

# **Remote interview: Web**

## **Project Communication (IMPORTANT READ AND ACKNOWLEDGE!!!)**

- **Before coming to the interview**
  - Please prepare your Django development environment
  - Read through the Project Description and think about it
  - If you are new to Django, we would advise that you read/try the tutorials prior to the interview
    - <https://docs.djangoproject.com/en/3.1/intro/tutorial01/>
    - <https://www.django-rest-framework.org/>
    - <https://jpadilla.github.io/django-rest-framework-jwt/>
  - Please be prepared to share your screen to code in Django
    - Coding before arriving for the interview is optional
- **During the remote interview**
  - (First hour - up to 1 hour) You would be able to further clarify the project specifications and expectations with your assigned project reviewer and to code on your own.
    - No requirement to share screen
  - (Next 1.5 hours) Project review and pair programming
    - Requirement to share screen
    - Where you will continue writing code, while your assigned reviewer reviews and advises
  - Possibly further rounds of interview
- Remember, your ability to communicate well is as important to us as your technical skills. Both are vital to your success!!

## **Development Requirements**

- **Must use**
  - Python 3
  - Django 3+
  - Django REST Framework with djangorestframework-jwt
  - MySQL (or SQLite)

## **Project Description / Requirements**

- **Relevant concepts**
  - Database Design
  - REST API Design
  - Django
    - Django ORM and Admin
    - RESTful using Django REST Framework
    - Automated testing
    - Relational Databases (only MySQL or SQLite)

- Task 1: Based on the provided usage scenario, design database schema, design REST API and enable Django Admin
  - Usage scenario
    - “We currently provide food delivery services for 2 restaurants. One of them is a burger restaurant, the other is a fried chicken restaurant.”
    - “The burger restaurant requires that the customer specify the (number of burgers), (meat type - ‘beef’, ‘chicken’, fish’) and (comments).”
    - “The fried chicken restaurant requires that the customer specify the (number of chicken parts ordered), (chicken flavor - ‘original’ or ‘extra crispy’) and (comments).”
    - “Our food delivery app is to support many kinds of restaurants. Hence please design the backend and the REST calls to be scalable.”
    - “Our app needs to be able to create an order”
    - “Our app needs to be able to retrieve all orders from all restaurants sorted by created date” (for delivery team)
    - “Our app needs to be able to retrieve all orders for a specific customer” (for customer records)
    - “Our app needs to be able to retrieve a specific order”
    - “Our app needs to display the delivery cost for all orders for a specific restaurant”
    - “Our app needs to display the average number of burgers ordered by a customer for the burger restaurant”
  - Design REST API
    - Please create REST calls required to support usage scenario above
    - The purpose of these web-services is to provide an end-point for a mobile-based client to interact with the backend web server
    - For authentication, please use the package: <https://github.com/GetBlimp/django-rest-framework-jwt>
    - For rest calls, please use package: <http://www.django-rest-framework.org/>
  - Enable Django Admin for all models
    - Ensure that list pages in Django Admin are useful
- Task 2: Future Plans
  - Present on what improvements can be performed in future
- Task 3(Optional): Present an automated test
  - Present **ONE** meaningful automated test testing one of the rest calls in Task 1
- The goal here is to evaluate your technical ability as well as effective communication skills in a real-world environment

**During the pair programming portion - Here's some sample topics to cover (feel free to follow and/or add more):**

- Tell us, in a concise manner, how the usage scenarios are met.
- Database schema: (in the form of Django Models or class diagrams)
- Code Architecture
- REST API (code; requests and expected outputs)