**Complete**

Django follows the "Batteries included" philosophy and provides almost everything developers might want to do "out of the box". Because everything you need is part of the one "product", it all works seamlessly together, follows consistent design principles, and has extensive and [up-to-date documentation](https://docs.djangoproject.com/en/stable/).

**Versatile**

Django can be (and has been) used to build almost any type of website — from content management systems and wikis, through to social networks and news sites. It can work with any client-side framework, and can deliver content in almost any format (including HTML, RSS feeds, JSON, XML, etc).

Internally, while it provides choices for almost any functionality you might want (e.g. several popular databases, templating engines, etc.), it can also be extended to use other components if needed.

**Secure**

Django helps developers avoid many common security mistakes by providing a framework that has been engineered to "do the right things" to protect the website automatically. For example, Django provides a secure way to manage user accounts and passwords, avoiding common mistakes like putting session information in cookies where it is vulnerable (instead cookies just contain a key, and the actual data is stored in the database) or directly storing passwords rather than a password hash.

Django enables protection against many vulnerabilities by default, including SQL injection, cross-site scripting, cross-site request forgery

**Scalable**

Django uses a component-based "[shared-nothing](https://en.wikipedia.org/wiki/Shared_nothing_architecture)" architecture (each part of the architecture is independent of the others, and can hence be replaced or changed if needed). Having a clear separation between the different parts means that it can scale for increased traffic by adding hardware at any level: caching servers, database servers, or application servers. Some of the busiest sites have successfully scaled Django to meet their demands (e.g. Instagram and Disqus, to name just two).

**Maintainable**

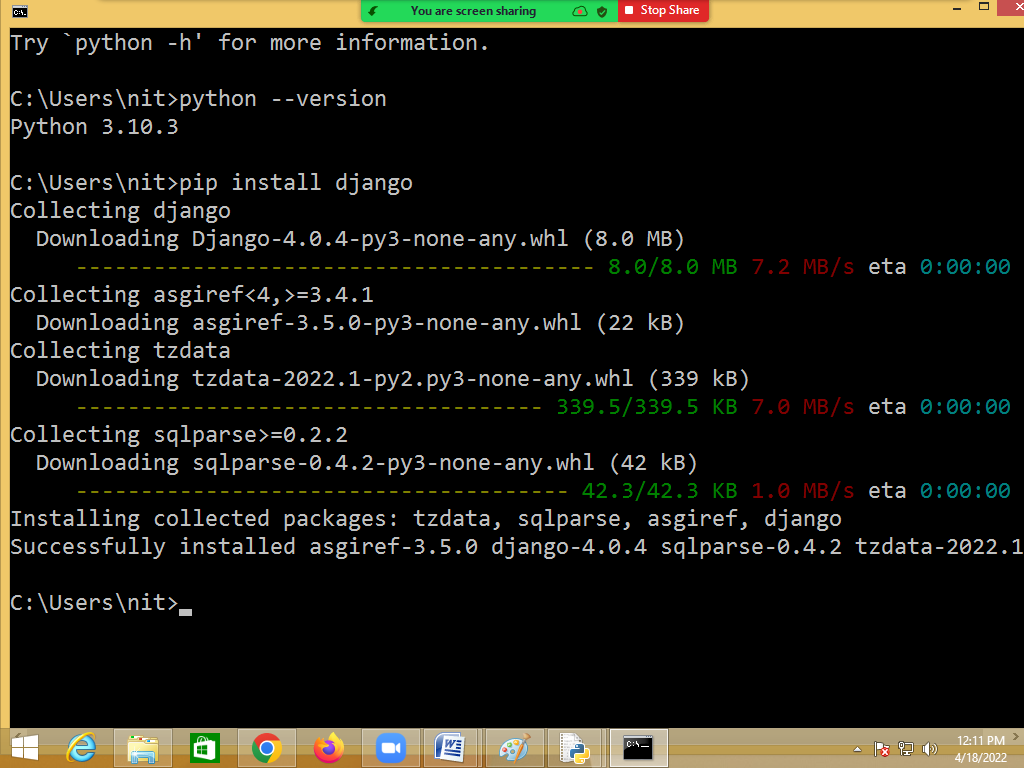
Django code is written using design principles and patterns that encourage the creation of maintainable and reusable code. In particular, it makes use of the Don't Repeat Yourself (DRY) principle so there is no unnecessary duplication, reducing the amount of code. Django also promotes the grouping of related functionality into reusable "applications" and, at a lower level, groups related code into modules (along the lines of the [Model View Controller (MVC)](https://developer.mozilla.org/en-US/docs/Glossary/MVC) pattern).

**Portable**

Django is written in Python, which runs on many platforms. That means that you are not tied to any particular server platform, and can run your applications on many flavors of Linux, Windows, and Mac OS X. Furthermore, Django is well-supported by many web hosting providers, who often provide specific infrastructure and documentation for hosting Django sites.

**Django installation**

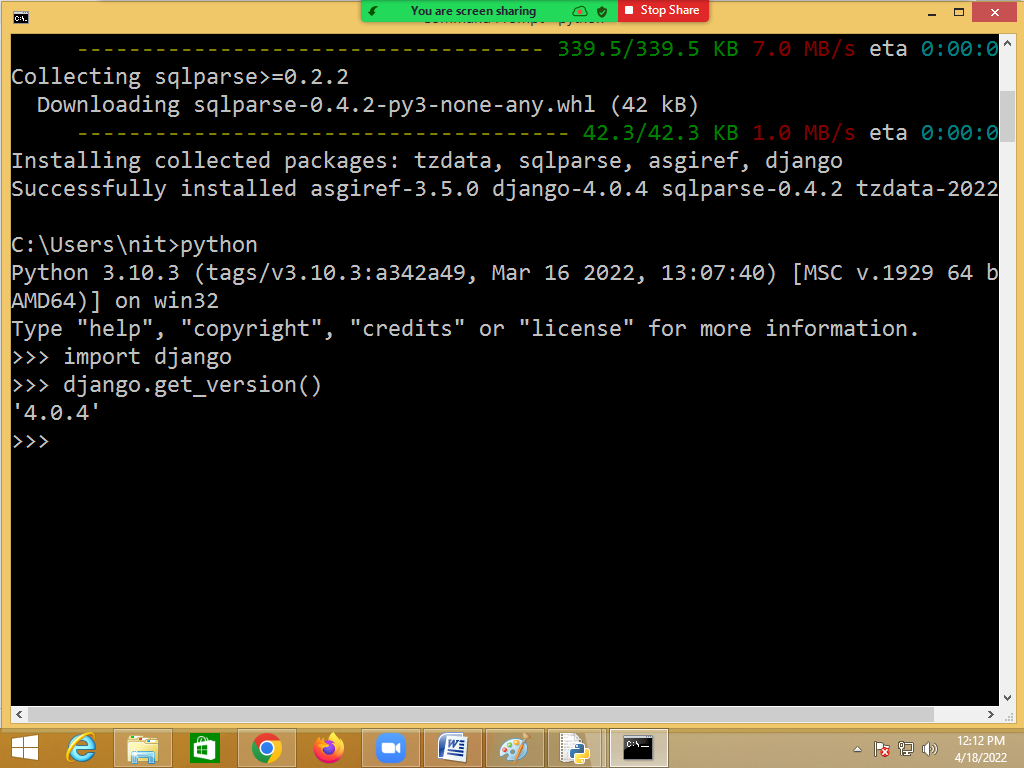
pip install Django



**Verifying Django installation**

>>>import django

>>>django.get\_version()



FAQ: Difference between django project and django app?

**Django project**

A project is Django folder that represents whole web application.

A project in Django consists of configurations and setting related to the entire web site

A single project can have multiple web applications/app.

**Django application or Django app**

An app in Django is a sub-module of a project, and it is used to implement some functionality.

Now, you can refer to an app as a standalone python module that is used to provide some functionality to your project.

We can create multiple apps within a single Django project. And these apps can be independent of one another. Theoretically, we can use an app from one Django project to another without making any changes to it.

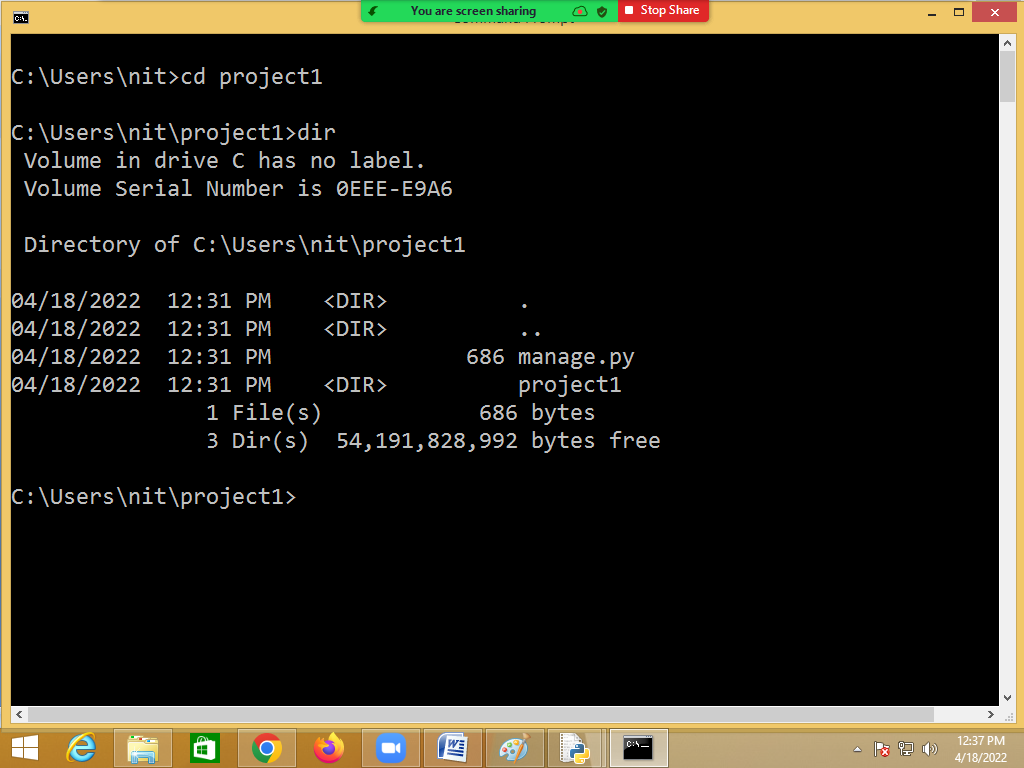
**Creating Django project**

django-admin startproject<project-name>

django-admin startproject project1

django-admin is a command used for creating project.

django-dimin command create a folder with project name



Project is created with,

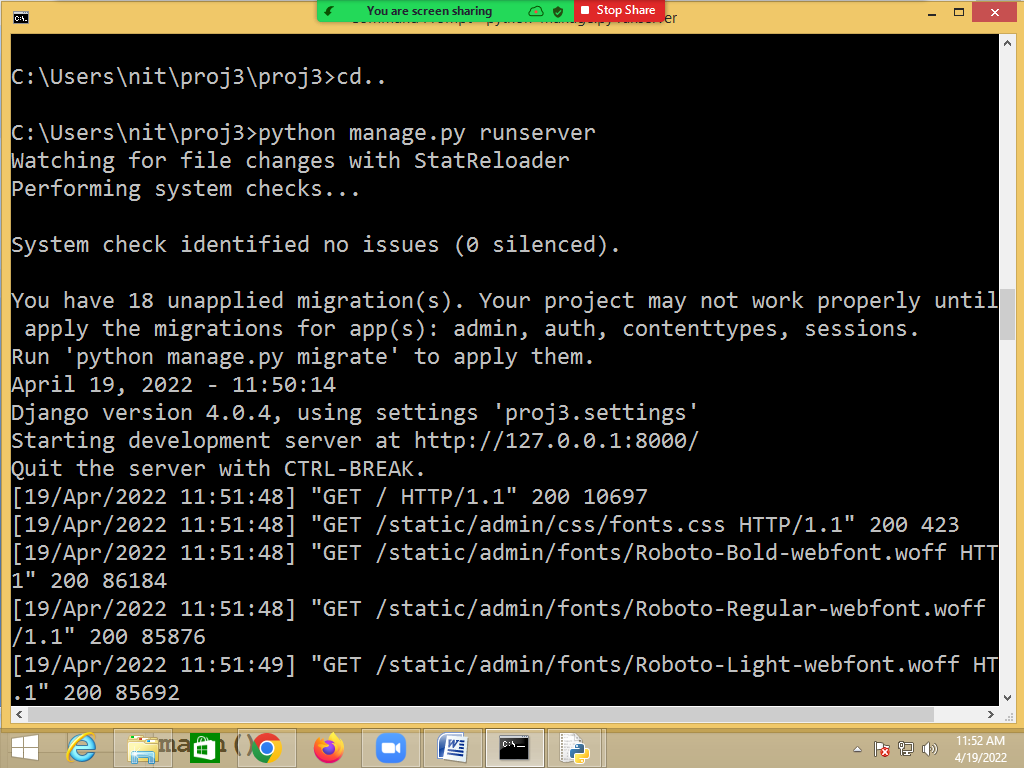
1. mange.py
2. project1 🡪 it is python package created with project name

**Creating Django app**

python manage.py startapp <application-name>

**Running server**

python manage.py runserver



open browser

type the following url

http://127.0.0.1:8000/