**Unit testing tools:**

**Unittest by Django:**

**Advantages:**

1. It helps you to detect bugs early in the development cycle
2. It helps you to write better programs
3. It syncs easily with other testing methods and tools
4. It will have many fewer bugs
5. It is easier to modify in future with very less consequence

**Disadvantages:**

1. Unit testing can't be expected to catch every error in a program. It is not possible to evaluate all execution paths even in the most trivial programs
2. Unit testing by its very nature focuses on a unit of code. Hence it can't catch integration errors or broad system level errors.

**tox:**

**Advantages:**

1. It is automated and easy to standardize testing in python.
2. running your tests in each of the environments, configuring your test tool of choice
3. acting as a frontend to Continuous Integration servers, greatly reducing boilerplate and merging CI and shell-based testing. Useful for Comprehensive Testing.

**Disadvantages:**

1. Uses routing pattern specify its URL.
2. Django is too monolithic.
3. Everything is based on Django ORM.
4. Components get deployed together.

**Docker:**

**Advantages:**

1. Docker is an open-source software containerization platform and is now the most popular alternative to virtual machines in the world. It provides the ability to isolate an application and its related dependencies into a lightweight, self-sufficient container that can be run virtually anywhere. It is easy to understand and find bugs if unit testing is done.
2. Lower resource requirements compared to running a virtual machine.
3. Support from the biggest cloud services.

**Disadvantages:**

1. Can’t say that every part of the system will work correctly.

**The preferred way to write tests in Django is using the unittest module built in to the Python standard library. This is covered in detail in the Writing and running tests document.**

\*\*We will use Unittest .\*\*