**Project Name: Django Company Count**

**1. Overview**

Django Company Count is a web-based application developed using the Django web framework.

It includes user authentication, registration.

we can upload file and can apply query on that to get desired data from file.

**Technologies Used**

**Django**: Web framework for rapid development.

**Python**: Programming language

**PostgreSQL**: Database

**HTML, CSS**: For frontend design.

**JavaScript**: For interactive features on the frontend.

**Bootstrap**: For responsive and modern UI design.

**2. Installation and Setup**

**Prerequisites**

Before running this project, ensure that you have the following installed:

* Python 3.x
* pip (Python package installer)
* Django 4.x (or specify the version used)
* Virtual env (optional but recommended)

**Steps to Set Up Locally**

django-admin startproject catalyst\_count

cd catalyst\_count

python manage.py runserver

**python manage.py startapp company**

**Install Dependencies**: Install the required Python packages using pip:

pip install -r requirements.txt

**Configure the Database**: this project uses PostgreSQL.

createdb -h localhost -p 5432 -U postgres catalyst\_db

settings.py.

DATABASES = {

    'default': {

        'ENGINE': 'django.db.backends.postgresql',

        'NAME': 'catalyst\_db',

        'USER': 'postgres',

        'PASSWORD': 'root',

        'HOST': 'localhost',

        'PORT': '5432',

    }

}

**Run Migrations**: To set up the database tables, run the following commands:

python manage.py makemigrations

python manage.py migrate

**Create a Superuser** (for accessing the Django admin panel):

python manage.py createsuperuser

**Run the Development Server**: Start the Django development server by running:

python manage.py runserver

**Access the Application**: Open your browser and go to http://127.0.0.1:8000/ to view the application.

**3. Project Structure**

Here’s an overview of the key files and directories in the project:

/project-directory

│

├── /catalyst\_count/ Django project settings

│ ├── settings.py Main settings configuration

│ ├── urls.py URL configuration

│ └── wsgi.py WSGI entry point for deployment

│

├── /company/ Main application directory

│ ├── models.py Database models

│ ├── views.py Views handling requests

│ ├── urls.py App-specific URLs

│ ├── forms.py Django forms (if used)

│ └── templates/ HTML templates for the app

├── /static/ Static files (CSS, JS, images)

├── /templates/ Global templates (for the entire project)

│

├── manage.py Django management script

├── requirements.txt List of dependencies

└── README.md Project documentation (this file)

**4. Features**

**User Authentication**

* **Login**: Users can log in using their email/username and password.
* **Registration**: New users can sign up by providing their email, username, and password.
* **Password Reset**: Users can reset their password if they forget it.
* **Logout**: Authenticated users can log out from the system.

Data Upload

Purpose: Allows users to upload data files to the platform.

Usage:

Click Browse to select a file from your local machine.

Press Start Upload to begin the upload process.

A progress bar displays the status of the upload.

**Query Builder**

**Purpose**: Allows users to search for records based on various parameters such as:

* **Keyword**: Search based on specific terms.
* **Industry**: Filter results by industry type.
* **Year Founded**: Filter by the year a company or entity was founded.
* **Location**: Filter based on city, state, and country.
* **Employee Size**: Filter based on the number of employees (both lower and upper bounds).

**Usage**:

* Enter the desired filters and click **Query Data** to get results.
* A success message will display how many records were found.
* The **Reset** button clears all filters.

**3 User Management**

**Purpose**: Manage users of the platform.

**Usage**:

* View the current users, their email addresses, and account statuses.
* Add new users using the **Add User** button.
* The system displays a confirmation message when a new user is successfully added.

**URL Structure**

Here’s a brief overview of the primary URLs for the platform:

| **URL Path** | **View/Functionality** |
| --- | --- |
| /upload/ | File upload interface |
| /query-builder/ | Query builder for filtering records |
| /users/ | User management dashboard |
| /login/ | User login |
| /logout/ | User logout |
| /admin/ | Django admin panel |