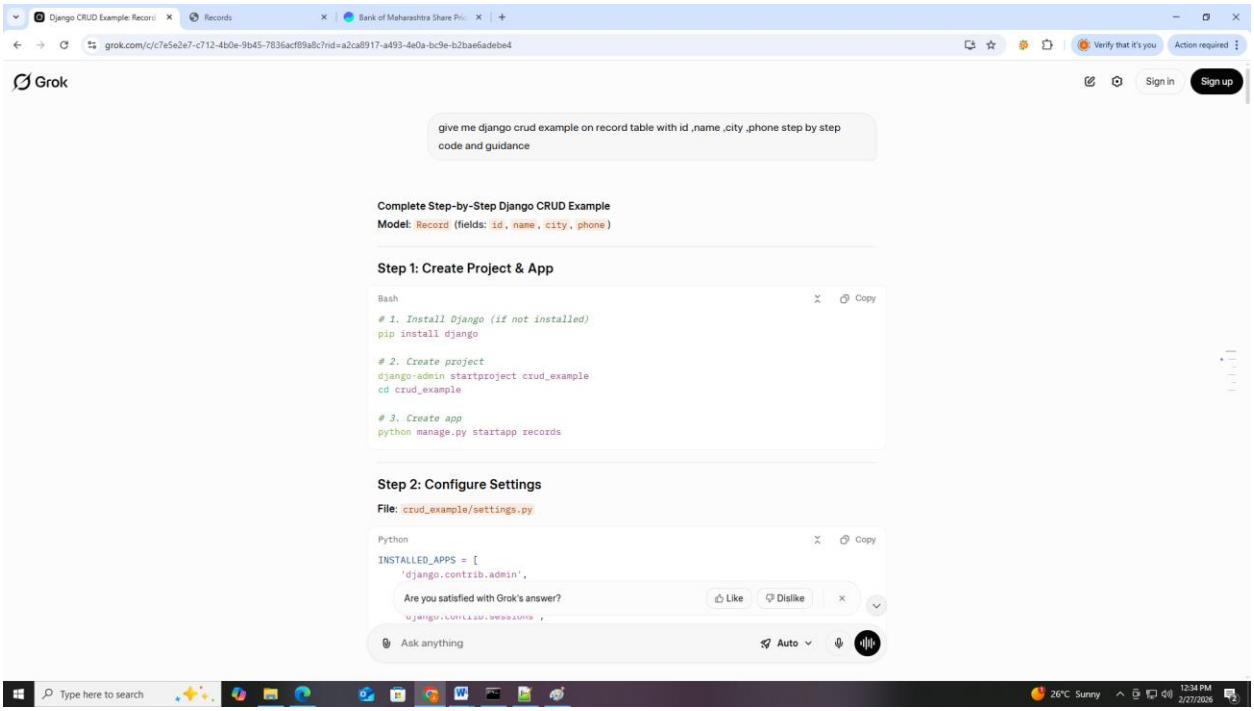
Grok.com Testing for coding Guidance:-



Complete Step-by-Step Django CRUD Example Model: Record (fields: id, name, city, phone)

Step 1: Create Project & App

Bash

```
# 1. Install Django (if not installed)
pip install django

# 2. Create project
django-admin startproject crud_example
cd crud_example

# 3. Create app
python manage.py startapp records
```

Step 2: Configure Settings

File: crud_example/settings.py

Python

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'records',          # ← Add this
]
```

Step 3: Create the Model

File: records/models.py

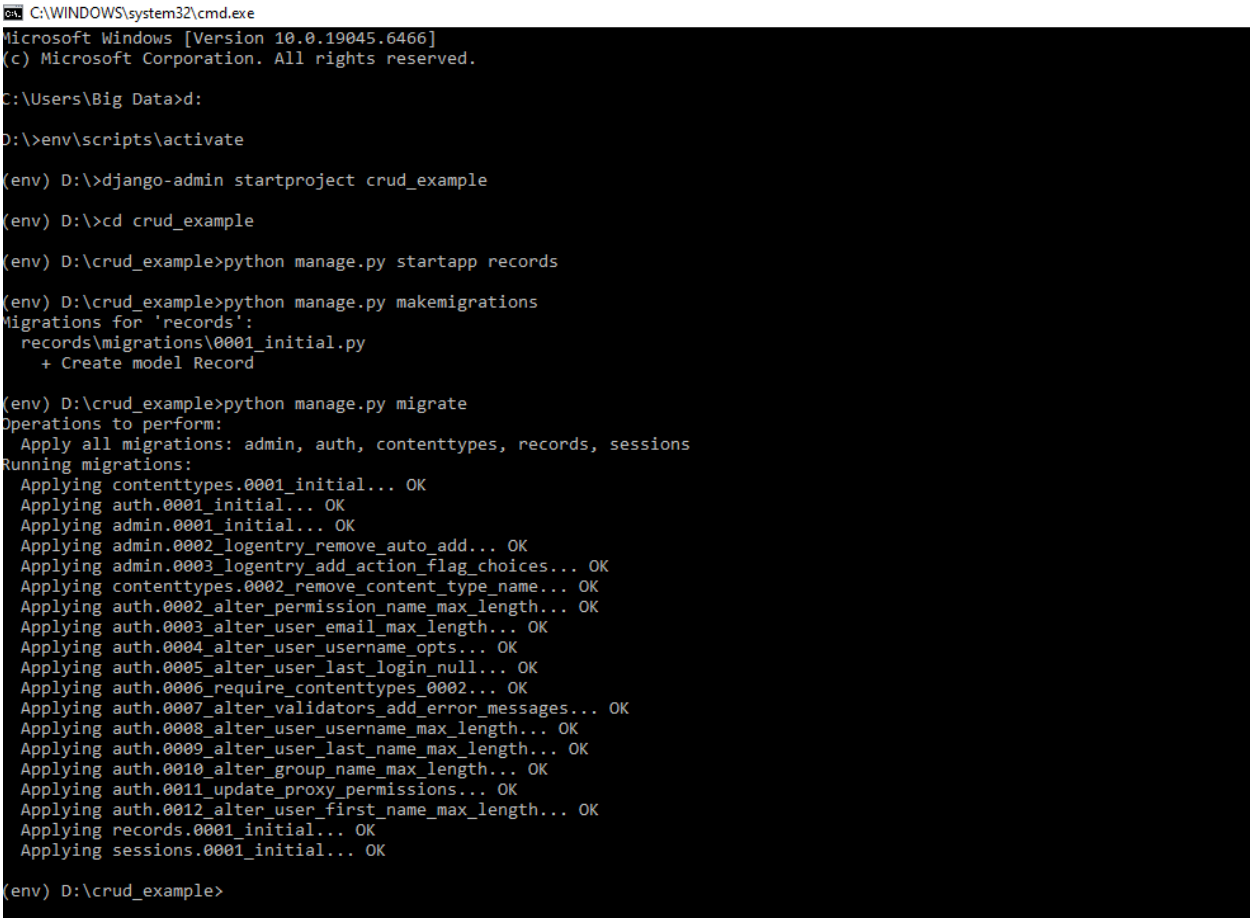
Python

```
from django.db import models

class Record(models.Model):
    name = models.CharField(max_length=100)
    city = models.CharField(max_length=100)
    phone = models.CharField(max_length=15)

    def __str__(self):
        return self.name

    class Meta:
        ordering = ['name']
```

Run migrations:

Bash

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



Step 4: Create Class-Based Views (CRUD)

File: records/views.py

Python

```
from django.views.generic import ListView, DetailView, CreateView,
UpdateView, DeleteView
from django.urls import reverse_lazy
from .models import Record
```

```python
class RecordListView(ListView):
    model = Record
    template_name = 'records/record_list.html'
    context_object_name = 'records'

class RecordDetailView(DetailView):
    model = Record
    template_name = 'records/record_detail.html'

class RecordCreateView(CreateView):
    model = Record
    fields = ['name', 'city', 'phone']
    template_name = 'records/record_form.html'
    success_url = reverse_lazy('record_list')

class RecordUpdateView(UpdateView):
    model = Record
    fields = ['name', 'city', 'phone']
    template_name = 'records/record_form.html'
    success_url = reverse_lazy('record_list')

class RecordDeleteView(DeleteView):
    model = Record
    template_name = 'records/record_confirm_delete.html'
    success_url = reverse_lazy('record_list')
```

Step 5: Set Up URLs

File: records/urls.py (create this file)

Python

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

urlpatterns = [
```

```
    path('', RecordListView.as_view(), name='record_list'),
    path('record/<int:pk>/', RecordDetailView.as_view(),
name='record_detail'),
    path('record/new/', RecordCreateView.as_view(), name='record_create'),
    path('record/<int:pk>/edit/', RecordUpdateView.as_view(),
name='record_update'),
    path('record/<int:pk>/delete/', RecordDeleteView.as_view(),
name='record_delete'),
]
```

File: crud_example/urls.py

Python

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

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('records.urls')),    # ← All records URLs at root
]
```

Step 6: Create Templates

Create folder structure:

records/templates/records

1. records/templates/records/record_list.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Records</title>
<style>
  body { font-family: Arial, sans-serif; margin: 20px; }
  table {
    border-collapse: collapse;
    width: 100%;
    max-width: 900px;
    margin: 20px 0;
  }
  th, td {
    border: 1px solid #ddd;
    padding: 10px;
    text-align: left;
  }
  th {
    background-color: #f4f4f4;
    font-weight: bold;
  }
  tr:nth-child(even) { background-color: #f9f9f9; }
  a { color: #0066cc; text-decoration: none; }
  a:hover { text-decoration: underline; }
  .actions a { margin-right: 8px; }
  .add-btn {
    display: inline-block;
```

```
            padding: 8px 16px;
            background-color: #28a745;
            color: white;
            border-radius: 4px;
            text-decoration: none;
            margin-bottom: 15px;
        }
        .add-btn:hover { background-color: #218838; }
    </style>
</head>
<body>

    <h1>All Records</h1>

    <a href="{% url 'record_create' %}" class="add-btn">+ Add New Record</a>

    <table>
        <thead>
            <tr>
                <th>#</th>          <!-- Changed from ID to visual row number -->
                <th>Name</th>
                <th>City</th>
                <th>Phone</th>
                <th>Actions</th>
            </tr>
```

```
    </thead>
    <tbody>
      {% for record in records %}
      <tr>
        <td>{{ forloop.counter }}</td>  <!-- 1, 2, 3, ... on this page -->
        <td>{{ record.name }}</td>
        <td>{{ record.city }}</td>
        <td>{{ record.phone }}</td>
        <td class="actions">
          <a href="{% url 'record_detail' record.pk %}">View</a> |
          <a href="{% url 'record_update' record.pk %}">Edit</a> |
          <a href="{% url 'record_delete' record.pk %}">Delete</a>
        </td>
      </tr>
      {% empty %}
      <tr>
        <td colspan="5" style="text-align:center; color:#777; padding:30px;">
          No records found.
        </td>
      </tr>
      {% endfor %}
    </tbody>
  </table>

</body>
```

```
</html>
```

2. records/templates/records/record_detail.html

```
<!DOCTYPE html>
<html>
<head><title>Record Detail</title></head>
<body>
    <h1>Record #{{ object.id }}</h1>
    <p><strong>Name:</strong> {{ object.name }}</p>
    <p><strong>City:</strong> {{ object.city }}</p>
    <p><strong>Phone:</strong> {{ object.phone }}</p>
    <hr>
    <a href="{% url 'record_list' %}">Back to List</a>
</body>
</html>
```

3. records/templates/records/record_form.html (used by Create & Update)

HTML

```
<!DOCTYPE html>
<html>
<head><title>{% if object %}Edit{% else %}New{% endif %}
Record</title></head>
<body>
    <h1>{% if object %}Edit{% else %}Create{% endif %} Record</h1>
    <form method="post">
        {% csrf_token %}
        {{ form.as_p }}
        <button type="submit">Save</button>
        <a href="{% url 'record_list' %}">Cancel</a>
    </form>
</body>
</html>
```

4. records/templates/records/record_confirm_delete.html

HTML

```
<!DOCTYPE html>
<html>
<head><title>Delete Record</title></head>
<body>
    <h1>Delete Record #{{ object.id }}</h1>
    <p>Are you sure you want to delete <strong>{{ object.name
}}</strong>?</p>
    <form method="post">
        {% csrf_token %}
        <button type="submit">Yes, Delete</button>
        <a href="{% url 'record_list' %}">Cancel</a>
    </form>
</body>
</html>
```

Step 7: (Optional) Register in Admin

File: records/admin.py

Python

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

admin.site.register(Record)
```

Create superuser:

Bash

```
python manage.py createsuperuser
```

Step 8: Run the Project

Bash

```
python manage.py runserver
```

Open browser: http://127.0.0.1:8000/

You will see the full CRUD interface.