**Assignment - 1**

**Query 1: BUG**

**"I’m unable to add a new product to our pricing plan. The button is grayed out, and I’m not sure why. Could you assist?"**

**Response:** It seems like you’re experiencing a bug with our services, and I'll be very happy to assist you with it. I’ll contact the development team regarding this issue that you’re facing and get it resolved shortly. Thank you for your patience.

**Query 2: Feature Request**

**"It would be helpful if Zenskar could send automated monthly summaries of invoices to our clients. Is this feature available?"**

**Response:** Thank you for reaching out to Zenskar. Unfortunately, we don’t have that feature implemented yet. However, I’ll share your feedback with the team for future feature implementations.

**Query 3: Customer Training**

**"I’m confused about how to configure a usage-based billing schema. Could you provide some guidance or documentation?"**

**Response:** Gladly, here’s the [link](https://www.zenskar.com/docs/usage-aggregates) to the documentation. Please let us know if you need any further assistance regarding this or any other issue you might face.

**Assignment - 2**

1. **Configure account**
2. **Upload usage data**
3. **Invoice**

**3 step plan to onboard a client to Zenskar.**

**Any documents needed to prepare the customer to onboard with Zenskar.**

1. **Configure and Set up the Customer:** Our first priority would be creating the customer’s account within our services. This can be done in a few ways — Through manually creating the account on Zenskar Platform, uploading Customer Data through a Zenskar-compatible CSV file, using third-party connectors or through the Zenskar API.  
     
   Even though they are all viable, **the recommended method** would be **to push their data through Zenskar’s customer-data-ingestion APIs.**
2. **Uploading usage data:** Zenskar can pull usage data from their data sources, or data can be fed to Zenskar’s data ingestion APIs, or can be uploaded through a Zenskar-compatible CSV file. However, **the first two methods are recommended**.
3. **Generating their first invoice:** The customer’s first invoice cannot be generated without first **creating a contract.** A contract can be created via Zenskar’s contract-data-ingestion APIs, or through manual creation in Zenskar’s app, or through Zenskar’s

CRM connector, or by uploading a Zenskar-compatible CSV.

Once the contract has been made, invoices will be generated at the end of each billing cycle automatically, or can be manually generated via **Contracts > [Select the contract for which the invoice needs to be generated] > Generate invoice.**

**Extra documentation that can help them with the process: https://www.zenskar.com/docs/data-migration-and-integration-options**

Zenskar compatible CSV format for CSV uploads, quick links to documentation regarding the issue they’re facing. Or demo videos for asynchronous guidance (if any available). And a quick reference sheet with support contacts for future quick support contact.

**Assignment - 3**

**During a recent meeting, a client shared the following feedback:**

* **"We love the functionality of Zenskar, but the UI feels outdated and hard to navigate."**
* **"We had a delay in getting responses to some of our support tickets."**

**Your task:**

1. **Suggest two action items to address this feedback.**
2. **Draft a short email acknowledging the client’s concerns and outlining next steps.**

**Action Items:**

* **Outdated UI:** Get in touch with the UI/UX or Frontend team regarding this issue and involve the client’s feedback during the dev cycle to meet the customer’s requests and needs better.
* **Response Delay:** Inform management about the response delay regarding support tickets and set rigid time periods for responding to support tickets, and push for automated response after a certain time of issue not being resolved.

**Email:**

**Subject:** Following up from Zenskar regarding your Feedback

Respected Sir,

We at Zenskar would like to thank you for sharing your feedback with us. As a team, we’re aiming to improve our customers' experience with us and making sure that we meet our clients' best interests.

To address the issues that you’ve been facing, we are:

1. **Improving UI:** Our dev team has started working on changing Zenskar’s interface to make it more user friendly, and they will be asking you for your input as they go through with the development.
2. **Improving support response times:** We’re changing how our ticket system works so that response times are faster than ever before, to make sure that the issues our clients might face get resolved timely.

We sincerely appreciate your partnership and want to make sure Zenskar runs smoothly for you. If you have any other feedback please don’t be afraid to get in touch!

Best Regards,   
Arihant

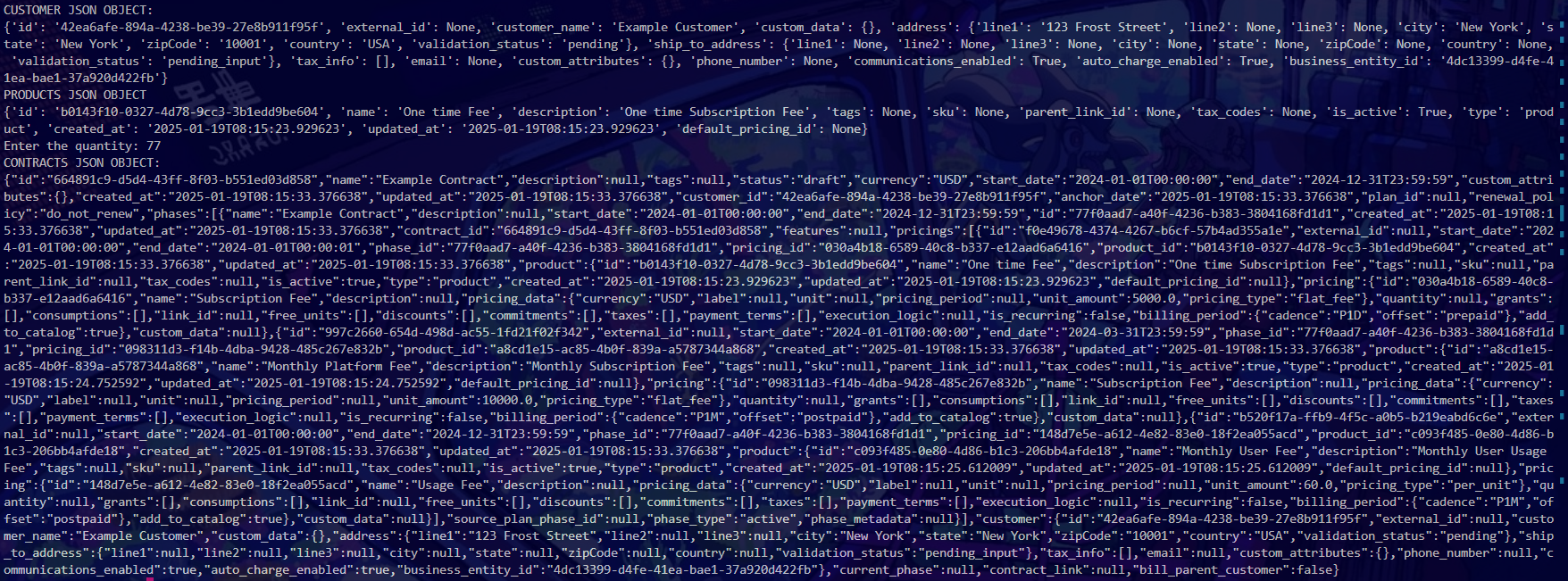
**Assignment - 4**

**Documentation:**

1. Create a virtual environment to prevent imports from clashing, and to ensure smooth development using the **venv** module. Once you’ve created a virtual environment, create a python file inside.
2. Create a .env file and store all your environment variables inside, that being your **API key** and **organisation ID.** Use the **dotenv** module to load the environment variables into your python file.
3. First step is to define the authentication headers such as the **x-api-key** and **organisation** header so as to authenticate with the Zenskar API with each request being made at the endpoints.
4. Use python’s **requests** module to send **POST** requests to the respective API endpoints, and if they all return a status of 200, and send a response text back, the request has been processed at the server.
5. Once the authentication headers are done, create a customer.
6. Enter the name, phone number and Address of the customer in order to create the customer. Send the request to the ‘**/customer**’ endpoint.
7. If the response status returns 200, the customer has been created successfully.
8. Similarly, at the ‘**/products**’ endpoint, create the three products and extract their id.
9. Use the **loads** function from **json** import from python libraries to convert the string response text to json. This will make it easier to extract the ID of the products, customer and pricing.
10. Now go to their pricing endpoint, ‘**/[product\_id]/pricing**’ to configure their pricing.
11. Once that is done, create a contract at the ‘**/contract\_v2**’ endpoint. Create a phase for how long the contract you want should be valid. And inside, declare pricing objects, with their start dates, end dates and cadence for how long and how frequently they have to be charged for the services they will be using. Do link the objects up with their respective ID’s.

**API endpoints used:**

* **/customer :** to create a customer object.
* **/products :** to create products for one time fee, monthly subscription fee, and monthly user fee
* **/products/[product\_id]/pricing :** to create the product pricing for the products created.
* **/contract\_v2 :** to create a contract consisting of the aforementioned products and pricings for the customer.

**API Responses:   
**

**Challenges Faced:**

I had a lot of problems understanding the structure of the contract, initially how to put a product inside a contract, then with the phases and pricing objects that needed to be created first. In the end I assumed the phase for all three products was supposed to be the same, while the pricings declared inside would have different start and end dates.   
  
The phone number provided (+1234567890) would not create a customer, and gave an error of “INVALID AREA CODE” so my solution to that problem was to omit it from the data where the customer is being created because it isn't necessarily required.

I also ran into this problem of not being able to assign ids to certain fields such as a product’s id to product\_id. That’s why i thought of converting string to JSON using the json module.

Please note the queries I sent on the email went unanswered, so I had to make do with assumptions. Please excuse me if the content of the code is subpar.