

Crud code explanation in details :-

1. `showform(request)`

```
python Copy Edit

def showform(request):
    form = FormContactForm(request.POST or None)
    if form.is_valid():
        form.save()
    context = {'form': form}
    return render(request, 'contactform.html', context)
```

- **Purpose:** Displays a contact form and saves the submitted data.
- **What's happening:**
 - `form = FormContactForm(request.POST or None)` initializes the form. If data is submitted (`POST`), it uses that data, otherwise it's empty.
 - `form.is_valid()` checks if the submitted data passes validation rules.
 - If valid, the data is saved to the database using `form.save()` .
 - The form (empty or with errors) is then passed to the template `contactform.html` .

2. `home(request)`

```
python Copy Edit

def home(request):
    serviceData = ContactForm.objects.all()
    data = {'serviceData': serviceData}
    return render(request, "home.html", data)
```

- **Purpose:** Displays all saved contact form records.
- **What's happening:**
 - Fetches all records from the `ContactForm` model.
 - Passes the data as `serviceData` to the `home.html` template for display.

3. editemp(request, id)

python

Copy

Edit

```
def editemp(request, id):
    data = ContactForm.objects.get(id=id)
    return render(request, "edit.html", {'editupdaterecord': data})
```

- **Purpose:** Retrieves a specific record to be edited.
 - **What's happening:**
 - Uses `id` to fetch a single record from the `ContactForm` table.
 - Passes that record to `edit.html`, likely to populate a form for editing.
-

4. updateemp(request, id)

python

Copy

```
def updateemp(request, id):
    if request.method == 'POST':
        pi = ContactForm.objects.get(id=id)
        form = FormContactForm(request.POST, instance=pi)
        if form.is_valid():
            form.save()
            msg = "record updated"
            return render(request, "update.html", {'editupdaterecord': msg})
```

- **Purpose:** Handles updating of an existing record.
- **What's happening:**
 - Checks if the method is POST (i.e., form was submitted).
 - Fetches the existing record (`pi`) using the `id`.
 - Creates a form with the submitted data, using the existing record as the instance.
 - If valid, updates the record in the database.
 - Then renders `update.html` with a success message.

Better way coding given Below :-

```
from django.shortcuts import render, get_object_or_404, redirect
from .models import ContactForm
from .forms import FormContactForm # assuming your form class is named this

# Display and handle contact form submission
def showform(request):
    form = FormContactForm(request.POST or None)
    if form.is_valid():
        form.save()
    context = {'form': form}
    return render(request, 'contactform.html', context)

# Display all records on home page
def home(request):
    service_data = ContactForm.objects.all()
    context = {'serviceData': service_data}
    return render(request, "home.html", context)

# Load a specific record in the edit form
def editemp(request, id):
    data = get_object_or_404(ContactForm, id=id)
    form = FormContactForm(instance=data)
    return render(request, "edit.html", {'form': form, 'id': id})

# Update a specific record
def updateemp(request, id):
    record = get_object_or_404(ContactForm, id=id)
    if request.method == 'POST':
        form = FormContactForm(request.POST, instance=record)
        if form.is_valid():
            form.save()
            msg = "Record updated successfully."
            return render(request, "update.html", {'message': msg})
    else:
        form = FormContactForm(instance=record)
    return render(request, "edit.html", {'form': form, 'id': id})

# Delete a specific record
def deleteemp(request, id):
    record = get_object_or_404(ContactForm, id=id)
    if request.method == 'POST':
        record.delete()
    return redirect('home') # assuming 'home' is the name of the URL pattern
```

```
return render(request, "confirm_delete.html", {'record': record})
```

1. contactform.html – Add Contact Form

```
html

<!DOCTYPE html>
<html>
<head>
  <title>Contact Form</title>
</head>
<body>
  <h2>Contact Form</h2>
  <form method="post">
    {% csrf_token %}
    {{ form.as_p }}
    <button type="submit">Submit</button>
  </form>
  <a href="{% url 'home' %}">Back to Home</a>
</body>
</html>
```

2. home.html – List All Records

```
<!DOCTYPE html>
<html>
<head>
  <title>All Contacts</title>
</head>
<body>
  <h2>All Contact Entries</h2>
  <a href="{% url 'showform' %}">Add New Entry</a>
  <table border="1">
    <thead>
      <tr>
        <th>ID</th>
        <th>Name</th>
        <th>Email</th>
        <th>Message</th>
        <th>Actions</th>
      </tr>
    </thead>
    <tbody>
```

```

        {% for entry in serviceData %}
        <tr>
            <td>{{ entry.id }}</td>
            <td>{{ entry.name }}</td>
            <td>{{ entry.email }}</td>
            <td>{{ entry.message }}</td>
            <td>
                <a href="{% url 'editemp' entry.id %}">Edit</a> |
                <a href="{% url 'deleteemp' entry.id %}">Delete</a>
            </td>
        </tr>
        {% empty %}
        <tr><td colspan="5">No records found.</td></tr>
        {% endfor %}
    </tbody>
</table>
</body>
</html>

```

3. edit.html - Edit Entry

```

<!DOCTYPE html>
<html>
<head>
    <title>Edit Contact</title>
</head>
<body>
    <h2>Edit Contact Entry</h2>
    <form method="post" action="{% url 'updateemp' id %}">
        {% csrf_token %}
        {{ form.as_p }}
        <button type="submit">Update</button>
    </form>
    <a href="{% url 'home' %}">Back to Home</a>
</body>
</html>

```

4. update.html - Update Confirmation

```

<!DOCTYPE html>
<html>
<head>
    <title>Update Status</title>

```

```
</head>
<body>
  <h2>{{ message }}</h2>
  <a href="{% url 'home' %}">Back to Home</a>
</body>
</html>
```

5. confirm_delete.html - Confirm Deletion

```
<!DOCTYPE html>
<html>
<head>
  <title>Delete Contact</title>
</head>
<body>
  <h2>Are you sure you want to delete this entry?</h2>
  <p>Name: {{ record.name }}</p>
  <p>Email: {{ record.email }}</p>
  <p>Message: {{ record.message }}</p>

  <form method="post">
    {% csrf_token %}
    <button type="submit">Yes, Delete</button>
    <a href="{% url 'home' %}">Cancel</a>
  </form>
</body>
</html>
```

1. `urls.py` – Route Views to URLs

```
python                                                                    Copy Edit

from django.urls import path
from . import views

urlpatterns = [
    path('', views.home, name='home'),
    path('add/', views.showform, name='showform'),
    path('edit/<int:id>/', views.editemp, name='editemp'),
    path('update/<int:id>/', views.updateemp, name='updateemp'),
    path('delete/<int:id>/', views.deleteemp, name='deleteemp'),
]
```

Make sure this is the `urls.py` inside your app directory, and that your main `urls.py` includes it using:

```
python                                                                    Copy Edit

path('', include('yourappname.urls')),
```

2. `forms.py` – Django Form Class

```
python                                                                    Copy Edit

from django import forms
from .models import ContactForm

class FormContactForm(forms.ModelForm):
    class Meta:
        model = ContactForm
        fields = ['name', 'email', 'message'] # Add/remove fields as needed
```

3. models.py – Django Model for Contact Form

```
python

from django.db import models

class ContactForm(models.Model):
    name = models.CharField(max_length=100)
    email = models.EmailField()
    message = models.TextField()

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

Final Touches:

After setting everything up:

1. Run migrations:

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

2. Create a superuser (optional):

```
python manage.py createsuperuser
```

3. Start the server:

```
python manage.py runserver
```

Now your full **Create, Read, Update, Delete (CRUD)** app should be working perfectly!