PYTHON Assignment

MODULE: 4 (DB and Python Framework)

1. Why Django should be used for web-development? Explain how you can create a project in Django?

Ans. Django is a popular and powerful web development framework for building web applications using the Python programming language. Here are some reasons why Django is often chosen for web development:

- **High-level Abstractions**: Django provides high-level abstractions for common web development tasks, allowing developers to focus on building the application's features rather than dealing with low-level details. It uses simplified, easy-to-understand concepts or tools that hide the complexity of underlying processes or technologies. It follows the "Don't Repeat Yourself" (DRY) and "Convention over Configuration" principles, reducing boilerplate code.
- **ORM (Object-Relational Mapping):** Django includes a robust ORM that allows developers to interact with databases using Python objects instead of raw SQL queries. This simplifies database operations and enhances code readability.
- **Built-in Admin Interface:** Django comes with a built-in admin interface that can be easily customized. This feature is useful for managing database records and provides a quick way to perform CRUD (Create, Read, Update, Delete) operations without having to build a separate admin panel.
- **Security:** Django includes built-in security features, such as protection against common web vulnerabilities like SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF). It encourages secure coding practices by default.
- **Scalability:** Django is designed to be scalable, and many large-scale websites and applications, such as Instagram and Pinterest, have been built using Django. It provides tools for handling increased traffic and data volume.
- **Versatility:** Django is a versatile framework that supports various types of web applications. It is suitable for building a wide range of projects, regardless of size and complexity.

❖ Pre requisite steps one should follow to create project in Django:

1. Install required version of Django:
   pip install django==(version name) – in command prompt
2. Create virtual environment for the project:
   create folder where the project needs to be saved
3. Activate the virtual environment:

    (folder name)\Scripts\activate – in project folder’s command prompt

4. Start project:
   django-admin startproject (projectname) – in project folder’s command prompt
5. To navigate to project directory:
   cd <space> (projectname) – in project folder’s command prompt
6. To open project file in VS code:
   code <space> .

❖ Following are the steps one should follow to create project in Django:

1. To run migrations:
   python .\manage.py migrate
2. To create superuser (optional):
   python .\manage.py createsuperuser
3. To run the development server:
   python .\manage.py runserver
4. To create application:
   python .\manage.py startapp (app_name)
5. To register your application:
   In the settings file of project folder, add the name of your application in string(‘’) format in the ‘installed app’ section
6. Create folder named ‘templates’ in your application folder
7. Define Models, Views and Templates:
   In the models.py file of your app, define data models
   In the views.py file of your app, define the logic for handling HTTP requests
8. Configure URL’s:
   Include your app’s URL’s in the project’s urls.py file

2. How to check installed version of django?

Ans. Type the following command in the computer’s command prompt to check the installed version of django:
django-admin<space>--version

3. Explain what does django-admin.py makemessages command is used for?

Ans. The django-admin.py makemessages command is used in Django to generate or update translation files for internationalization (i18n) and localization (l10n) of your project. This command scans your Django project's codebase for marked translation strings and creates or updates message files containing these strings. These message files serve as templates for translators to provide translations in different languages.

Here's a breakdown of the command:

- django-admin.py: This is the command-line utility for interacting with various Django management tasks.
- makemessages: This is the specific task or action you're instructing Django to perform. In this case, it's the generation or update of translation message files.

4. What is Django URLs? Make program to create django url.

Ans. In Django, URLs (Uniform Resource Locators) are defined using the urls.py file. This file is responsible for mapping URLs to views within your Django application. It helps Django determine which view function should handle a specific URL pattern.

Here's a basic example of creating a Django URLs configuration:

Assuming you have a Django app named "myapp" with a view called "my_view," follow these steps:

- **Create Django App:**

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

- **Define a view:**

In myapp/views.py:

```
from django.http import HttpResponse
def my_view(request):
   return HttpResponse("Hello, this is my view!")
```

- **Create a URLs Configuration:**

In myapp/urls.py:

```
from django.urls import path
from .views import my_view
urlpatterns = [
   path('my-view/', my_view, name='my_view'),
]
```

- **Include the App URLs in the Project URLs:**

In the main project's urls.py (located in the project's main directory), include the app's URLs like this:

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

```
urlpatterns = [
   path('admin/', admin.site.urls),
   path('myapp/', include('myapp.urls')),  # Include the app's URLs here
]
```

- **Run the Development Server:**

```
python .\manage.py runserver
```

5. What is a QuerySet? Write program to create a new Post object in database.

Ans. A QuerySet is a collection of database queries that can be used to retrieve data from the database. It acts as an intermediary between the database and Django application, allowing us to filter, order, and manipulate data before retrieving it.

- Following are the steps required to create a new Post object in database:
- Within Django app directory, define 'post' model in a file called models.py, as follows:

```
# models.py
from django.db import models
class Post(models.Model):
   title = models.CharField(max_length=100)
   content = models.TextField()
```

- Once model is defined, create a new Post object and save it to the database using the Django ORM (Object-Relational Mapper), as follows:

```
# Import the Post model
from myapp.models import Post
# Create a new Post object
new_post = Post.objects.create(
   title="Example Title",
   content="This is the content of the post.")
# Save the new Post object to the database
new_post.save()
```

6. Mention what command line can be used to load data into Django.

Ans. In Django, you can use the loaddata management command to load data from a fixture file into your database. A fixture file is typically a JSON or XML file containing serialized data for your Django models.

- python manage.py loaddata myapp/fixtures/data.json