Server

Host

The server runs Ubuntu on the host sddec24-22.ece.iastate.edu or private IP 10.29.162.58. To connect remotely, connect to ISU's private network and SSH vm-user@sddec24-22.ece.iastate.edu.

CI/CD

CI/CD runs on every change to the main branch in GitLab. There are two runners set up on the server named Django and React. The runners call two system services, "system-django" and "system-react" that start and keep the applications running. These services can be monitored and altered using the systemctl commands. The services call their respective bash file in the server's app folder that runs all necessary commands to start the applications.

Django

Django Commands

To start the application, first start the local virtual environment based on the local OS. Then, use the command "python manage.py runserver" to run Django on localhost:8000.

To update SQL tables based on model changes, use 'python manage.py make migrations', and then 'python manage.py migrate'.

Django Dependencies

The dependencies for Python are stored in a virtual environment for the local machine. The virtual environment must be activated to run Django and separate the dependencies from other applications. The dependencies are not pushed to GitLab as they are different depending on the OS of the machine and may require different setups.

This link runs a demo to build a virtual environment and the necessary commands to run applications. "setup-virtual-environments-in-python/."

Dependency List

1. Django

Django is a requirement for the application to execute and provides the basis for functionality.

2. Mysqlclient

Mysqlclient provides easy connections with the MySQL server. Models made within Django can be automatically migrated to MySQL tables and relationships. It also simplifies querying the database with Python functions. The functions are called on the model objects, which automatically builds a database connection and queries once the data is needed. Commands to migrate the models are 'python manage.py make migrations', which creates the SQL script, and 'python manage.py migrate', which executes the script against MySQL.

3. Djangorestframework

This framework simplifies the creation of REST APIs and separates the functionality into multiple classes. The first class is route.py, which stores all REST API routes for the application. The second is view.py, the REST API functions that are called and routed to. These routes can provide and update data from user requests. The final class is serializer.py, where serializers are made for each model in models.py. These serializers allow query sets and models to be converted into native Python datatypes, easily converted into JSON.

4. Django-cors-headers

CORS provides security for the Django application. It dictates what requests can reach the application depending on their origin.

5. Channels

Channels provides WebSocket development in Django. It provides built-in functions that simplify connections and decrease the complexity. All WebScoket routes are created in the routing.py class. These WebSocket requests are routed to channel consumer classes. Consumer classes handle connections between users and output data through the channel.

6. Daphne

Daphne is an ASGI server for Django that handles incoming requests. An ASGI server provides asynchronous tasking and gives pre-built functions that simplify sync development. Daphne also simplifies connection request handling and provides asynchronous tasking.

6. Python-dotenv

Dotenv simplifies the creation of environment variables. It requires having one .env file that stores a list of all variables that Django can load. Given the variable values, Django loads the .env file in its manage.py class to start the server. This separates development and production environments and provides security for hidden information, including keys.

React

Node Version

20.9.0

React Commands

To download dependencies, run “npm install” and “npm start” to run the application.