Introduction ：

This is an employee messaging web application. It is built using Python Flask and has artificial intelligence capabilities integrated by OpenAI's GPT-3 API. The application is designed for HR managers and those involved in employee management, providing a convenient and effective way to track employee information and gain insight into their needs and concerns.

Main Features,Installation, and implementation：

https://github.com/wangleiz166/cs551n-lw-assessment/blob/main/readme.md

Database Design：

The application uses an SQLite database to store employee information data.

1. The "staff_information" table contains basic information about employees, such as employee ID, age, attrition status, department, education level, job role, and marital status. Total 3480 data items.

2. The "staff_privacy_information" table contains sensitive personal information of employees, such as business travel history, distance from home to work, number of employees in the company, environment, job satisfaction, work-life balance, and performance rating. This information requires strict confidentiality and management. Total 3480 data items.

Front-end design：

The application is designed using the Bootstrap front-end framework and includes the following pages:

1.index.html：For displaying the list of employees.

2.detail.html：For displaying employee details.

3.chat_detail.html：For displaying feedback results from chatbots.

Back-end design：

The application is designed using Python's Flask framework and includes the following components:

1. Flask applications：Used to process HTTP requests and responses.

2.SQLite Database：Used to store employee information and employee privacy information data.

3.OpenAI Library：For natural language processing and generating intelligent chat logs.

Development：

In developing the staff project, we completed the following tasks.

1. Write the front-end code. Designed the staff information display page and the chat tool page using the Bootstrap front-end framework. Embedded the data contained within the template. And implement the paging front-end logic.

2. Implement the back-end logic. Imported the CSV data into the db file and used the Flask framework to implement the back-end logic, connected to the SQLite database to display the data, and implement the paging back-end logic. Including handling HTTP requests and responses, and calling the OpenAI API for AI interaction. Write test cases and perform testing.

Use:

1. To access the staff list, open your web browser and navigate to the URL where the staff project is hosted. This will bring you to the home page, which displays a list of all employees.

2. Click on the "Details" button next to an employee's name to view their detailed information, including age, gender, job role, job satisfaction, and more.

3. To analyze an employee's information using the GPT-3 chatbot, click the "Analysis" button on the employee details page. The chatbot will provide personalized advice based on the employee's information, helping managers identify and address issues related to employee satisfaction, productivity, and retention.

4. To return to the staff list, click the "Return" button at the bottom of the employee details page.