**Django Report**

* **What does this technology (library/framework/service) accomplish for you?**
  + **Explain what this technology does in your project. What problems does it solve for you?**

Django handles any http requests, authentication, and communication with our docker’s mysql database. Django also fills in templates with the specified content.

* **How does this technology accomplish what it does?**
  + **For each problem this technology solves for you, explain in detail how it solves it starting with a TCP socket connection. You must include links to any relevant documentation or code. If the technology is open-source, you must include links to the exact code that solves your problem for you.**
  + **Ex. If you use express for your web framework you must explain how express handles an HTTP request starting with a TCP connection to include how it routes requests, how headers are parsed, etc. and link to the exact code on GitHub that parses HTTP headers and routes paths. Any other express features used must also be explained and the code linked. You do not have to explain features of the library that are not used in your project.**

HTTP requests and responses

* How Django receives HttpRequests
  + Nginx listens on port 80 for requests and when it receives one, it passes it to uwsgi on the app through port 8000 with an environ dict containing the request data and a start response function. Then, uwsgi, through wsgihandler starts up a limited stream which reads the request data based on the content length var in the environ dict.
    - <https://github.com/django/django/blob/205c36b58fed5a1a0ff462593fc61b58189027d8/django/core/handlers/wsgi.py#L23>
  + The wsgihandler then uses the basehandler to load up the middlewares to process the request. After the request has been processed and a response has been created, the response function from nginx returns everything to nginx which sends it back to the client.
    - <https://github.com/django/django/blob/205c36b58fed5a1a0ff462593fc61b58189027d8/django/core/handlers/base.py#L20>
* Middlewares
  + <https://data-flair.training/blogs/django-request-response-cycle/>
  + Requests gets processed by a list of middlewares, and if none of them deems the request bad then it gets sent to Django’s url dispatcher
    - <https://github.com/django/django/blob/main/django/middleware/security.py> [Checks security]
    - <https://github.com/django/django/blob/main/django/middleware/common.py> [Basic Operations (e.g. extract path)]
    - <https://github.com/django/django/blob/main/django/middleware/csrf.py> [Checks csrf token]
    - <https://github.com/django/django/blob/main/django/middleware/clickjacking.py> [Checks for clickjacking]
    - <https://github.com/django/django/blob/main/django/contrib/sessions/middleware.py> [Session]
    - <https://github.com/django/django/blob/main/django/contrib/auth/middleware.py> [Authentication]
* Authentication
  + The middleware (link to code shown above) authenticates the user automatically if he/she had logged in before (checks with django/contrib/sessions/middleware.py). Otherwise, he/she will have to login again which will add the user’s profile to the middleware sessions.
  + During registration, passwords are also validated through django’s validators
    - <https://docs.djangoproject.com/en/3.1/topics/auth/passwords/#password-validation>
* Dango’s URL dispatcher
  + <https://docs.djangoproject.com/en/3.1/topics/http/urls/>
  + Extracts the url from the path, and attempts to match it with a url in a predefined list of urlpatterns. If any matches, django calls the assigned view/function for that, and passes an instance of the HttpRequest object as an argument. Each view is responsible for returning an HttpResponse object. Also, if no urls are matched or an error occurs, django invokes the appropriate error-handling view.
    - HttpRequest object handles dissecting the http request into usable parts (splitting, etc.)
    - <https://docs.djangoproject.com/en/3.1/ref/request-response/#django.http.HttpRequest> [Object that contains metadata about request]
    - <https://docs.djangoproject.com/en/3.1/ref/request-response/#django.http.HttpResponse> [Object for Http Response]

Database Communication

* Database connections are managed through django models
  + <https://github.com/django/django/tree/main/django/db/models>
* When the model is created, it is mapped to the sql database and a table is created through django’s sql compiler. Queries are handled by query.py in the django model. Of course, the database itself had been created through docker.
  + <https://github.com/django/django/blob/main/django/db/models/sql/compiler.py>
  + <https://github.com/django/django/blob/main/django/db/models/sql/query.py>

Templates

* Replaces html in template with content that's passed as an argument
  + <https://github.com/django/django/blob/main/django/shortcuts.py>
* **What license(s) or terms of service apply to this technology?**
  + **When using technology you did not write you will be subject to certain agreements. List any licences attached to the project (Ex. MIT, Apache) and what that license means for your project. If using a service, review the terms of service and explain all the implications of these terms for your project.**

Django License Agreement: <https://github.com/django/django/blob/main/LICENSE>

Redistribution and use, with or without modification, is permitted only if this copyright notice is included. The name Django and its contributors cannot be used for endorsement or promotional purposes without written permission. The copyright owner and contributors are also not responsible for any damages involving Django.

There aren’t many terms, but essentially, people can use django for whatever they want except for endorsements or promotion without permission. This seems to just be to protect Django and its contributors from any liabilities.

.

Django Trademark License Agreement: <https://www.djangoproject.com/trademarks/>

Similar to the previous agreement, this is just to protect the trademark Django from being used without permission. The name could be used for educational purposes but some commercial uses, but definitely not in the name of a company or product. However, with a proper license from Django, special use may be allowed.