

## Foodiezz App

To build a fully featured Web App from the command line app, I will evaluate the requirements that are deemed necessary. From my learning on the job, I think a web app should have a user authentication module. It needs to be responsive and fast while fetching and displaying data. And it goes without saying that it needs to be scalable. With these things in mind, I will start by structuring my app in 2 main blocks. 1. Backend REST API block and 2. Frontend block. By dividing in 2 blocks, it will help us decouple the frontend (or client app) from the backend. We can develop any kind of frontend (like mobile or desktop app) while using the same backend. I will start by using Django and django-rest-framework for the backend. The reason I am choosing Django is because it is a fully featured web framework which provides modules like user authentication and authorization along with other core modules out of the box. Django also is an ORM which will help develop and implement SQL queries quickly. Django-rest-framework will help me develop the REST APIs for interacting with the data. For the frontend, I will go for frameworks like Angular or React. They both are Single Page Application frameworks for frontend. Because of this the app will be fast and responsive on the client's browser.

Other main components of the architecture that I will focus on will be load balancing and caching of data. These things will help us in scaling the app and making sure it is always available when the load increases. For load balancing, I will use Nginx as it provides a lot of flexibility in designing how the web traffic moves from the client app to the web servers. For caching, I will use Redis cache. I will store the Food truck data in Redis with an expiry of 30 mins. This way the web server will not have to make multiple requests to the SoDA API when multiple users are requesting the data at the same time and can respond back to the client quickly. I am choosing this specific time interval for the expiry because looking at response data I found that the start times of the food trucks differ at max by 30 mins. We can change this expiry to be more granular if the data changes in future.

Last component of the architecture would be how quickly and easily we can deploy the app. For this I will use Docker for containerization of the app and Jenkins as my build and deploy tool. Docker is a powerful tool and allows me to run the app in any production OS. For deployment, I will choose a distributed cloud platform like AWS or Microsoft Azure as it helps us to achieve dynamic scalability based on the load on the web app.