2) models.py

```
from django.db import models
class Author(models.Model):
    name = models.CharField(max_length=100)
    book = models.CharField(max_length=100)
    bookid = models.IntegerField()
    def __str__(self):
        return self.name
```

forms.py

```
from django import forms
from .models import Author
class AuthorForm(forms.ModelForm):
    class Meta:
        model = Author
        fields = ['name', 'book', 'bookid']
```

settings.py

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

Terminal

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

views.py

```
from django.shortcuts import render, redirect
from .forms import AuthorForm
def create_author(request):
    if request.method == 'POST':
        form = AuthorForm(request.POST)
        if form.is_valid():
            form.save()
            return redirect('author-list')
    else:
        form = AuthorForm()
    return render(request, 'create_author.html', {'form': form})
```

3)Terminal

```
python manage.py startapp myapp
```

view.py

```
from django.shortcuts import render
def get_name(request):
    return render(request, 'get_name.html')
def post_name(request):
```

```
    if request.method == 'POST':
        first_name = request.POST.get('first_name')
        last_name = request.POST.get('last_name')
        return render(request, 'post_name.html', {'first_name': first_name, 'last_name':
last_name})
    else:
        return render(request, 'get_name.html')
```

get_name.html

```
<form method="GET" action="{% url 'post-name' %}">
    {% csrf_token %}
    <label for="first_name">First name:</label>
    <input type="text" id="first_name" name="first_name" required>
    <br>
    <label for="last_name">Last name:</label>
    <input type="text" id="last_name" name="last_name" required>
    <br>
    <input type="submit" value="Submit">
</form>
```

post_name.py

```
<p>First name: {{ first_name }}</p>
<p>Last name: {{ last_name }}</p>
```

urls.py/app

```
from django.urls import path
from . import views
urlpatterns = [
    path('', views.get_name, name='get-name'),
    path('post/', views.post_name, name='post-name'),
]
```

settings.py

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

4)myapp/forms.py

```
from django import forms
from django.contrib.auth.forms import AuthenticationForm
class LoginForm(AuthenticationForm):
    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
        self.fields['username'].widget.attrs.update({'class': 'form-control'})
        self.fields['password'].widget.attrs.update({'class': 'form-control'})
```

views.py

```
from django.contrib.auth.views import LoginView
from .forms import LoginForm
```

```
class MyLoginView(LoginView):
    form_class = LoginForm
    template_name = 'login.html'
templates/login.html
{% extends 'base.html' %}
{% block content %}
<h2>Login</h2>
<form method="post">
    {% csrf_token %}
    {{ form.as_p }}
    <button type="submit">Login</button>
</form>
{% endblock %}
urls.py
from django.urls import path
from .views import MyLoginView
urlpatterns = [
    path('login/', MyLoginView.as_view(), name='login'),
]
settings.py
INSTALLED_APPS = [
    ...
    'myapp',
    ...
]
4) forms.py
from django import forms
from .models import sampleModel
class SampleModelForm(forms.ModelForm):
    class Meta:
        model = sampleModel
        fields = ['title', 'description']
admin.py
from django.contrib import admin
from .models import sampleModel
from .forms import SampleModelForm
class SampleModelAdmin(admin.ModelAdmin):
    form = SampleModelForm
admin.site.register(sampleModel, SampleModelAdmin)
shell
from myapp.models import sampleModel
sample = sampleModel(title='My Sample Title', description='My Sample Description')
sample.save()
sample = sampleModel.objects.get(title='My Sample Title')
sample.description = 'Updated Sample Description'
sample.save()
```

```
sample = sampleModel.objects.get(title='My Sample Title')
sample.delete()
2)models.py
from django.db import models
class TextFile(models.Model):
    file = models.FileField(upload_to='text_files/')
views.py
from django.shortcuts import render, redirect
from .forms import TextFileForm
def upload_file(request):
    if request.method == 'POST':
        form = TextFileForm(request.POST, request.FILES)
        if form.is_valid():
            form.save()
            return redirect('admin:index')
    else:
        form = TextFileForm()
    return render(request, 'upload_file.html', {'form': form})
template
<form method="post" enctype="multipart/form-data">
    {% csrf_token %}
    {{ form.as_p }}
    <button type="submit">Upload</button>
</form>
from django.contrib import admin
from .models import TextFile
admin.site.register(TextFile)
3)pip install django-dynamic-csv
views.py
from django.shortcuts import render
from dynamic_rawid.admin import DynamicRawIDMixin
from dynamic_rawid.widgets import ManyToManyRawIdWidget
from dynamic_rawid.fields import DynamicRawIDField
from django.views.generic import ListView
from dynamic_csv.views import CSVView
from .models import Employee
class EmployeeListView(DynamicRawIDMixin, ListView):
    model = Employee
class EmployeeCSVView(CSVView):
    model = Employee
    filename = 'employees.csv'
    header = ['Ename', 'EID']
    delimiter = ','
urls.py
from django.urls import path
from .views import EmployeeListView, EmployeeCSVView
```

```
urlpatterns = [
    path('employees/', EmployeeListView.as_view(), name='employee_list'),
    path('employees/csv/', EmployeeCSVView.as_view(), name='employee_csv'),
]
```

html

```
{% extends 'base.html' %}
{% block content %}
  <h1>Employee List</h1>
  <table>
    <thead>
      <tr>
        <th>Name</th>
        <th>ID</th>
      </tr>
    </thead>
    <tbody>
      {% for employee in object_list %}
        <tr>
          <td>{{ employee.Ename }}</td>
          <td>{{ employee.EID }}</td>
        </tr>
      {% endfor %}
    </tbody>
  </table>
  <a href="{% url 'employee_csv' %}">Export CSV</a>
{% endblock %}
```

4)

```
from reportlab.pdfgen import canvas
# Get string message from user
message = input("Enter a message: ")
# Create PDF file
pdf_file = canvas.Canvas("message.pdf")
# Set font and font size
pdf_file.setFont("Helvetica", 12)
# Write message into PDF file
pdf_file.drawString(100, 750, message)
# Save and close the PDF file
pdf_file.save()
print("Message saved in message.pdf")
```