Technical Issue (Encryption and Security):

The overarching lesson we learned from researching encryption as it pertains to web application security is to never trust any data from untrusted sources. A feature of Django that proves to be incredibly beneficial is a low-level API that is used for cryptographically signing values. Additionally, Django provides a high-level API that sets and reads signed cookies, used often in web applications.

Django uses these two APIs in a complementary fashion. One auto-generated feature of new projects in Django, the SECRET_KEY value found in settings.py, is the first step in various signing procedures. Besides run of the mill signing. Django also provides other ways of authenticating security including its ability to verify timestamped data and its ability to protect complex data structures through dumps and loads functions in signing modules.

These tools allow for Django users like ourselves to be able to generate security and encryption protocols that are relatively complex and are able to thwart off a large number of malicious security attacks.

Sources:

https://docs.djangoproject.com/en/1.10/topics/signing/

http://www.tylerlesmann.com/2008/dec/19/encrypting-database-data-django/