body>
</html>
```

Step 7: Create URLs

Lastly, we need to define the URLs to access our views. Open the `urls.py` file inside the `my_app` folder and add the following code:

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

urlpatterns = [
  path('', views.home, name='home'),
  path('item/<int:item_id>/', views.item_detail, name='item_detail'),
  path('about/', views.about, name='about'),
]
```

The above code maps the URLs to the corresponding views.

Step 8: Run the Development Server

Finally, it's time to run our Django development server and see our project in action:

python manage.py runserver

Visit `http://127.0.0.1:8000/` in your web browser to access the home page with the list of items. Clicking on an item name will take you to the item detail page, and you can also access the about page from the navigation.

Congratulations! You've successfully created a Django project with a table and three pages to query and view data from the browser.