**Main contrasts:**

* Flask provides *simplicity*, *flexibility* and *fine-grained control*. It is *unopinionated* (it lets you decide how you want to implement things).
* Django provides an *all-inclusive* experience: you get an *admin panel*, *database interfaces*, an *ORM*, and directory structure for your apps and projects out of the box.

**Flexibility:**

* The batteries that are included in Django help developers to create a variety of web applications without using any third-party tools and libraries. But the developers lack the options to make changes to the modules provided by Django.
* Flask is a small but extensible web framework enabling developers in developing web applications with flexibility using various web development tools and libraries. Many developers find it easier to learn Flask than Django due to its smooth and customizable architecture.

**Admin System:**

* Django has an admin system that comes with an ORM (Object Relational Mapper) database system and directory structure. Developers feel that it is an all-inclusive experience when it comes to developing with Django meaning that multiple projects have the same directory structure.
* Flask, on the other hand, does not have these features, If you want to have an admin system or use an ORM, you’ll need to install custom modules. The Flask framework leaves this it up to the developer. It gives them the option to use it with SQLAlchemy, MongoDB or something more simple like SQLite. This can be a preferable choice as with ORM you might sometimes waste development time if you are unable to modify the SQL query directly.

**Development Speed:**

* Django frameworks are known amongst developers for their fast development of complex web applications. Since it is fully featured; developers have all the tools that they require to implement and develop easily reliable, scalable, and maintainable web applications, in record time
* On the other hand, Flask’s simplicity allows experienced developers to create smaller applications in short timeframes.

**Template Engine:**

* Django is with a built-in template engine that enables developers to define the web application’s user-facing layer without putting extra time and effort. It even allows developers to stimulate custom user interface development by writing templates in Django template language (DTL).
* Flask is developed based on the Jinja2 template engine. It enables developers to stimulate the development of dynamic web applications by taking help of an integrated sandboxed environment and writing templates in an expressive language.

Built in Bootstrapping:

* Django has a built-in bootstrapping tool called Django-admin. It enables developers to build web applications without any external input and to divide a single project into multiple numbers of applications.

They take different approaches to design a web application and create a REST-based API design. Django framework uses its own built-in user-model which facilitates API authorization and authentication. It alone can be used to create RESTful API. Flask, on the other hand, comes up with the basic set of tools for authorization or third-party plugin such as Flask HTTP-Auth.