

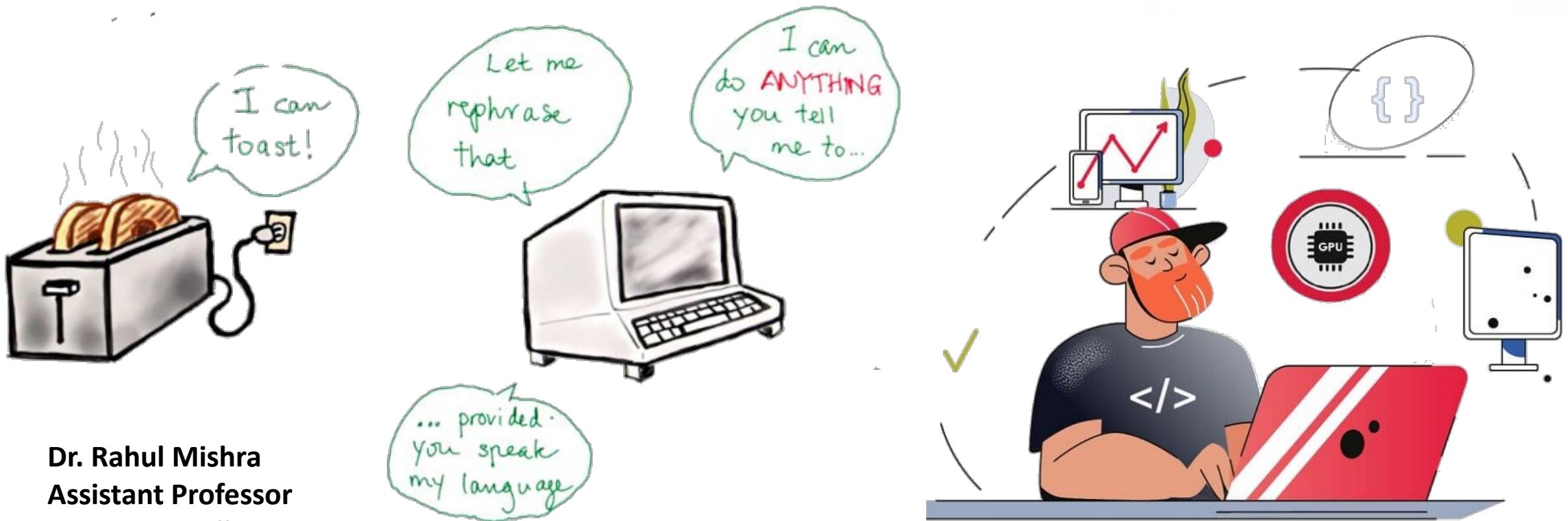


ENGINEERS WITH
SOCIAL RESPONSIBILITY

Programming Lab

Autumn Semester

Course code: PC503



Dr. Rahul Mishra
Assistant Professor
DA-IICT, Gandhinagar



Lecture 24

Django

DJANGO →

- 1) Today's websites are rich applications that act like fully developed desktop applications.
- 2) Python has a great set of tools called Django for building web applications.
- 3) Django is a web framework — a set of tools designed to help you build interactive websites.
- 4) Here, we will learn how to use Django to build a project called Learning Log
- 5) The Learning Log is an online journal system that lets you keep track of information you have learned about particular topics.
- 6) Our target to create a web page using Django which can respond to page requests and make it easier to read and write to a database, manage users, and much more.

► Setting Up a Project:-

- > When begining a project, we first need to describe the project in a specification or spec.
- > Then we will set up a virtual environment in which to build the project.
- * Writing a spec :-
 - » A full spec details the project goals, describe the project's functionality, and discuss its appearance and user interface.
 - » Spec of our project -
 - We will write a web app called Learning Log, that allows users to log the topics they are interested in and to make journal entries as they learn about each topic.
 - The Learning Log home page will describe the site and invite users to either register or log in.
 - Once logged in, a user can create new topics, add new entries, and read and exit existing entries.

► Creating a Virtual Environment :-

- To work with Django, we will first set up virtual environment
- A virtual environment is a place on your system where you can install packages and isolate them from all other Python packages.
- Separating one project's libraries from other projects is beneficial and will be necessary when we deploy Learning Log to a server.

① Create a new directory for your project called "learning_log"

② switch to that directory in a terminal, and enter the following code to create a virtual environment:

```
learning-log$ python -m venv ll-env
```

```
learning-log$
```

- We are running the `venv` environment module and using it to create a virtual environment named `ll-env` (note that this is `ll-env` with two lowercase ls, not two ones). ④
- IF you use a command such as `python3` when running programs or installing packages.

► Activating the Virtual Environment:

```
learning-log$ source ll-env/bin/activate  
(ll-env)learning-log$
```

Note: IF you are using Windows, use the command `ll-env\scripts\activate` (without the word `source`) to activate the virtual environment. If you are using Powershell, you might need to capitalize `Activate`

* To deactivate use the following:-

```
(ll-env)learning-log$ deactivate  
learning-log$
```

▶ Installing Django

```
(ll-env) learning-log$ pip install django
```

Collecting django

-- snip --

Installing collected packages: pytz, django
Successfully installed django

```
(ll-env) learning-log$
```

→ We are working in a virtual environment, which is its own self-contained environment, this command is the same on all systems.

→ There's no need to use the ~~-E~~ ~~useenv~~ flag, and there's no need to use longer commands, such as python -m pip install package_name.

⑥

Creating a Project in Django →

(ll-env) learning-log\$ django-admin startproject learning-log.

(ll-env) learning-log\$ ls

learning-log ll-env manage.py

(ll-env) learning-log\$ ls learning-log

--init-- by setting.py urls.py wsgi.py

Creating the Database →

> Django stores most of the information for a project in a database, so next we need to create a database that Django can work with. ~~ll~~ Use some active environment!

(ll-env) learning-log\$ python manage.py migrate

> operations to perform:

Apply all migrations: admin, auth, contenttypes, sessions

> Running migrations:

Applying contenttypes.0001_initial... OK

Applying auth.0001_initial... OK

--snip--

Applying sessions.0001_initial... OK

(ll-env) learning-log\$ ls #dir

db.sqlite3 learning-log ll-env manage.py

Note: ① Anytime we modify a database, we say we are migrating the database.
② Issuing the migrate command for the first time tells Django to make sure that database matches the current state of the project
③ SQLite is a database that runs off a single file; it's ideal for writing simple apps because you won't have to pay much attention to managing the database.

* In an active virtual environment, use the command python to run manage.py commands, even if you use something different, like for python3, to run the program. ⑧

► Viewing the Project:-

- Let us make sure that Django has set-up the project properly.
- Enter the run server command as follows to view the project in its current state:-

(ll-env) learning-log\$ python manage.py runserver

Watchman unavailable: pywatchman not installed

Watching for file changes with statReloader

Performing system checks...

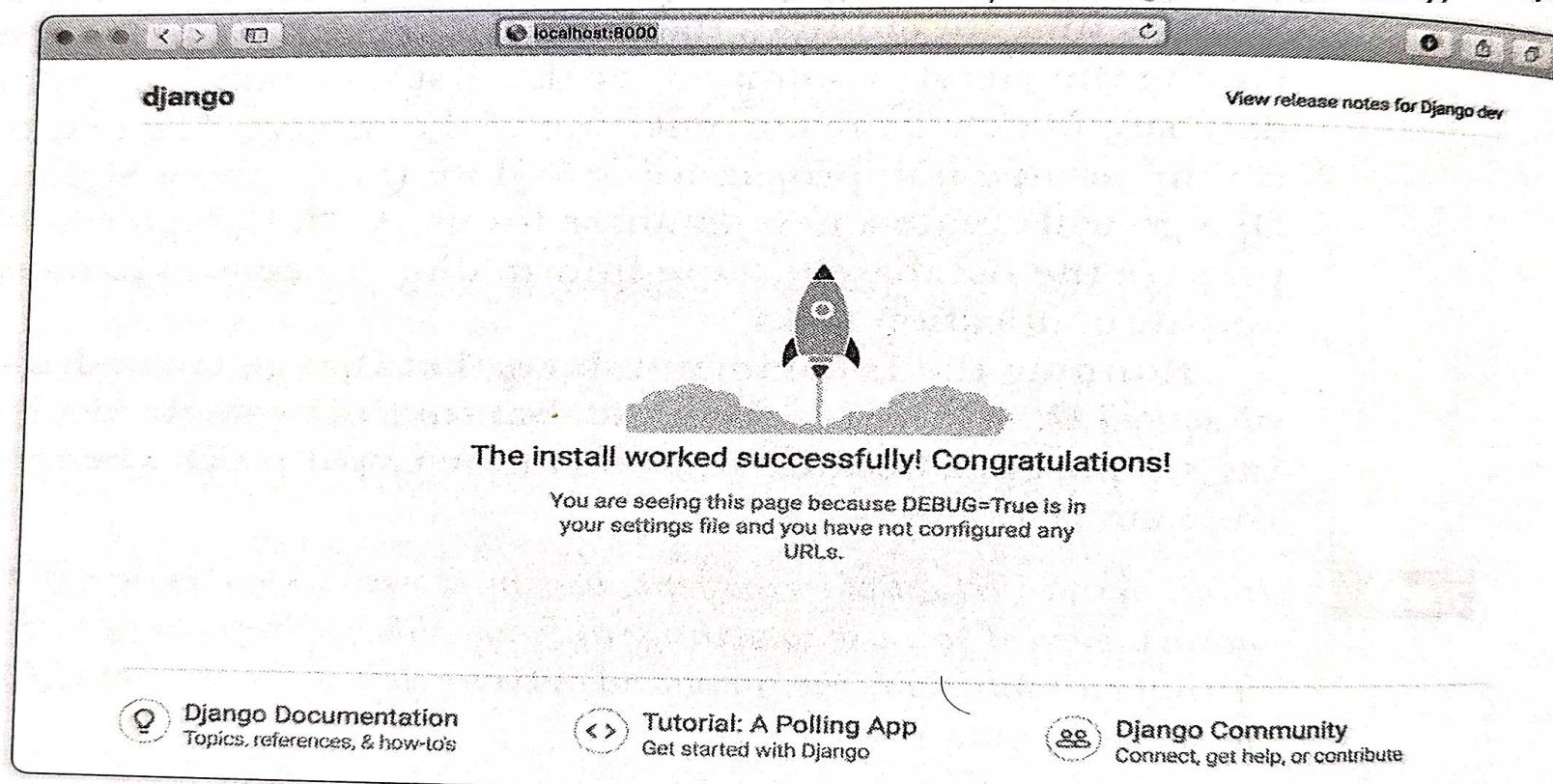
①- System check identified no issue (0 silenced)

②- Django version using settings 'learning-log.settings'

③- Starting development server at <http://127.0.0.1:8000/>

QUIT the server with CONTROL-C

- Django should start a server called the development server, so you view the project on your system to see its work. ⑨
- The URL `http://127.0.0.1:8000/` indicates that the project is listening for requests on port 8000 on your computer, which is local host.
- The term local-host refers to a server that only processes requests on your system.
- Open a web browser and enter the URL `http://localhost:8000/` or `http://127.0.0.1:8000/`



* If you receive the error message That port is already in use, tell Django to use a different port by entering `python manage.py runserver 8001`. 10

► Starting an App

- > A Django project is organized as a group of individual apps that works together to make the project work as a whole.
- > You should leave the development server running in the terminal window opened earlier.
- > Open a new terminal window (ctrl+tab), and navigate to the directory that contains `manage.py`
- > Activate the virtual environment, and then run the startapp command:

```
learning-log$ ll-env\Scripts\activate  # source ll-env/bin/activate  
(ll-env) learning-log$ python manage.py startapp learning-logs  
(ll-env) learning-log$ ls # dir  
db.sqlite3 learning-log learning-logs ll-env manage.py  
(ll-env) learning-log$ ls learning-logs/ # dir learning-logs/
```

- > The command `startapp appname` tells Django to create the infrastructure needed to build an app.
- > When you look in the project directory now, we will see a new folder called `learning-logs`.
- > The most important files are `models.py`, `admin.py`, and `views.py`
- > We will use `model.py` to define the data we want to manage in our app.

* Defining Models :-

- > Each users will need to create a number of topics in their learning log.
- > Each entry they make will be tied to a topic, and these entries will be displayed as text.
- > We will also need to store the timestamp of each entry
- > Open the file [models.py]

```

models.py
    from django.db import models
    class Topic(models.Model):
        "A topic the user is learning about"
        text = models.CharField(max_length=200)
        date_added = models.DateTimeField(auto_now_add=True)
        def __str__(self):
            "Return a string representation of the model."
            return self.text
    
```

→ We have created a class called Topic, which inherits from Model — a parent class included in Django.

* Activating Models -

Open settings.py (in the learning-log / learning-log directory); you will see a section that tell Django which apps are installed and work together in the project.

settings.py

```
-- snip --
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages'
]
```

-- snip --
INSTALLED_APPS = [
 # My apps
 'learning_log',
 ## Default django apps

* We need to tell Django to modify the database so it can store information related to the model Topic. (i)

(ll-env) learning-log \$ python manage.py makemigrations learning-logs

* Now, we will apply this migration and have Django modify the database

(ll-env) learning-log \$ python manage.py migrate

operations to perform:

Applying learning-logs.0001_initial... ok

Django confirms that the migration for learning-logs worked ok.

Whenever we want to modify the data that Learning Log manages, we will follow these three steps:

- > modify `models.py`
- > call `makemigrations` on `LearningLogs`,
- > tell Django to migrate the project.

* \Rightarrow The Django Admin Site

- * Django makes it easy to work with your models through the admin site.
- * Only the site's administrators are users of the admin site, not general users.

* Setting Up a Superuser :-

- Django allows you to create a superuser, or user who has all privileges available on the site