**Project Information**

**Title: CaterGeek**

**Members:**

* **Akshay –** **akshayjain1996**
* **Sneh – snehpatel3**
* **Priyen – c5patelp**
* **Siddharth – Sidd0107**

**Description:**

**CaterGeek is a website where users’ food requirements for events or celebrations, or anything are met. Users who have a need for food can sign up and check out profiles’ of caterers. Different caterers are capable of different things and once a user finds a good caterer, they can make an order related to their needs. A user is able to sort through the listings by rating of the caterer, specialties, and more.**

**A User can also activate a caterer profile after they sign up, should they want to be a caterer. Once a caterer profile is activated, they can list their specialties, they will have the opportunity to pickup/accept/refuse/complete orders. We leave the financial details up to the caterer/user.**

**Users can leave reviews of caterers and caterers maintain a satisfaction rating and list of reviews.**

**Detailed Design:**

* Front-end
  + Blank
  + Blank
* Back-end
  + Blank
  + blank

**Security Testing:**

* Attempting to use RESTApi without appropriate permissions
  + As a normal user: Client is forbidden
    - Only Front-end code will have access to back-end calls for relevant data
  + As a non-authenticated user: Client is forbidden
  + As admin: works
* Attempting to access admin stats related pages
  + As a normal user: Client is forbidden
  + As a non-authenticated user: Client is forbidden
  + As admin: works
* My

**Performance Optimizations:**

* **How did we improve site performance?**
* **We need graphs of before and after**

**Video Demo:**

**Testing:**

We use a testing framework Locust (<http://locust.io/>) to throughly test our app. We test all our routes to make sure they are working at all times. The testing code is in the file locust.py. Here is a screenshot of the test we ran:

