WP LAB 9– Databases – Part I

Name: Pranamya G Kulal
Class: CSE A
Reg no: 220905018
Roll no: 8

Q1) 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 visits, and number of likes.
• 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.

i) settings.py

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'directory',
]

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': BASE_DIR / 'db.sqlite3',
    }
}
```

ii) website_directory/urls.py

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

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('directory.urls')),
]
```

iii) directory/urls.py

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

urlpatterns = [
    path('', views.category_list, name='category_list'),
    path('category/<int:category_id>/', views.page_list, name='page_list'),
    path('category/add/', views.add_category, name='add_category'),
    path('category/<int:category_id>/add/', views.add_page, name='add_page'),
]
```

iv) views.py

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

def category_list(request):
    categories = Category.objects.all()
    return render(request, 'directory/category_list.html', {'categories': categories})

def page_list(request, category_id):
    category = Category.objects.get(id=category_id)
    pages = Page.objects.filter(category=category)
    return render(request, 'directory/page_list.html', {'category': category, 'pages': pages})

def add_category(request):
    if request.method == 'POST':
        form = CategoryForm(request.POST)
        if form.is_valid():
            form.save()
            return redirect('category_list')
    else:
        form = CategoryForm()
    return render(request, 'directory/add_category.html', {'form': form})

def add_page(request, category_id):
    category = Category.objects.get(id=category_id)
    if request.method == 'POST':
        form = PageForm(request.POST)
        if form.is_valid():
            page = form.save(commit=False)
            page.category = category
            page.save()
            return redirect('page_list', category_id=category.id)
    else:
        form = PageForm()
    return render(request, 'directory/add_page.html', {'form': form, 'category': category})
```

v) models.py

```
from django.db import models

class Category(models.Model):
    name = models.CharField(max_length=255)
    visits = models.IntegerField(default=0)
    likes = models.IntegerField(default=0)

    def __str__(self):
        return self.name

class Page(models.Model):
    category = models.ForeignKey(Category, on_delete=models.CASCADE, related_name='pages')
    title = models.CharField(max_length=255)
    url = models.URLField()
    views = models.IntegerField(default=0)
    def __str__(self):
        return self.title
```

vi) forms.py

```
from django import forms
from .models import Category, Page

class CategoryForm(forms.ModelForm):
    class Meta:
        model = Category
        fields = ['name', 'visits', 'likes']

class PageForm(forms.ModelForm):
    class Meta:
        model = Page
        fields = ['category', 'title', 'url', 'views']
```

vii) admin.py

```
from django.contrib import admin
from .models import Category, Page

admin.site.register(Category)
admin.site.register(Page)
```

viii) add_category.html

```
{% load static %}

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Add Category</title>
    <link rel="stylesheet" href="{% static 'css/styles.css' %}">
</head>
<body>
    <div class="container">
        <h1>Add Category</h1>
        <form method="post">
            {% csrf_token %}
            {{ form.as_p }}
            <button type="submit">Save</button>
        </form>
    </div>
</body>
</html>
```

ix) add_page.html

```
{% load static %}

<!DOCTYPE html>
```

```
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Add Page</title>
    <link rel="stylesheet" href="{% static 'css/styles.css' %}">
</head>
<body>
    <div class="container">
        <h1>Add Page to {{ category.name }}</h1>
        <form method="post">
            {% csrf_token %}
            {{ form.as_p }}
            <button type="submit">Save</button>
        </form>
    </div>
</body>
</html>
```

x) category_detail.html

```
<!DOCTYPE html>
<html>
<head>
    <title>{{ category.name }} - Details</title>
</head>
<body>
    <h1>{{ category.name }}</h1>
    <p>Visits: {{ category.visits }}</p>
    <p>Likes: {{ category.likes }}</p>

    <h2>Pages:</h2>
    <ul>
        {% for page in pages %}
            <li>{{ page.title }} - Views: {{ page.views }} <a href="{{ page.url }}"
target="_blank">{{ page.url }}</a></li>
        {% endfor %}
    </ul>

    <a href="{% url 'index' %}">Back to Categories</a>
</body>
</html>
```

xi) category_list.html

```
{% load static %}

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Website Directory</title>
    <link rel="stylesheet" href="{% static 'css/styles.css' %}">
```

```
</head>
<body>
    <div class="container">
        <h1>Categories</h1>
        <ul>
            {% for category in categories %}
                <li><a href="{% url 'page_list' category.id %}">{{ category.name }}</a></li>
            {% endfor %}
        </ul>
        <a href="{% url 'add_category' %}">Add Category</a>
    </div>
</body>
</html>
```

xii) page_list.html

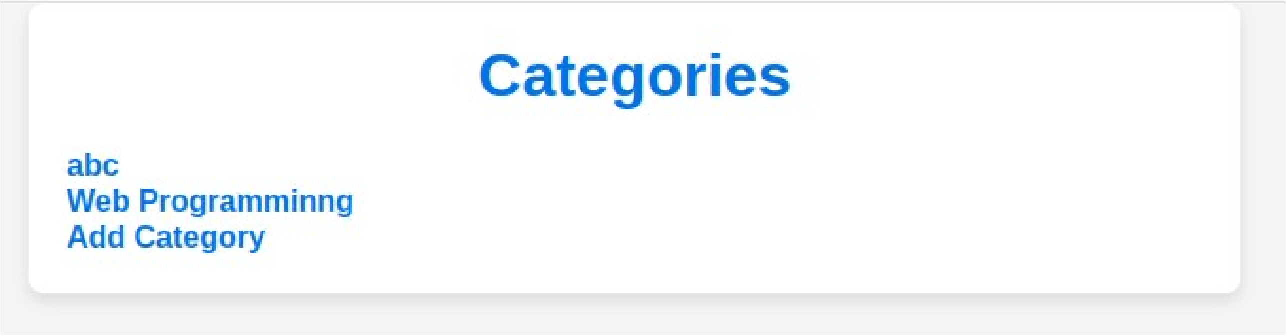
```
{% load static %}

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Pages in {{ category.name }}</title>
    <link rel="stylesheet" href="{% static 'css/styles.css' %}">
</head>
<body>
    <div class="container">
        <h1>Pages in {{ category.name }}</h1>
        <ul>
            {% for page in pages %}
                <li><a href="{{ page.url }}">{{ page.title }}</a></li>
            {% endfor %}
        </ul>
        <a href="{% url 'add_page' category.id %}">Add Page</a>
    </div>
</body>
</html>
```

xiii) style.css

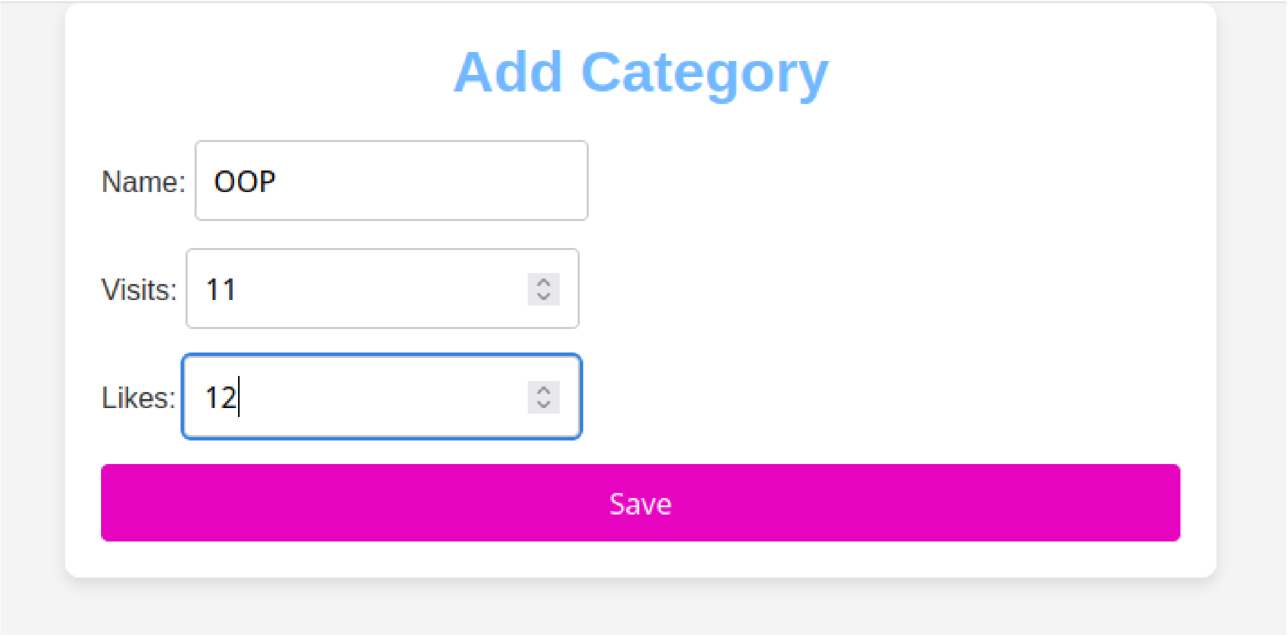
```
/* directory/static/css/styles.css */

/* Basic reset */
* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}

body {
    font-family: Arial, sans-serif;
    background-color: #f4f4f4;
    color: #333;
}

h1 {
    color: #71b8ff;
    text-align: center;
    margin-bottom: 20px;
}

ul {
    list-style-type: none;
    padding: 0;
}

a {
    text-decoration: none;
    color: #64008b;
    font-weight: bold;
}

a:hover {
    text-decoration: underline;
}

.container {
    width: 80%;
    margin: 0 auto;
    padding: 20px;
    background-color: #fff;
    border-radius: 8px;
    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
}

form {
    display: flex;
    flex-direction: column;
    gap: 15px;
    margin-top: 20px;
}

input, button, select, textarea {
    padding: 10px;
    border: 1px solid #ccc;
    border-radius: 4px;
    font-size: 16px;
}

button {
    background-color: #e600c0fb;
    color: white;
    cursor: pointer;
    border: none;
}
```

```
button:hover {
    background-color: #005bb5;
}
```

xiv) Screenshots


Categories

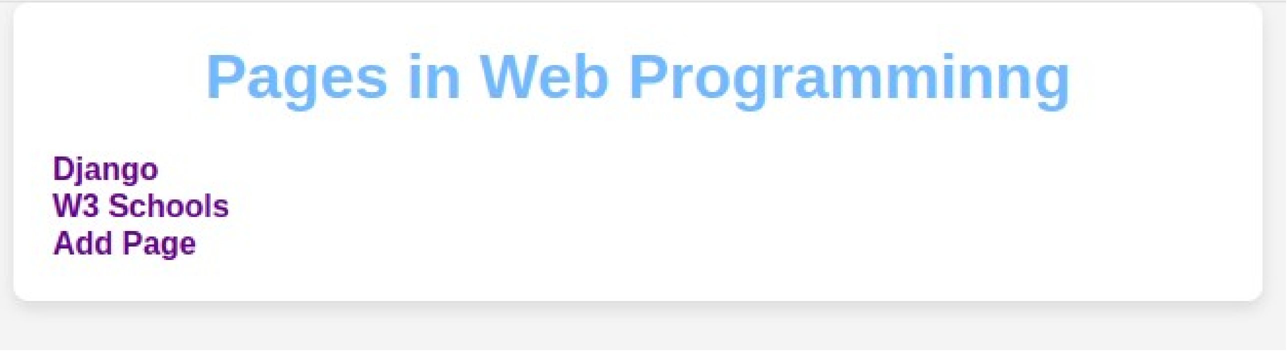
abc
Web Programminng
Add Category


Add Category

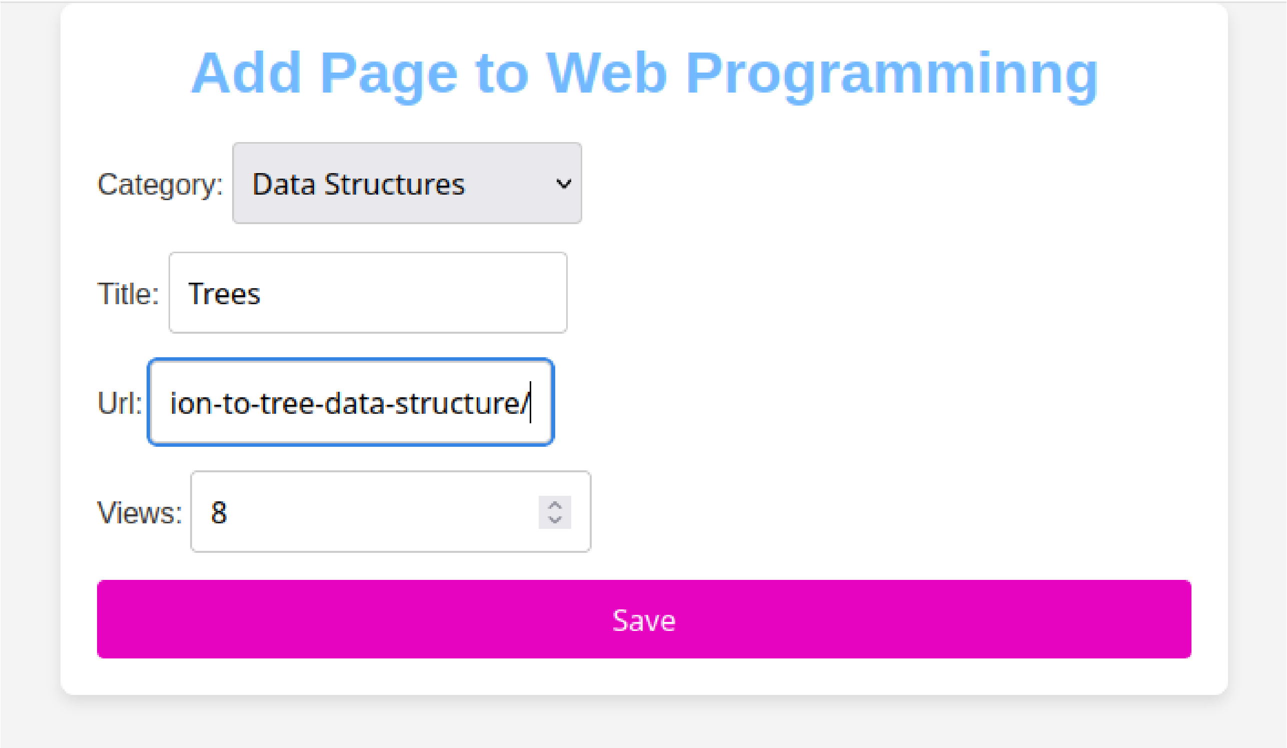
Name: OOP

Visits: 11

Likes: 12


Pages in Web Programminng

Django
W3 Schools
Add Page



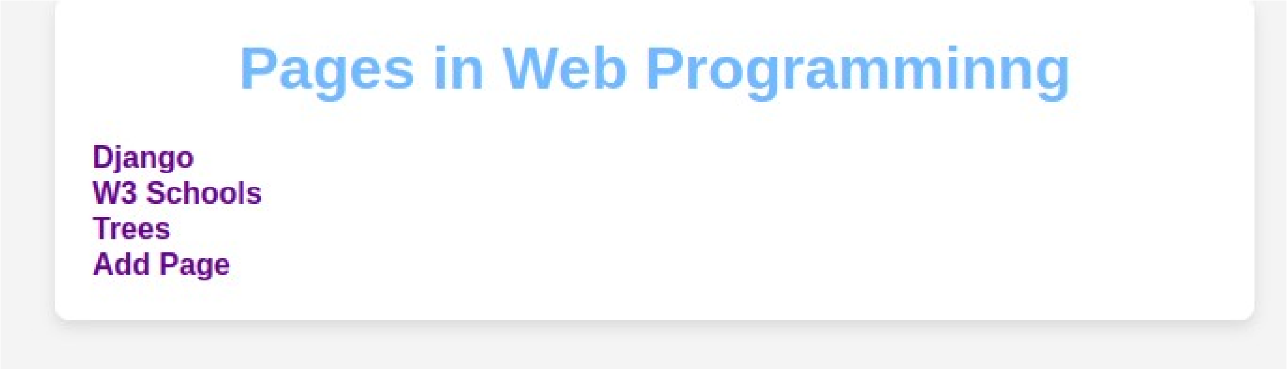
Add Page to Web Programminng

Category: Data Structures

Title: Trees

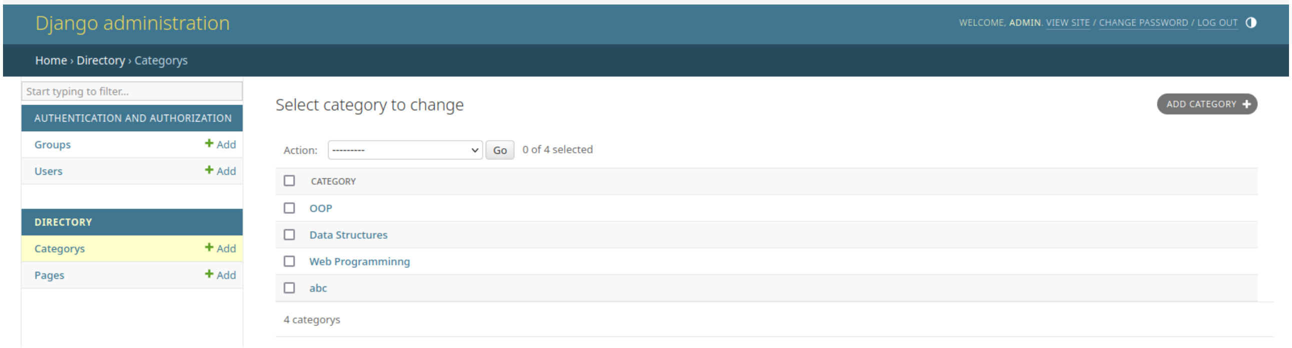
Url: ion-to-tree-data-structure/

Views: 8



Pages in Web Programminng

Django
W3 Schools
Trees
Add Page



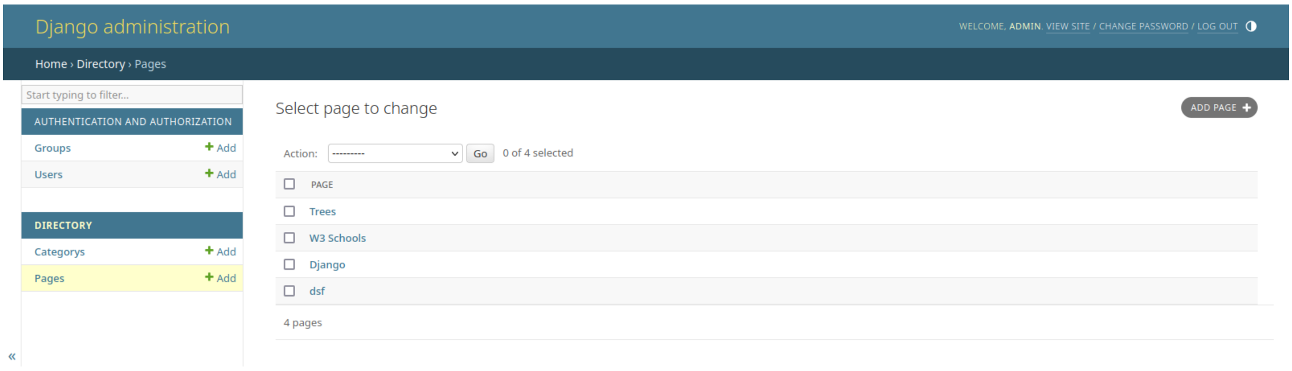
Django administration
WELCOME, ADMIN. VIEW SITE / CHANGE PASSWORD / LOG OUT
Home › Directory › Categorys
Start typing to filter...
AUTHENTICATION AND AUTHORIZATION
Groups + Add
Users + Add
DIRECTORY
Categorys + Add
Pages + Add
ADD CATEGORY +
Go

Select category to change

Action: --------- Go 0 of 4 selected

□	CATEGORY
□	OOP
□	Data Structures
□	Web Programminng
□	abc

4 categorys



Select page to change

Action: --------- Go 0 of 4 selected

□	PAGE
□	Trees
□	W3 Schools
□	Django
□	dsf

4 pages

Q2. Consider the following tables:
WORKS(person-name,Company-name,Salary)
LIVES(Person_name, Street, City)
Assume Table data suitably. Design a Django webpage and include an option to insert data into WORKS table by accepting data from the user using TextBoxes. 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).

i) settings.py

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'company'
]

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': BASE_DIR / 'db.sqlite3',
    }
}
```

ii) company_directory/urls.py

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

urlpatterns = [
    path('admin/', admin.site.urls),
    path('company/', include('company.urls')),
]
```

iii) company/urls.py

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

urlpatterns = [
    path('insert/works/', views.insert_works_data, name='insert_works_data'),
    path('insert/lives/', views.insert_lives_data, name='insert_lives_data'),
    path('search/', views.search_company, name='search_company'),
]
```

iv) views.py

```
from django.shortcuts import render
from .forms import WorksForm, LivesForm, CompanySearchForm
from .models import Works, Lives
```

```
def insert_works_data(request):
    if request.method == 'POST':
        form = WorksForm(request.POST)
        if form.is_valid():
            form.save()
            message = "Work data inserted successfully!"
            return render(request, 'company/insert_works_data.html', {'form': form, 'message':
message})
    else:
        form = WorksForm()
    return render(request, 'company/insert_works_data.html', {'form': form})

def insert_lives_data(request):
    if request.method == 'POST':
        form = LivesForm(request.POST)
        if form.is_valid():
            form.save()
            message = "Lives data inserted successfully!"
            return render(request, 'company/insert_lives_data.html', {'form': form, 'message': message})
    else:
        form = LivesForm()
    return render(request, 'company/insert_lives_data.html', {'form': form})

def search_company(request):
    if request.method == 'POST':
        form = CompanySearchForm(request.POST)
        if form.is_valid():
            company_name = form.cleaned_data['company_name']
            works = Works.objects.filter(company_name=company_name)
            people = []
            for work in works:
                person_name = work.person_name
                # Get related city information
                person_lives = Lives.objects.filter(person_name=person_name).first()
                if person_lives:
                    city = person_lives.city
                    people.append({'name': person_name, 'city': city})
                else:
                    people.append({'name': person_name, 'city': 'No city found'})
            return render(request, 'company/search_result.html', {'people': people, 'company_name':
company_name})
    else:
        form = CompanySearchForm()
    return render(request, 'company/search_company.html', {'form': form})
```

v) models.py

```
from django.db import models

class Works(models.Model):
    person_name = models.CharField(max_length=100)
    company_name = models.CharField(max_length=100)
    salary = models.DecimalField(max_digits=10, decimal_places=2)
    def __str__(self):
        return f"{self.person_name} works at {self.company_name}"

class Lives(models.Model):
    person_name = models.CharField(max_length=100)
    street = models.CharField(max_length=200)
    city = models.CharField(max_length=100)

    def __str__(self):
        return f"{self.person_name} lives at {self.street}, {self.city}"
```

vi) forms.py

```
from django import forms
from .models import Works, Lives

class WorksForm(forms.ModelForm):
    class Meta:
        model = Works
        fields = ['person_name', 'company_name', 'salary']

class LivesForm(forms.ModelForm):
    class Meta:
        model = Lives
        fields = ['person_name', 'street', 'city']

class CompanySearchForm(forms.Form):
    company_name = forms.CharField(max_length=100, label="Enter Company Name")
```

vii) admin.py

```
from django.contrib import admin
from .models import Works, Lives

# Register the Works model
admin.site.register(Works)

# Register the Lives model
admin.site.register(Lives)
```

viii) search_result.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Search Results</title>
</head>
<body>
    <h1>People working at {{ company_name }}</h1>
    <table border="1">
        <tr>
            <th>Name</th>
            <th>City</th>
        </tr>
        {% for person in people %}
            <tr>
                <td>{{ person.name }}</td>
                <td>{{ person.city }}</td>
            </tr>
        {% endfor %}
    </table>
</body>
</html>
```

ix) search_company.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Search Company</title>
</head>
<body>
    <h1>Search for People Working at a Company</h1>
    <form method="post">
        {% csrf_token %}
        {{ form.as_p }}
        <button type="submit">Search</button>
    </form>
</body>
</html>
```

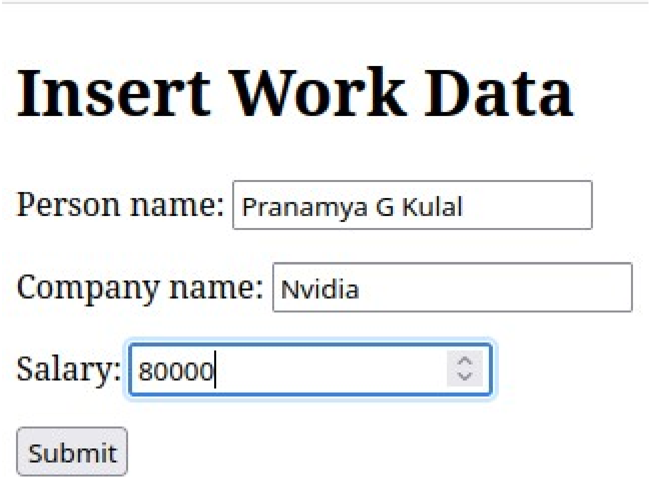
x) insert_works_data.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Insert Work Data</title>
</head>
<body>
    <h1>Insert Work Data</h1>
    <form method="post">
        {% csrf_token %}
        {{ form.as_p }}
        <button type="submit">Submit</button>
    </form>
    {% if message %}
        <p>{{ message }}</p>
    {% endif %}
</body>
</html>
```

xi) insert_lives_data

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Insert Work Data</title>
</head>
<body>
    <h1>Insert Work Data</h1>
    <form method="post">
        {% csrf_token %}
        {{ form.as_p }}
        <button type="submit">Submit</button>
    </form>
    {% if message %}
        <p>{{ message }}</p>
    {% endif %}
</body>
</html>
```

xii) Screenshots

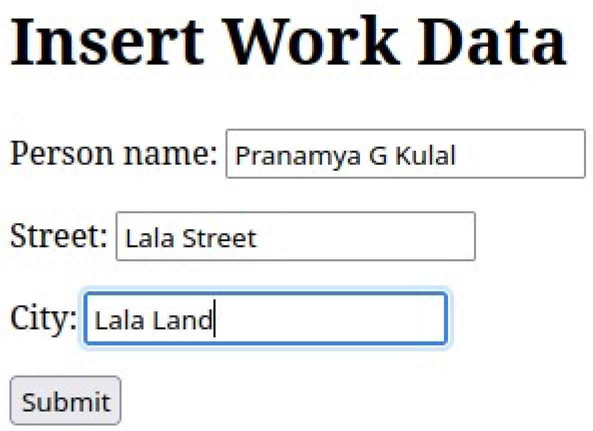


Insert Work Data

Person name: Pranamya G Kulal

Company name: Nvidia

Salary: 80000

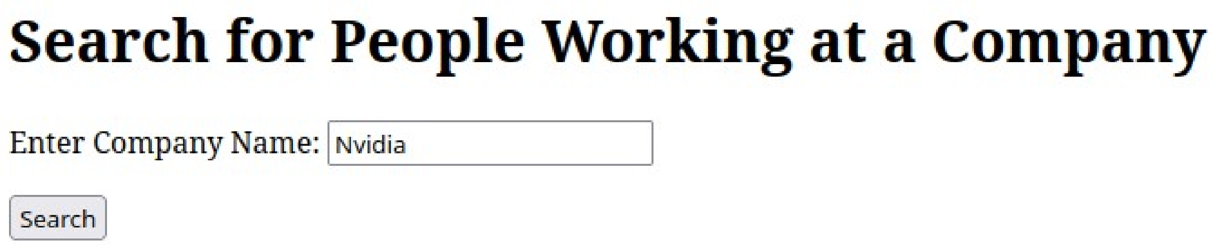


Insert Work Data

Person name: Pranamya G Kulal

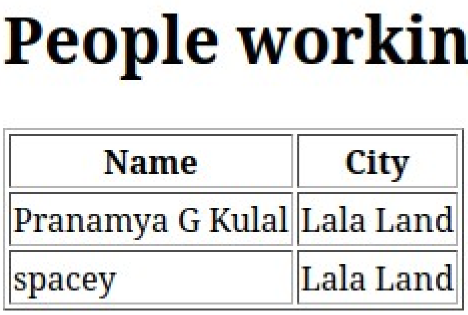
Street: Lala Street

City: Lala Land



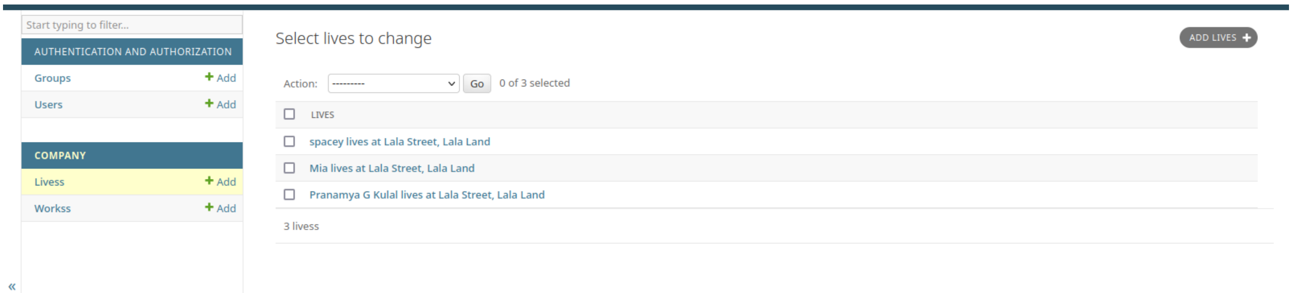
Search for People Working at a Company

Enter Company Name: Nvidia



People working at Nvidia

Name	City
Pranamya G Kulal	Lala Land
spacey	Lala Land

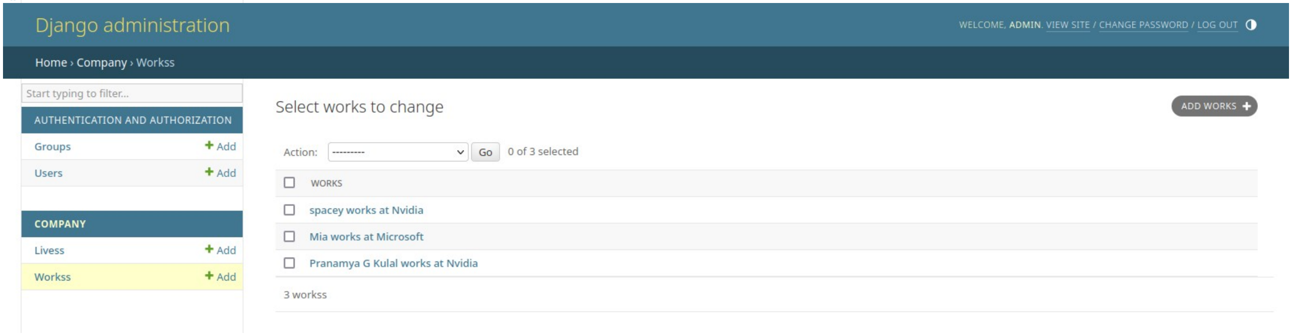


Select lives to change

Action: --------- Go 0 of 3 selected

□	LIVES
□	spacey lives at Lala Street, Lala Land
□	Mia lives at Lala Street, Lala Land
□	Pranamya G Kulal lives at Lala Street, Lala Land

3 livess



Django administration
WELCOME, ADMIN. VIEW SITE / CHANGE PASSWORD / LOG OUT
Home › Company › Workss
Start typing to filter...
AUTHENTICATION AND AUTHORIZATION
Groups + Add
Users + Add
COMPANY
Livess + Add
Workss + Add

Select works to change

Action: --------- Go 0 of 3 selected

□	WORKS
□	spacey works at Nvidia
□	Mia works at Microsoft
□	Pranamya G Kulal works at Nvidia

3 workss