

Secret Santa Game

Background:

Company "Acme" has decided to organize a Secret Santa event among its employees. Each employee is required to choose another employee as their secret child, to whom they will anonymously give a gift during the event. The company wants to automate the process of assigning secret children to employees based on the provided employee information. However, the Secret Santa system has some additional requirements and constraints.

Requirements:

1. **Input Format:** The input will be given as a CSV file containing employee information. Each row in the CSV file will represent an employee and will include the following fields:
 - Employee_Name: The name of the employee.
 - Employee_EmailID: The email ID of the employee.
2. **Secret Santa Game:** Your task is to create a program that parses the provided CSV file and assigns a secret child to each employee, following these rules:
 - An employee cannot choose themselves as their secret child.
 - An employee cannot be assigned to the same secret child as in the previous year's Secret Santa event, if applicable.
 - Each employee must have exactly one secret child.
 - Each secret child should be assigned to only one employee.
3. **Output Format:** The program should generate a new CSV file as output, containing the following fields for each employee:
 - Employee_Name: The name of the employee.
 - Employee_EmailID: The email ID of the employee.
 - Secret_Child_Name: The name of the assigned secret child.
 - Secret_Child_EmailID: The email ID of the assigned secret child.
4. **Constraints:** The previous year's Secret Santa assignments will be provided as input for avoiding repetitive assignments. The assignments will be given as a CSV file containing the fields: Employee_Name, Employee_EmailID, Secret_Child_Name, Secret_Child_EmailID.

5. **Solution Expectations:**

a. **Modularity and Extensibility:** The solution should be modular and extensible, following best practices and principles of object-oriented programming (OOP). It should not be a monolithic codebase; It should not be a single class/function which contains every logic needed to solve this "current" problem

b. **Tests:** Including tests for the solution is needed to demonstrate code quality, correctness, and reliability. This is an added advantage for submission.

c. **Error Handling:** Implement appropriate error handling mechanisms to handle potential exceptions, such as invalid input or file handling issues.

d. **Documentation:** Provide comprehensive documentation, including a README file, that explains the solution, instructions for installation, running the program, and any additional information required for understanding and using the solution.

e. **Version Control:** The submission should be done via a version control system like GitHub, GitLab, or Bitbucket. Solutions sent as zip files or Google Drive links will not be accepted.

Note: The focus is on designing an efficient and reliable Secret Santa game.