# **Programming Challenge: Secret Santa**

To be honest the way I started to tackle the problem was straight forward – Google

The problem was very common and I thought I would easily get it online. My thought process was why to reinvent the wheel if somebody has already done it for you.

I selected Python as my base language as I have been recently working a lot with Python and I visualised it as a simple python script to get it working

I did find a simple secret santa program - <https://medium.com/analytics-vidhya/i-made-a-contactless-secret-santa-algorithm-with-python-7374d4a79c56> by Monique Cheng which automatically solved by Part 1 Challenge

The input to the application can be done via 2 methods

1. Load a Csv file which stores value as

{name} |{email\_id} | {family\_members\_comma\_seperated}

1. Direct input to the program

Part 2 and 3 was where I actually applied my brain and skills.

For Part 2 of the challenge I went with a simple approach of storing the history in a Dictionary per user using a simple file created through Python. Named it SantaHistory.txt. If the users change, the history has to be cleared before having a re-run.

The output is stored in an output file named as SantaAllocations.txt for easy viewing and even mail can be sent to the respective users.

On repetitive runs the dictionary is appended with names. On the fourth run the first name corresponding to the user is cleared and a new entry is added to the end thus maintaining the history for 3 yrs.

Part 3 of the challenge has been tackled by getting the family information directly at the start or via txt file.

And similar logic like the history check was applied to avoid family names too.

Scope of Improvement :

1. More edge cases could be found and covered
2. Backend integration with a DB
3. Unit Tests
4. Cleaner and leaner code

Scaling :

My thought process of scaling the application is to provide a frontend UI page to capture the user session so that multiple users can use the application at the same time and having the history and family info per logged in user. connecting the script to a relational database which would store the user session info and the corresponding history and family information.

Production Ready :

The application can be deployed to any cloud server like AWS, GCP, Azure, Heroku.

Having good hands-on with AWS I would suggest to deploy the application on a AWS Lambda or AWS Glue where the api to get/fetch/store information can be deployed on AWS Gateway through which the UI can be connected.

We could also containerize it on Docker for sharing it with others for local use or deploying it.