**Module 5 – DB and Python Framework**

**Assignment**

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

**Ans –**

* **Rapid Development**: Django's design philosophy emphasizes the "Don't Repeat Yourself" (DRY) principle and includes many built-in features like authentication, admin interface, and an ORM, which speed up the development process.
* **Scalability**: Django is designed to handle high traffic and large volumes of data, making it suitable for both small and large applications.
* **Security**: Django provides strong security features out of the box, such as protection against SQL injection, cross-site scripting, and cross-site request forgery.
* **Versatility**: Django is versatile enough to build a wide range of web applications, from content management systems and e-commerce platforms to social networks and scientific computing platforms.
* **ORM (Object-Relational Mapping)**: Django’s ORM allows developers to interact with databases using Python code instead of SQL, making database operations easier and more intuitive.
* **Community and Documentation**: Django has a large, active community and comprehensive documentation, making it easier to find support and resources.

**Creating a Project in Django**

1. **Install Django**:
   * pip install django
2. **Create a Project**:
   * django-admin startproject projectname
3. **Navigate to Project Directory**:
   * cd projectname
4. **Run Development Server**:
   * python manage.py runserver
   * Visit http://127.0.0.1:8000/
5. **Create an App**:
   * python manage.py startapp appname
6. **Configure App**:
   * Add app to INSTALLED\_APPS in settings.py
7. **Define Models**:
   * Create models in appname/models.py
8. **Create Database Tables**:
   * python manage.py makemigrations
   * python manage.py migrate
9. **Create Views**:
   * Define views in appname/views.py
10. **Map URLs to Views**:
    * Define URLs in appname/urls.py
    * Include app URLs in projectname/urls.py
11. **Run Development Server Again**:
    * python manage.py runserver
    * Test your URLs in the browser

**Qs 2 :- How to check installed version of django?**

**Ans –**

To check the installed version of Django, you can use the following methods:

**Method 1: Using the Command Line**

Open your command line interface (CLI) and run the following command:

**python -m django --version**

**Method 2: Using Python Script**

You can also check the Django version from within a Python script. Create a new Python file (or use an existing one) and add the following code:

**import django**

**print(django.get\_version())**

Run the script using your Python interpreter.

**Method 3: Using Django Shell**

If you have a Django project set up, you can use the Django shell to check the version. First, navigate to your project directory in the command line, then run:

**python manage.py shell**

Once inside the shell, enter the following commands:

**import django**

**django.get\_version()**

Any of these methods will display the currently installed version of Django.

**Qs 3 :- Explain what does django-admin.py make messages command is used for?**

**Ans –**

Sure! The `django-admin.py makemessages` command is used to help translate your Django application into different languages. Here’s a simple explanation:

1. **Find Translatable Text :** It looks through your code for any text marked for translation.

2. **Create Translation Files :** It creates files where you (or translators) can add translations for the text it found.

3. **Update Translation Files** : If you already have translation files, it updates them with any new text that needs translating.

**How to Use It**

**1.** **Mark Text for Translation :** In your code, mark text that should be translated.

**from django.utils.translation import gettext as \_**

**def my\_view(request):**

**message = \_("Hello, world!")**

**return HttpResponse(message)**

**2. Run the Command :** In your terminal, run the command with the language code you want to translate to (e.g., `fr` for French).

**django-admin.py makemessages -l fr**

**3. Add Translations :** Open the created `.po` file and add your translations.

**msgid "Hello, world!"**

**msgstr "Bonjour, le monde!"**

**4. Compile Translations :** Run another command to compile your translations.

**django-admin.py compilemessages**

That's it! Now your app can display text in different languages based on the translations you added.

**Qs 4 :-What is Django URLs?make program to create django urls**

**Ans –**

Django URLs are a way to map different URL patterns to specific views within a Django application. This allows Django to handle web requests and render the appropriate responses. Each URL pattern is associated with a view function that processes the request and returns a response, such as an HTML page, JSON data, or a file.

**Steps to Create Django URLs**

1. **Set up a Django project and app:**
   * Create a Django project.
   * Create a Django app within the project.
2. **Define URLs in the app:**
   * Create a urls.py file in your app directory.
   * Define URL patterns in the urls.py file and map them to view functions.
3. **Include the app's URLs in the project's URLs:**
   * Add the app's URL configuration to the project's urls.py file.