## Final Exam: **Project Outline with Docker for 100 Points**

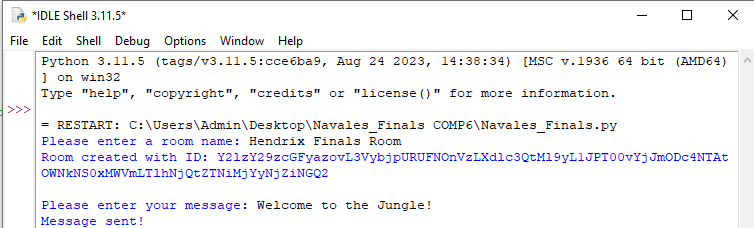
### ****Total Structure and Points****

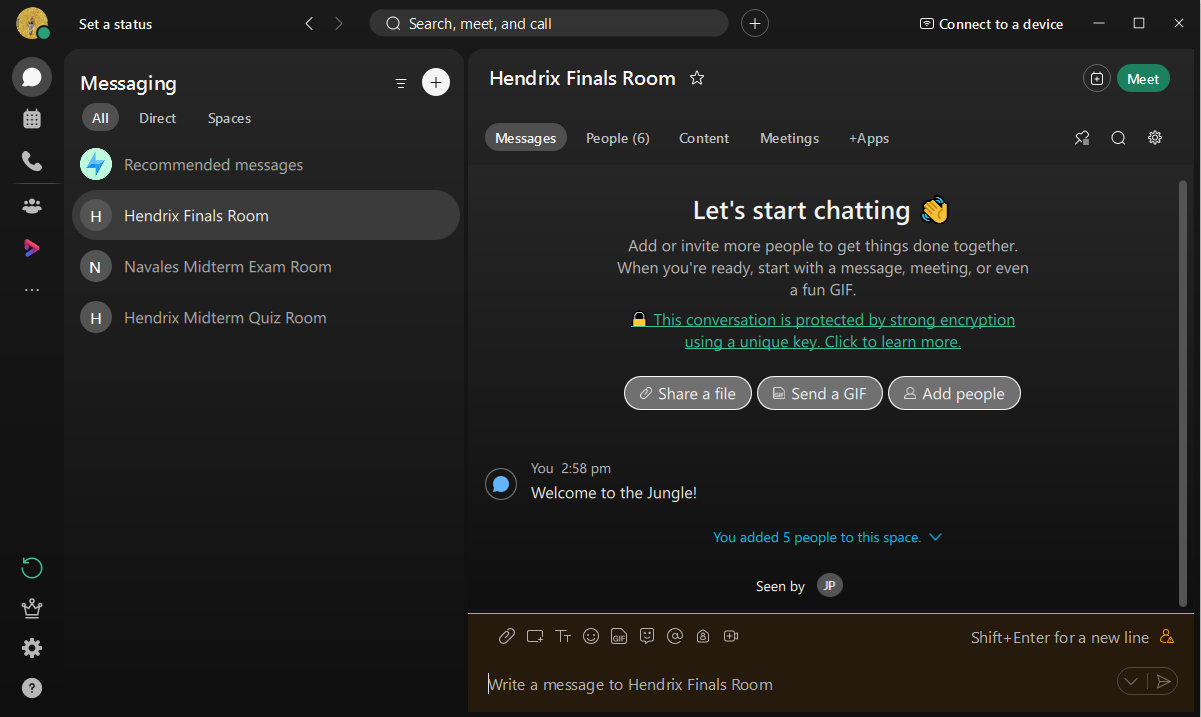
1. **Part 1: Webex Teams API Integration (30 Points)**
2. **Part 2: Faker Data Generation (20 Points)**
3. **Part 3: Docker Setup and Integration (20 Points)**
4. **Part 4: GitHub Documentation and Submission (30 Points)**

### ****Detailed Breakdown****

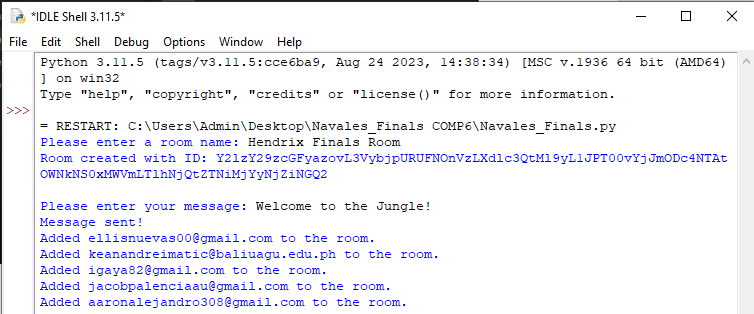
### ****Part 1: Webex Teams API Integration (30 Points)****

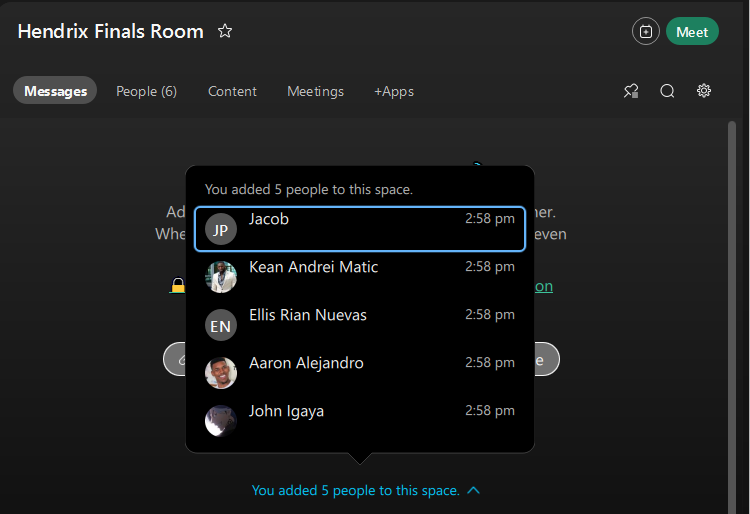
1. **Create a Room and Send a Message (10 Points)**
   * Create a script that uses webexteamssdk to create a room in Webex and send a welcome message.
   * Take user inputs for the room title and the message content.



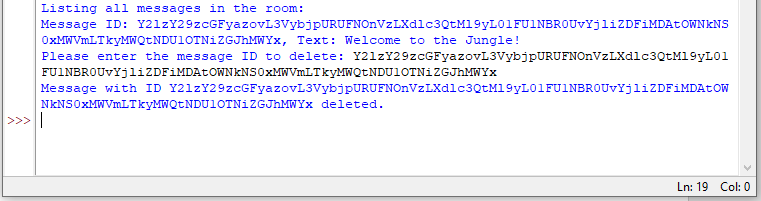


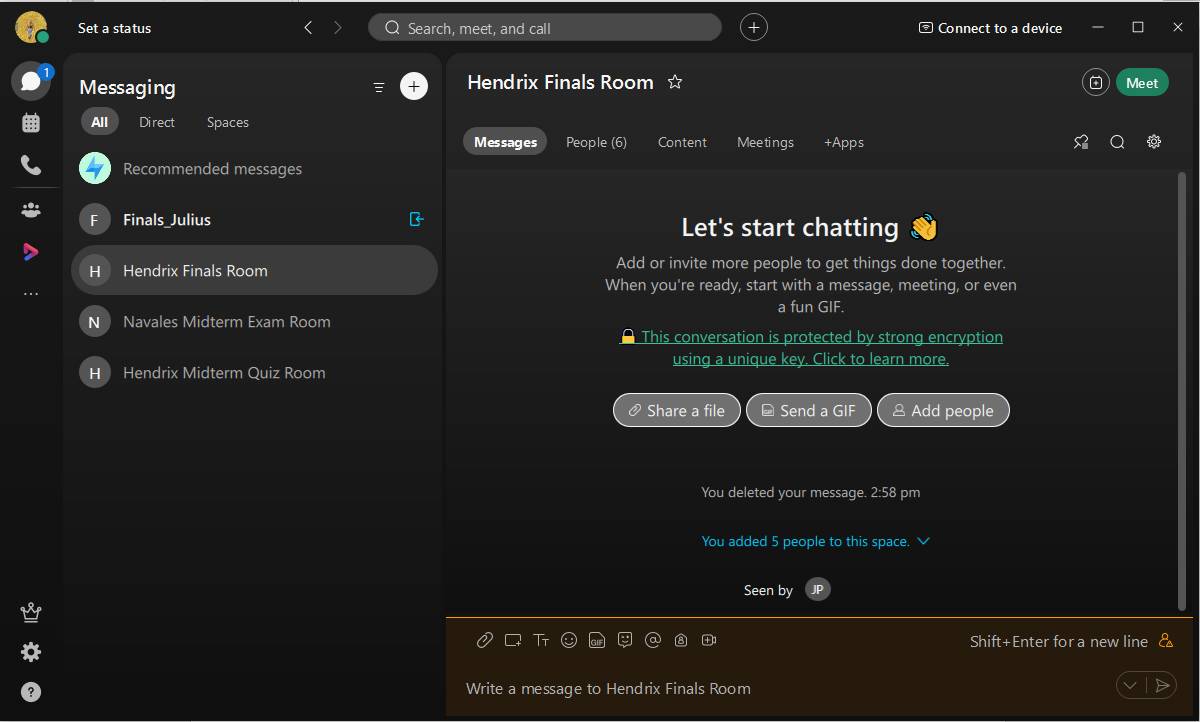
1. **Add Multiple Participants to the Room (10 Points)**
   * Extend the script to include a function that reads participant email addresses from a predefined list and adds them to the room.





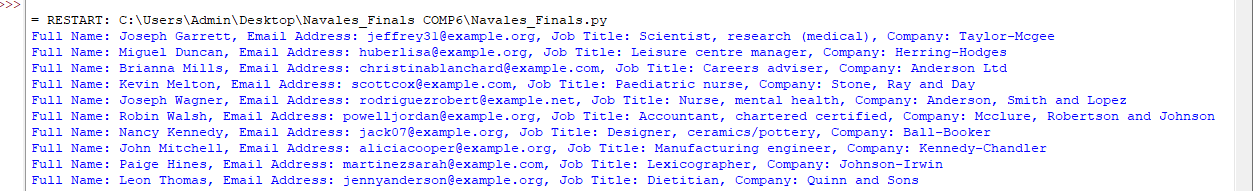
1. **List Room Messages and Delete a Specific Message (10 Points)**
   * Create a function to list all messages in the room with their message IDs.
   * Allow the user to delete a message by providing its message ID.

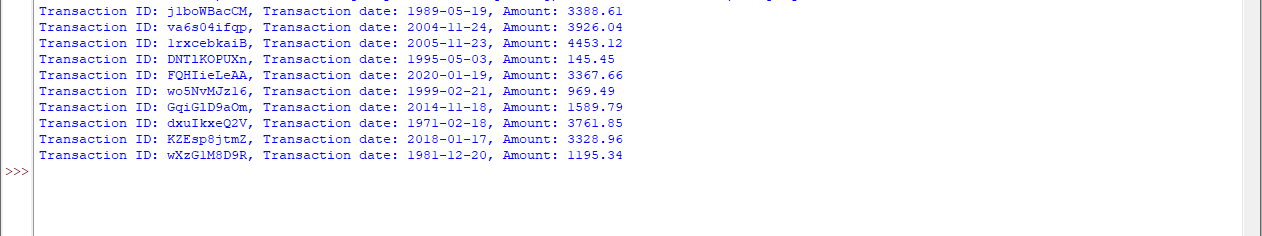




### ****Part 2: Faker Data Generation (20 Points)****

1. **Generate Fake User Profiles (10 Points)**
   * Use the faker library to generate 10 fake user profiles, including name, email, job title, and company.
   * Output each profile in a readable, structured format.



1. **Generate Fake Transaction Records (10 Points)**
   * Generate 10 fake transaction records, each with a unique ID, transaction date, and amount.
   * Display each transaction in a readable format.

### ****Part 3: Docker Setup and Integration (20 Points)****

1. **Create a Dockerfile (10 Points)**
   * **Base Image**: Use a lightweight Python image.
   * **Working Directory**: Set a working directory in the container.
   * **Install Dependencies**: Include webexteamssdk and faker in a requirements.txt file, and configure the Dockerfile to install them.
   * **Environment Variables**: Use environment variables to pass sensitive information, like the Webex access token.
   * **Volumes and Data Storage**: Configure volumes to store logs or data persistently outside the container.
2. **Docker Commands for Running Tasks (10 Points)**
   * Write Docker commands for running each script independently in the container, specifying environment variables and any necessary inputs at runtime.
   * Include clear instructions for building and running the Docker container with both Webex and Faker tasks.

### ****Part 4: GitHub Documentation and Submission (30 Points)****

1. **Git Repository and Initial Commit (10 Points)**
   * Initialize a new Git repository, add the necessary project files, and commit the initial setup.
2. **README and Documentation (10 Points)**
   * Write a README.md with clear instructions on how to set up, run, and test the application using Docker.
   * Include steps to install Docker (if needed), build the Docker image, and run the container for each task.
3. **Screenshots and Project Submission (10 Points)**
   * Run the scripts and take screenshots to demonstrate the project’s functionality, including Docker builds, Webex room creation, message sending, data generation, etc.
   * Include screenshots in a folder in the repository for easy reference.
4. **GitHub Repository Link**
   * Provide a link to the GitHub repository with all scripts, Docker setup, and documentation included for review.