

Databases

- 1) Design a web site using Django, which is a website directory - A site containing links to other websites. A web page has different categories.
- A category table has a name, number of works, and number of views
 - A page table refers to a category, has a title, URL and many views
- Design a form that populates the above database and displays it.

Settings.py

import os

```
INSTALLED_APPS = [
    'dirapp',
    'django.contrib.admin',
    'django.contrib.auth',
]
```

TEMPLATES = [

```
{'BACKEND': 'django.template.backends.django.DjangoTemplates',
'DIRS': [os.path.join(BASE_DIR, 'dirapp/templates')],  

'APP_DIRS': True,  

}]
```

DATABASES = {

```
'default': {
    'ENGINE': 'django.db.backends.postgresql',
    'NAME': 'iHabwk8',
    'USER': 'iHabUser',
    'PASSWORD': 'incorrect',
    'HOST': 'localhost'
}
```

J

J

models.py (present in dirapp)

```

from django.db import models
class Category (models.Model):
    name = models.CharField(max_length=100, primary_key=True)
    number_of_visits = models.IntegerField()
    number_of_likes = models.IntegerField()
class Page (models.Model):
    category = models.CharField(max_length=100)
    title = models.CharField(max_length=100)
    url = models.URLField(primary_key=True)
    views = models.IntegerField()

```

forms.py

```

from dirapp.models import Category
from django import forms
class CategoryForm (forms.Form):
    name = forms.CharField(max_length=100)
    number_of_visits = forms.IntegerField()
    number_of_likes = forms.IntegerField()
class PageForm (forms.Form):
    category = forms.CharField(max_length=100)
    title = forms.CharField(max_length=100)
    url = forms.URLField()
    views = forms.IntegerField()

```

views.py

```

from django.shortcuts import render
from .forms import CategoryForm, PageForm
from .models import Category, Page

```

```

def home(request):
    return render(request, 'proj1.html')

def category(request):
    form = CategoryForm()
    if request.method == 'POST':
        form = CategoryForm(request.POST)
        if form.is_valid():
            name = form.cleaned_data['name']
            no_of_visits = form.cleaned_data['number_of_visits']
            no_of_likes = form.cleaned_data['number_of_likes']
            Category.objects.create(name=name, number_of_visits=no_of_visits,
                                   number_of_likes=no_of_likes)
    return render(request, 'proj1p1.html', {'form': form})

def page(request):
    form1 = PageForm()
    if request.method == 'POST':
        form1 = PageForm(request.POST)
        if form1.is_valid():
            category = form1.cleaned_data['category']
            title = form1.cleaned_data['title']
            url = form1.cleaned_data['url']
            view = form1.cleaned_data['view']
            Page.objects.create(category=category, title=title, url=url, view=view)
    return render(request, 'proj1p2.html', {'form': form1})

def display(request):
    pages = Page.objects.all()
    categories = Category.objects.all()
    return render(request, 'proj1p3.html', {'pages': pages, 'categories': categories})

```

vote.py (from drapp)

```

from django.urls import path
from . import views

```

urlpatterns = [

path('', views.home, name='home')

path('category', views.category, name='category')

path('page', views.page, name='page')

path('display', views.display, name='display')]

use.py from Directory App

from django.contrib import admin

from django.urls import path, include

urlpatterns = [

path('admin/', admin.site.urls),

path('', include('diroapp.urls'))

]

prog1.html

{html}

{head}

{title} Question 1 {/title}

{/head}

{body}

{a href = "#y. url('category')"} Enter information to
category table {/a} {br}

{a href = "#y. url('page')"} Enter information to Page Table {/a}

{a href = "#y. url('display')"} Display category table and page
table {/a} {br}

{/body}

{/html}

prog1p1.html

{html}

{head} {title} Question 1 {/title} {/head}

{body}

{form action = "category" method = "POST"}

{% csrf_token %}

{table} {{ form.as_table }} {/table}

{input type = "submit" value = "insert"}

{/form} {br}

{a href = "#".url 'home'} {/a} Go back to home {/a}

{/body}

{/html}

prog1p2.html

{html}

{head} {title} Question 1 {/title} {/head}

{body}

{form action = "page" method = "POST"}

{% csrf_token %}

{table} {{ form.as_table }} {/table}

{input type = "submit" value = "insert"}

{/form} {br}

{a href = "#".url 'home'} {/a} Go back to home {/a}

{/body}

{/html}

prog1p3.html

{html}

{head} {title} Question 1 {/title} {/head}

{body}

{/tr} Category Table : {/tr}{/tr}

{table}

{thead}

{tr}{td} Name:{/td}

{td} Number of Visits:{/td}

{td} Number of Likes:{/td}

{thead}

{tr} for category in categories

{tr}

{td} Name {{category.name}} {/td}

{td} {{category.number_of_visits}} {/td}

{td} {{category.number_of_likes}} {/td}

{tr}

{tr} endfor

{table}{/tr}

{tr} Page Table : {/tr}{/tr}

{table}

{thead}

{td} Category {/td}

{td} Title {/td}

{td} URL {/td}

{td} View {/td}

{thead}

{tr} for page in pages

{tr}

{td} {{page.category}} {/td}

{td} {{page.title}} {/td}

{td} {{page.url}} {/td}

{td} {{page.view}} {/td}

{tr}

{tr} endfor

{/table} {/br}
{a href="#" style="text-decoration: none; color: inherit;">Back to Home {/a}

{/body}

{/html}

Output

Enter Information to category table

Edit Information to Page table

Display category table and page table

Name :
Number of Visits :
Number of Likes :
Insert

Go back to home

Category Table :

Name	Number of Visits	Number of Likes
shopping	100	750
Inorbit	1000	85326
Megamart	624623	45623

Page Table :

Category Title URL

Entertainment Netflix <http://www.netflix.com/>
Furniture IKEA <http://ikea.us.com/>

2) Consider the following tables:

WORKS (person-name, company-name, salary)

LIVES (Person-name, Street, City)

Assume that data suitably. Design a Django webapp and include an option to insert data into WORKS table by accepting data from the user using Textarea. Also, include an option to retrieve the names of people who work for a particular company along with the cities they live in (particular company name must be accepted from the user).

models.py

```
from django.db import models
```

```
from django.db.models.related import ForeignKey
```

```
class Works (models.Model):
```

```
    name = models.CharField(max_length=100)
```

```
    company = models.CharField(max_length=100)
```

```
    salary = models.IntegerField()
```

```
class Lives (models.Model):
```

```
    name = models.CharField(max_length=100)
```

```
    street = models.CharField(max_length=200)
```

```
    city = models.CharField(max_length=50)
```

forms.py

```
from django import forms
```

```
class Employee (forms.Form):
```

```
    name = forms.CharField(max_length=100)
```

```
    company = forms.CharField(max_length=100)
```

```
    salary = forms.IntegerField()
```

```
    street = forms.CharField(max_length=200)
```

```
    city = forms.CharField(max_length=50)
```

```
class Company (forms.Form):
```

company = forms.CharField(max_length=100)

views.py

from django.shortcuts import render

from .models import Works, Lines

from .forms import Employee, Company

def home(request):

return render(request, 'proj2.html')

def portal(request):

form = Employee()

form1 = Employee(request, POST)

if form1.is_valid():

name = form1.cleaned_data['name']

company = form1.cleaned_data['company']

salary = form1.cleaned_data['salary']

street = form1.cleaned_data['street']

city = form1.cleaned_data['city']

Works.objects.create(name=name, company=company, salary=salary)

Lines.objects.create(name=name, street=street, city=city)

return render(request, 'proj3p1.html', {"form": form})

def search(request):

form = Company()

form1 = Company(request, POST)

if form1.is_valid():

company = form1.cleaned_data['company']

employees = Works.objects.all().filter(company=company)

employees = []

for e in employees:

employees.append(Lines.objects.get(name=e.name))

return render(request, "proj3p2.html", {"form": form1, "employees": employees})

return render(request, "proj3p2.html", {"form": form})

~~empapp~~~~empapp/urls.py~~

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

```
from . import views
```

```
urlpatterns = [
```

```
    path('', views.home, name='home')
```

```
    path('portal', views.portal, name='portal')
```

```
    path('search', views.search, name='search')
```

```
]
```

Employee/ urls.py

```
from django.contrib import admin
```

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

```
urlpatterns = [
```

```
    path('admin/', admin.site.urls),
```

```
    path('', include('pr.empapp.urls')),
```

```
]
```

page2.html

{% extends "base.html" %}

{% block title %} Question 2 {% endblock %}

{% block %}

{% if req == "GET" and 'portal' in req %} Update Employee Portal Status

{% if req == "GET" and 'search' in req %} Find the employee list of a company

{% block %}

{% endblock %}

page2.bl.html

{% extends "base.html" %}

```
{/head} {/title} Question 2{/title}{/head}
{body}
  {form action = "portal" method = "POST"}
    {x-csrf-token} x
  {/form}
  {if form.on-table}
    {input type = "submit" value = "Insert"}
  {/if}
  {a href = "#j. url('home') x"} Go back to home {/a}
{/body}
{/html}
```

proj2b2.html

```
{/html}
{body}
  {head} {title} Question 2{/title}{/head}
  {body}
    {form action = "search" method = "POST"}
      {x-csrf-token} x
    {/form}
    {input type = "submit" value = "Search"} {/br}
    {/form}{/br}
    {input type = "table"}
      {thead}
        {td} Name {/td}
        {td} City {/td}
      {/thead}
      {for employee in employees}
        {tr}
          {td} {{employee.name}} {/td}
          {td} {{employee.city}} {/td}
        {/tr}
      {/for}
    {/input}
  {/body}
```

[Start]

(a) Log - "S.Y. will 'Home' I.Y." Go back to home [Start]

[Body] An option is provided with a right arrow key
to move to the next section, a scroll bar is also present to

move up and down between sections of menu or function menu

Output

An output window is present in the middle of screen.

It shows all records been entered with their record number

along with certain code in

Update Employee portal

Find the employee list of a companyName : Company : Salary : Street : City : InsertGo back to homeCompany : Search

Name

City

Rohan

Cupertino

Thomas

Cupertino

Go back to home

- 3) There are three tables in the database: an author table has a first name, a last name and an email address. A publisher table has a name, a street address, a city, a state/province, a country, and a website. A book table has a title and a publication date. It also has one or more authors (a many-to-many relationship with authors) and a single publisher (a one-to-many relationship - aka foreign key). Design a form which populates and retrieves the information from the chart database using Django.

models.py

```
from django.db import models
```

```
from django.db.models.aggregates import Count
```

```
class Publisher(models.Model):
```

```
    name = models.CharField(max_length=100)
```

```
    street = models.CharField(max_length=200)
```

```
    city = models.CharField(max_length=50)
```

```
    state = models.CharField(max_length=50)
```

```
    country = models.CharField(max_length=50)
```

```
    site = models.URLField()
```

```
class Author(models.Model):
```

```
    fname = models.CharField(max_length=100)
```

```
    lname = models.CharField(max_length=100)
```

```
    email = models.EmailField()
```

```
class Book(models.Model):
```

```
    title = models.CharField(max_length=200)
```

```
    pubdate = models.DateField()
```

```
    authors = models.ManyToManyField(Author)
```

```
    publisher = models.ForeignKey(Publisher, on_delete=models.CASCADE)
```

forms.py

from django import forms

class PublisherForm(forms.Form):

name = forms.CharField(max_length=100)

street = forms.CharField(max_length=200)

city = forms.CharField(max_length=50)

state = forms.CharField(max_length=50)

country = forms.CharField(max_length=50)

site = forms.URLField()

class AuthorForm(forms.Form):

name = forms.CharField(max_length=100)

name = forms.CharField(max_length=100)

email = forms.EmailField()

class BookForm(forms.Form):

title = forms.CharField(max_length=200)

pubdate = forms.DateField(label="publication date")

phone = forms.CharField(max_length=100, label="Publisher name")

names = forms.CharField(max_length=400, label="Enter first
names of authors by space separation")

class BookSearch(forms.Form):

title = forms.CharField(max_length=200)

class AuthorSearch(forms.Form):

name = forms.CharField(max_length=100, label="Enter the first name")

class PublisherSearch(forms.Form):

name = forms.CharField(max_length=100)

views.py

from django.shortcuts import render

from forms import AuthorForm, PublisherForm, BookForm, AuthorSearch, PublisherSearch, BookSearch

```
from models import Au, Publisher, Book
```

```
def home(request):
```

```
    return render(request, 'proj3.html')
```

```
def publisher(request):
```

```
    form = PublisherForm()
```

```
    form1 = PublisherForm(request.POST)
```

```
    if form1.is_valid():
```

```
        name = form1.cleaned_data['name']
```

```
        street = form1.cleaned_data['street']
```

```
        city = form1.cleaned_data['city']
```

```
        state = form1.cleaned_data['state']
```

```
        country = form1.cleaned_data['country']
```

```
        site = form1.cleaned_data['site']
```

```
Publisher.objects.create(name=name, street=street, city=city, state=state,
```

```
        country=country, site=site)
```

```
return render(request, 'proj3p1.html', {'form': form})
```

```
def authorEntry(request):
```

```
    form = AuthorForm()
```

```
    form1 = AuthorForm(request.POST)
```

```
    if form1.is_valid():
```

```
        fromm = form1.cleaned_data['fromm']
```

```
        knowm = form1.cleaned_data['knowm']
```

```
        email = form1.cleaned_data['email']
```

```
Au.objects.create(fromm=fromm, knowm=knowm, email=email)
```

```
return render(request, 'proj3p2.html', {'form': form})
```

```
def bookEntry(request):
```

```
    form = BookForm()
```

```
    form1 = BookForm(request.POST)
```

```
    if form1.is_valid():
```

```
        a = form1.cleaned_data
```

```
        title = a['title']
```

```
        pdot = a['pdot']
```

```

    phone = a['phone']
    name = a['name'].split()
    print(name)
    publisher = Publisher.objects.get(name=name)
    authors = []
    book = Book(title=title, price=price, publisher=publisher)
    book.save()
    for i in aname:
        a = Au.objects.get(fname=i)
        books = book.authors.add(a)
        book.save()
    return render(request, 'proj3p3.html', {'form': form})
def searchBook(request):
    form = BookSearch()
    form1 = BookSearch(request.POST)
    if form1.is_valid():
        title = form1.cleaned_data['title']
        book = Book.objects.get(title=title)
        return render(request, 'proj3p3.html', {'form': form1, 'book': book})
    return render(request, 'proj3p3.html', {'form': form})
def searchAuthor(request):
    form = AuthorSearch()
    form1 = AuthorSearch(request.POST)
    if form1.is_valid():
        fname = form1.cleaned_data['fname']
        author = Au.objects.get(fname=fname)
        return render(request, 'proj3p5.html', {'form': form1, 'author': author})
    return render(request, 'proj3p5.html', {'form': form1})
def searchPublisher(request):
    form = PublisherSearch()
    form1 = PublisherSearch(request.POST)
    if form1.is_valid():

```

```

name = form[0].cleaned_data["name"]
publisher = Publisher.objects.get(name=name)
return render(request, 'pg3pb.html', {"form": form, "publisher": publisher})
return render(request, 'pg3pb.html', {"form": form})
    
```

url libapp.urls.py

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

```
from . import views
```

```
urlpatterns = [
```

```

path('', views.home, name='home'),
path('publisherEntry', views.publisherEntry, name='publisherEntry'),
path('authorEntry', views.authorEntry, name='authorEntry'),
path('bookEntry', views.bookEntry, name='bookEntry'),
path('searchBook', views.searchBook, name='searchBook'),
path('searchAuthor', views.searchAuthor, name='searchAuthor'),
path('searchPublisher', views.searchPublisher, name='searchPublisher')
    
```

]

Library/urls.py

```
from django.contrib import admin
```

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

```
urlpatterns = [
```

```

path('admin/', admin.site.urls),
    
```

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

]

prog 3 pt 1.html

```
{html}
```

```
<head> <title> Publisher Entry </title> </head>
```

```
{body}
```

```
<h1> Publisher Registration : </h1>
```

```
<form action = "publisherEntry" method = "POST">
```

```
    <input type = "text" name = "name" />
```

```
    <table> <tr> <td> form.on - table </td> </tr>
```

```
    <input type = "submit" value = "register" />
```

```
</form>
```

```
<a href = "?x. url 'home' y. b" > Go back to home </a>
```

```
</body>
```

```
</html>
```

prog 3.html

```
{html}
```

```
<head> <title> Main Page </title> </head>
```

```
{body}
```

```
<a href = "?x. url 'publisherEntry' y. b" > Register a publisher </a>
```

```
<a href = "?x. url 'authorEntry' y. b" > Register an author </a> <b>
```

```
<a href = "?x. url 'bookEntry' y. b" > Register a book </a> <br>
```

```
<a href = "?x. url 'searchBook' y. b" > Search for a book </a> <br>
```

```
<a href = "?x. url 'searchAuthor' y. b" > Search for an author </a>
```

```
<a href = "?x. url 'searchPublisher' y. b" > Search for a publisher </a>
```

```
</body>
```

```
</html>
```

proj3fb.html

{/html}

{/head} {/title} Author Entry {/title} {/head}

{/body}

{/h1} Author Registration {/h1}

{form action = "authorEntry" method = "POST"}

{/input type = "text" name = "name" value = "John"}

{/table} {/form} {/div} {/table}

{input type = "submit" value = "register"}

{/form}

{a href = "{/f. wsl 'home' }"} Go back to home {/a}

{/body}

{/html}

proj3fb2.html

{/html}

{/head} {/title} Book Entry {/title} {/head}

{/body}

{/h1} Book Registration {/h1}

{form action = "bookEntry" method = "POST"}

{/input type = "text" name = "name" value = "John"}

{/table} {/form} {/div} {/table}

{input type = "submit" value = "register"}

{/form}

{a href = "{/f. wsl 'home' }"} Go back to home {/a}

{/body}

{/html}

~~book3.html~~

{HTML}

{Read} {HTML} Book Search {HTML} {Read}

{Body}

{HTML} Search for book {HTML}

{form action = "searchBook" method = "post"}

{input type = "text" name = "bookname" value = "Search"}

{form}

{Table}

{Thread}

{Td} Title {Td}

{Td} Published Date {Td}

{Td} Name of the Publisher {Td}

{Td} Name of the Authors {Td}

{/Thread}

{try}

{td} {{book.title}} {Td}

{td} {{book.published}} {Td}

{td} {{book.publishername}} {Td}

{td} {{book.authors}} {Td}

{if for author in book.authors.all: }

{author.name} {author.name} {/if}

{if end for x: }

{/td}

{/tr}

{/table}

{a href = "{if not 'home': 'x'}"} Go back to home {/a}

{/body}

{/html}

page 3 part 5.html

{/html}

{head} {title} Author Search{/title} {/head}

{body}

{/h1} Search for Author{/h1}

From action = "searchAuthor" method = "POST"

py-wsgi-form is performed

{form} input type = "submit" value = "Search"

{/form}

{table}

{thead}

{tr} First Name{/td}

{tr} Last Name{/td}

{tr} Email{/td}

{/thead}

{tr}

{td} {{author}}.from {{td}}

{td} {{author}}.names {{td}}

{td} {{author}}.em {{td}}

{/tr}

{/table}

{a href = "#"} Go back to home{/a}

{/body}

{/html}

page 3 part 6.html

{/html}

{head} {title} Publisher Search{/title} {/head}

{body}

{/h1} Search for Publisher{/h1}

Form action = "search Publisher" method = "post"
Ex-cogn-token \rightarrow $\{$ form $\}$

Form input type = "submit" value = "Search" $\}$

{/form}

{/table}

{/thead}

{/td} Title Name {/td}

{/td} Street {/td}

{/td} City {/td}

{/td} State {/td}

{/td} Country {/td}

{/td} Website {/td}

{/thead}

{/tr}

{/td} $\{$ publisher.name $\}$ {/td}

{/td} $\{$ publisher.street $\}$ {/td}

{/td} $\{$ publisher.city $\}$ {/td}

{/td} $\{$ publisher.state {/td}

{/td} $\{$ publisher.country $\}$ {/td}

{/td} $\{$ publisher.website {/td}

{/tr}

{/table}

So long = "if x url 'home' \Rightarrow " } Go back to home {/a}

{/body}

{/html}

Output

- Register a publisher
- Register an author
- Search a book
- Search for a book
- Search for an author
- Search for a publisher

Book Registration :

Title :

LOTR

Publication Date :

9th Aug 1956

Publisher name :

Penguin Books

Enter first name of author by space separator : J.R.R. Tolkien

Registers

Go back to home

Author Registration :

First name :

J.R.

Last name :

Tolkien

Email :

jmr.Hn@gmail.com

Registers

Go back to home

Publisher Registration :

Name :

Penguin Books

Street :

Westminster

City :

London

State :

England

Country :

UK

Site :

<http://www.pbn.com/>

Registers

Go back to home

Search for Author

enter the first name : J.K.R.

Search

First Name

Last Name

Email

J.K.R

Tolkien

jrr.tkn@gmail.com

Go back to home

Search for Publisher

Name : Penguin Books

Search

Name

street

City

W.state

Country

Sit

Penguin Books Westminster London England UK [http://www.
pgn.com/](http://www.pgn.com/)

Go back to home

Search for book

Title : LOTR

Search

Title Published Date Name of the Publisher Name of the author
LOTR 8th Aug 1954 Penguin Books J.K.R Tolkien

Go back to home

- 4) Create a Django Page for entry of a Product information (name, and description) and save it to the db. Create the index page where you would view the product entries in an unordered list.

models.py

```
from django.db import models
class Product(models.Model):
    title = models.CharField(max_length=100)
    price = models.IntegerField()
    desc = models.TextField()
```

forms.py

```
from django import forms
class ProductForm(forms.Form):
    title = forms.CharField(max_length=100)
    price = forms.IntegerField()
    desc = forms.CharField(widget=forms.Textarea, label='description')
```

views.py

```
from django.shortcuts import render
from .forms import ProductForm
from .models import Product
def home(request):
    return render(request, 'proj1.html')
def entry(request):
    form1 = ProductForm(request.POST)
    form = ProductForm()
    if form1.is_valid():
        pass
```

```

title = form1.cleaned_data['title']
prior = form1.cleaned_data['prior']
desc = form1.cleaned_data['desc']
Product.objects.create(title=title, prior=prior, desc=desc)
return render(request, 'proj1p1.html', {'form': form1})
def product(request):
    products = Product.objects.all()
    return render(request, 'proj1p2.html', {'products': products})

```

prodapp/urls.py

```

from django.urls import path
from . import views
urlpatterns = [
    path('', views.home, name='home'),
    path('entry', views.entry, name='entry'),
    path('index', views.index, name='index')
]

```

Product/urls.py

```

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

```

proj1.html

```

<head><title>Home</title></head>
<body>

```

{a href = "f.y. url 'order' y. j"} Enter a new product {/a},
{a href = "f.y. url 'index' y. j"} View Products {/a},
{/body}
{/html}

proj 4p1.html

{html}

{head} {title} Entry Page {/title} {/head}

{body}

{form action = "entry" method = "post"}

f.y. wif-token xj

{table} {form as table} {/table}

{input type = "submit" value = "Add"}

{/form}

{a href = "f.y. url 'home' y. j"} Go back to home {/a}

{/body}

{/html}

proj 4p2.html

{html}

{head} {title} Index {/title} {/head}

{body}

{h1} products: {/h1} {br}

{ul}

f.y. for product in products xj

{li} {{product.title}} {br} {{product.price}} {br}

{{product.desc}} {/li}

f.y. endfor r.j

{ul} {/body}

{/html}

Output

Enter a new product

New Product

Title : Cello Pen

Price : ₹ 20

Description :

Waterproof pen

Add

Go back to home

Products :

or

• Cello Pen

₹ 20

Waterproof Pen

• Reynolds Prism

₹ 55

Gel Pen

• Compass Box

₹ 250

Used for geometric and mathematical purposes