Develop a python based web application that enables co-workers to submit their coffee order by SMS. At the same time, every day, those orders are summarized into a single email and sent to an email address so that the person can go and pick up the coffee. Upon return from the coffee shop the person submits the total cost of the order on a web form and receives a receipt.

Additionally, there is a UI that enables an admin to review all the previous orders. I will provide the credentials for Twilio for SMS and expect the developer to create a document and diagram outlining the approach, including hosting before developing the application The application should enable users to submit their coffee orders via SMS. The application should generate a daily summary email of all orders and send it to a designated email address. The application should provide a web form for users to submit the total cost of the order and receive a receipt. The application should provide a UI for an admin to review all previous orders.

1. User sends SMS to a fixed number, which is linked to twilio account.
2. SMS content should have 3 essential items
   1. User mobile number linked to his/her profile
   2. Coffee type
   3. Coffee quantity
3. Web application receives the above information along with date and time from twilio API
4. The above information will be stored in a postgres or mysql database. It can also be stored in google sheet, if you want to save on database cost
5. At a fixed time, all the orders will be consolidated and sent to a particular user by email. Which email client are you using? If you want to use Microsoft 365 email as sender email, it needs your admin to do certain configurations. However you can also use google mail or some IMAP email to send this email to the user.
6. I would prefer postgres database, for a live application. Database will be 3 classes
   1. USER
      1. User id
      2. Email
      3. Name
      4. Admin / Standard
      5. Mobile number
   2. ORDER
      1. Coffee type
      2. Quantity
      3. Date
      4. Price
      5. Payment Status
      6. Child – Parent (User)
   3. RECEIPT
      1. Child – Parent (Order)
      2. Child – Parent(User)
7. Web App will be developed in flask framework using HTML and CSS. I will keep it simple so that anyone can maintain it in future. I will use an existing template from <https://html5up.net/>. You can also select which template would you want.
8. Webapp will have pages for user and admin
   1. User can login to see his orders and payment status
   2. Admin can login and see all the orders and payment status
   3. I think delivery status is not needed
9. Hosting has 2 options
   1. You can host in Heroku apps or pythonanywhere. You can link your domain or stay with coffeeclub.herokuapps.com
   2. You can host in godaddy or strato or any webhosting provider. Strato is cheapest in Germany, but you can see in your region. Look for one which has database included in hosting fees.
   3. Hosting with Heroku– 10 EUR/month. <https://www.heroku.com/pricing/estimates/90672efbb96078b8998da128b8ce27fcd50f1f4c61764b77a5238ad71bafa918>

<https://www.twilio.com/docs/sms/tutorials/how-to-receive-and-reply-python>

<https://learn.microsoft.com/en-us/azure/devops/pipelines/ecosystems/python?view=azure-devops>

<https://aka.ms/azpipelines-parallelism-request>

<https://www.jetbrains.com/help/pycharm/azure-sql-database.html>

<https://learn.microsoft.com/en-us/azure/app-service/quickstart-python?tabs=flask%2Cwindows%2Cazure-portal%2Cvscode-deploy%2Cdeploy-instructions-azportal%2Cterminal-bash%2Cdeploy-instructions-zip-azcli>

0QI7PAHYPAJU248J$

kmcoesdhhj