Making Payments from Inside Microsoft Teams—sandy inspires

A bit about me!

I work as a Data Engineer at Cognizant; I've been actively participating in hackathons and building solution that I wish to see for the past three years. Won five hackathons in last two years out of which **three are Microsoft hosted ones**. I hold six Azure certifications (including **DP-203 and DP-100**). Almost all my weekends are dedicated to hackathons, open source, and building on Azure. Recently delivered a Speech at **Azure Community Conference 2022** held at The Leela Palace, Chennai, India (Asia's Largest Azure Community Conference). I'm also the Azure Developer Community Lead for Chennai region.

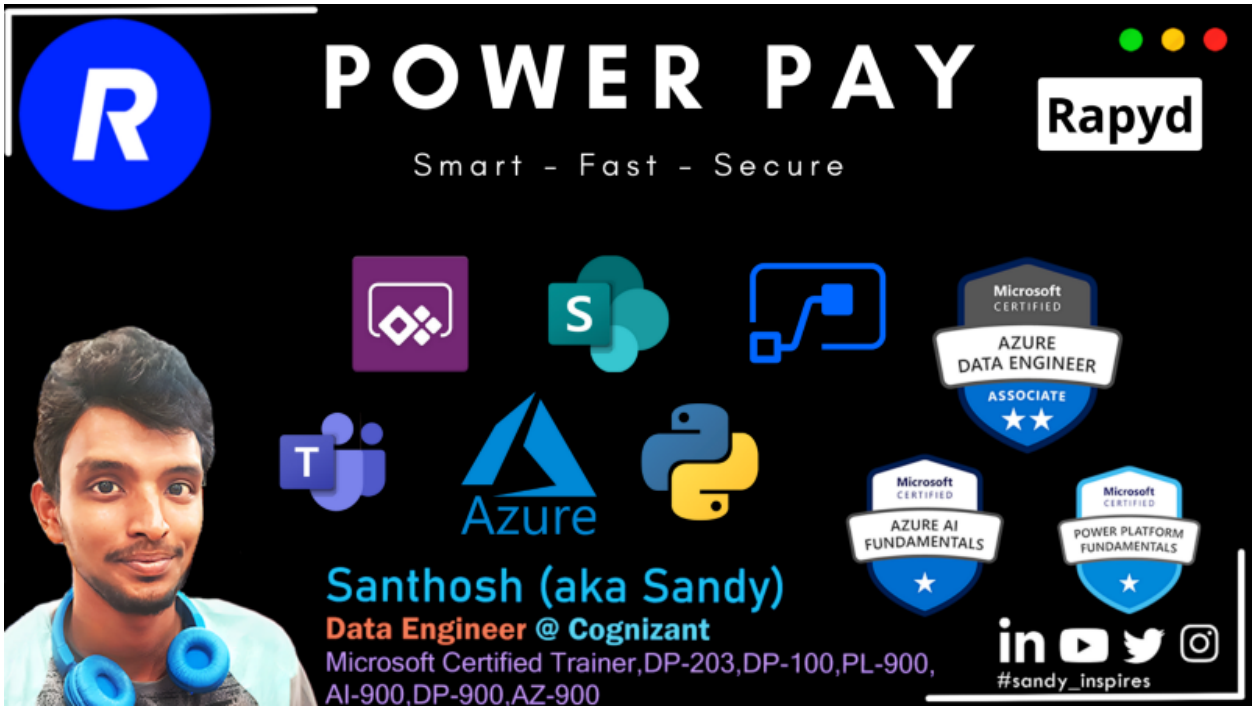
Problem we are solving!

There is this notion of fear when you enter your credit card details into a website. This application eliminates the fear by providing payments from within the Microsoft Teams application. Teams has now roughly **270M+ active monthly users**, so if you have a shop and it's available on the Microsoft AppSource and you need a payment solution, mostly it would be a redirect to an external page. Why is that a case, can't we just put it inside Teams so people can be sure that the application is legit, complient with government standards, and secure. Boom! Power Pay is the solution for it. Microsoft's review process for a new app in AppSource is vigorous and follows the best standards to eliminate all Spam content and stores.

Learn more about it here—**Make your solutions available in Microsoft AppSource and within Office | Microsoft Learn**

For those people who say, "Just show me what you have built", here you go -> 🕵️‍♂️💍✨

Watch the Live Demo on YouTube 👇



Working demo video of Power Pay

https://youtu.be/-iR4m0lcx9Y

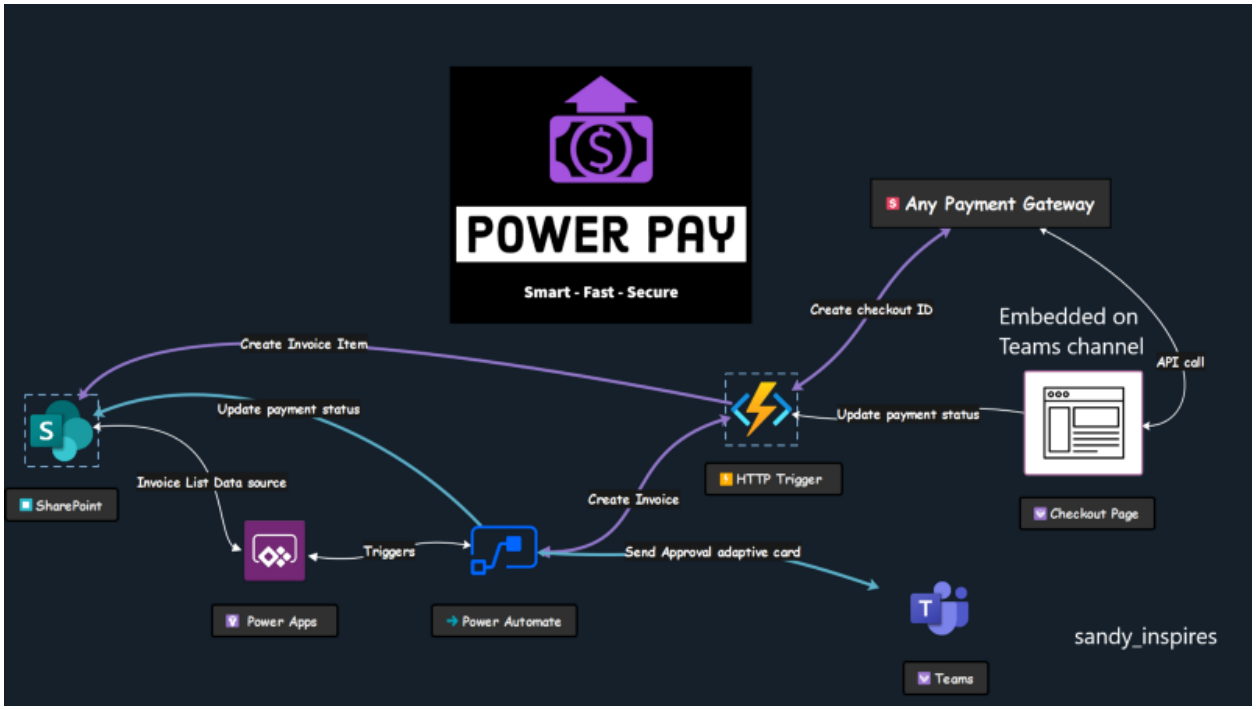
For those people who say, "Talk is cheap just show me the code", here you go -> 👨‍💻✌️

Santhoshkumard11/Power-Pay-with-Rapyd (github.com)

Azure Services Used

As developers we love automation tools, the complete solution is built on the following to deliver the value and speed to the clients

- **Power Apps**—a No/Low code platform
- **Power Automate**—Workflow automate
- **Adaptive Card**—seamless integration with Power Automate
- **Azure Function**—Runs **Python** code to manage all the backend services for robust usage
- **SharePoint List**— using it as a database for storing all the data



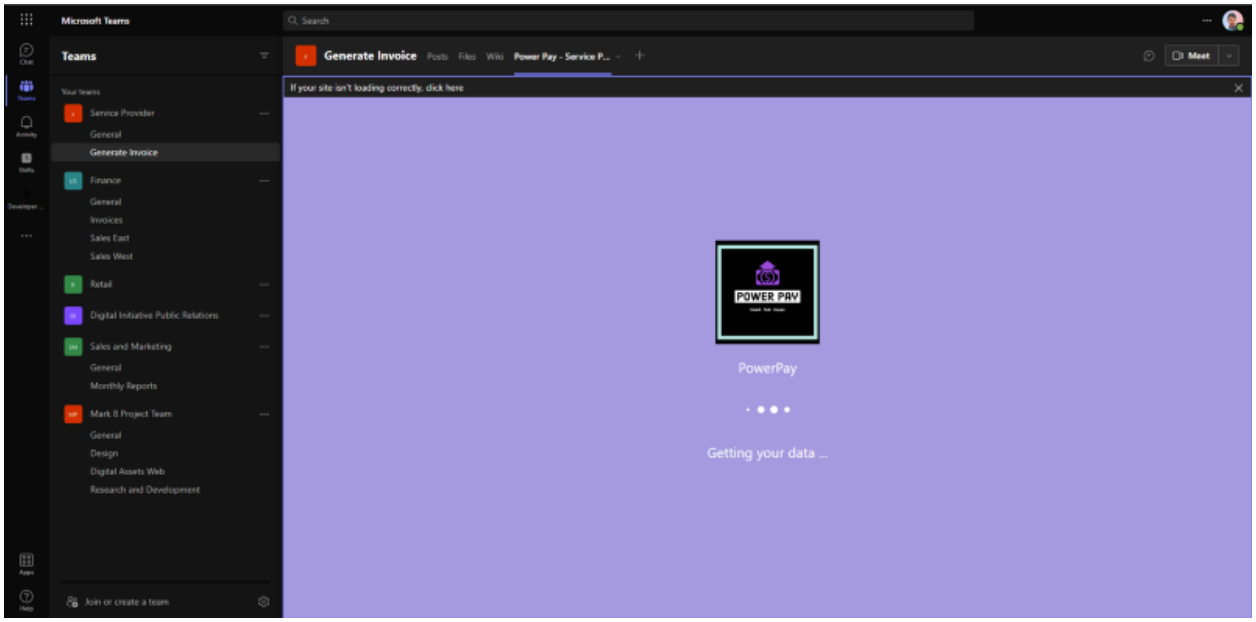
Power Pay — Architecture Diagram

I have used **Python** extensively in this one, feel free to replicate the same in your favorite language. 🐱👤🐱👤

What it can do!

The complete invoice payment experience is done from within your most used application (Microsoft Teams) for B2B payments of invoices

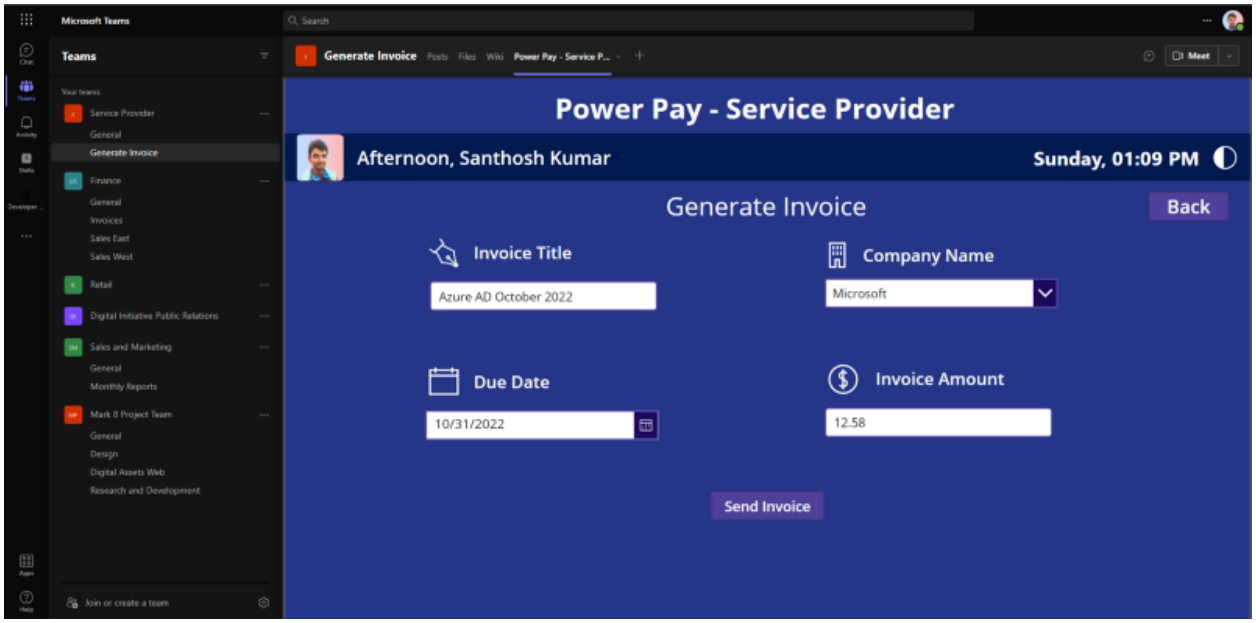
- Pay shop invoices from within Teams
- Track invoice status within Teams
- Generate invoices from Teams and send them to the client
- Invoice approval flow
- Theme Toggle—Light/Dark theme matching the Microsoft Teams themes



Initial Loading Screen

Flow Started 💨💨

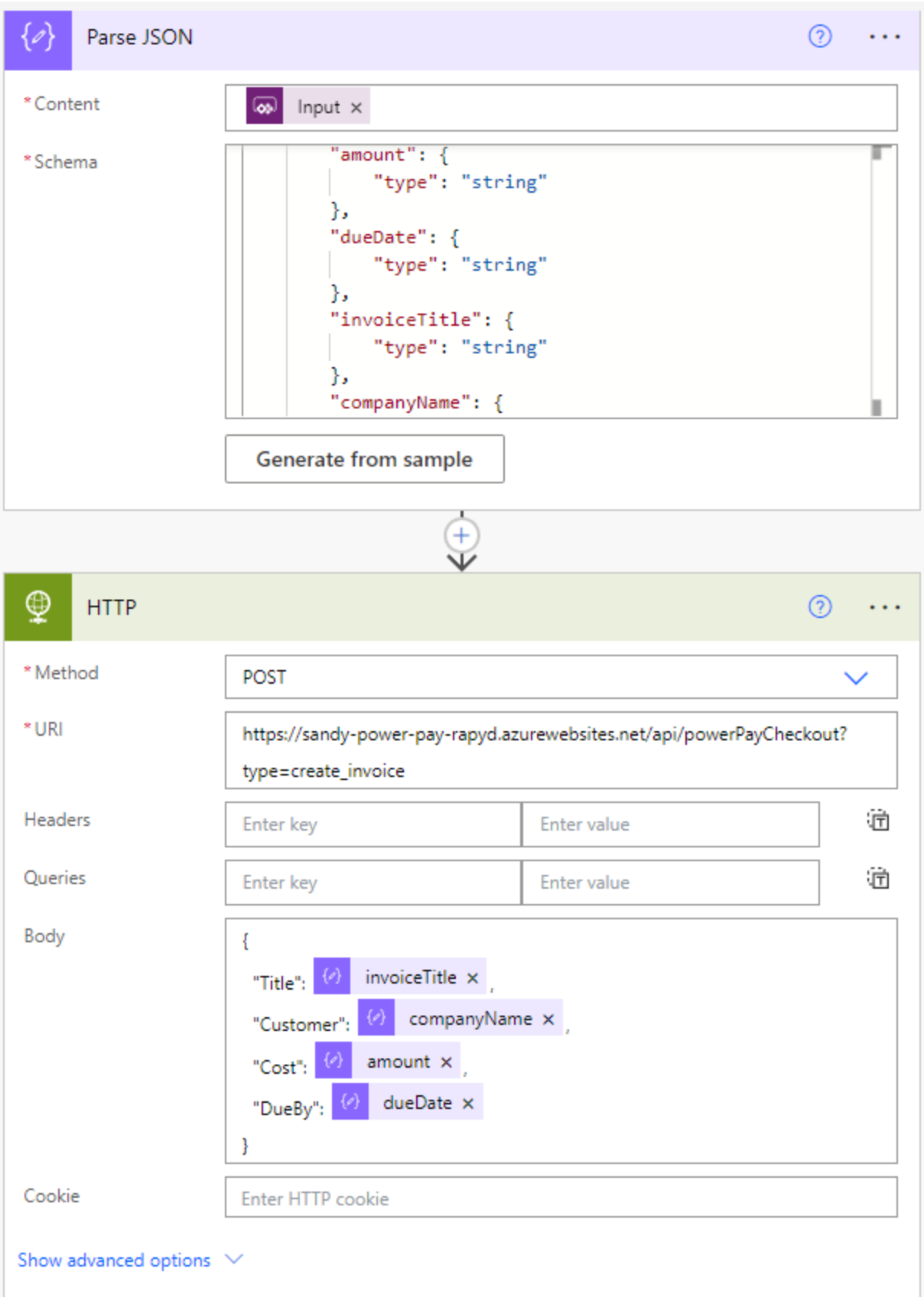
0️⃣First step is to generate an invoice and send it across to the client, mentioning the invoice title, the name of the client (company), due date, and amount payable.

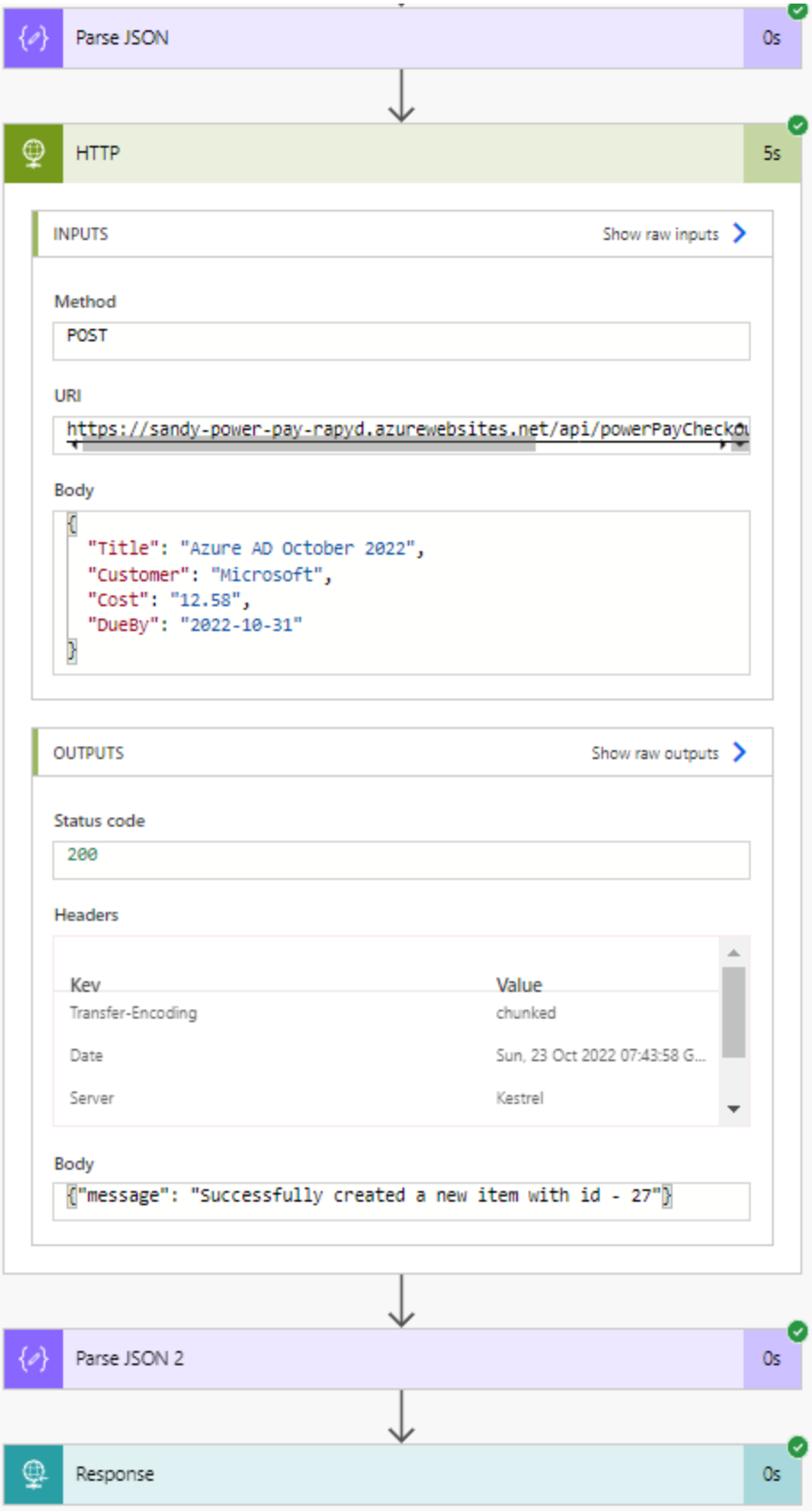


Power Apps — Generate Invoice Screen

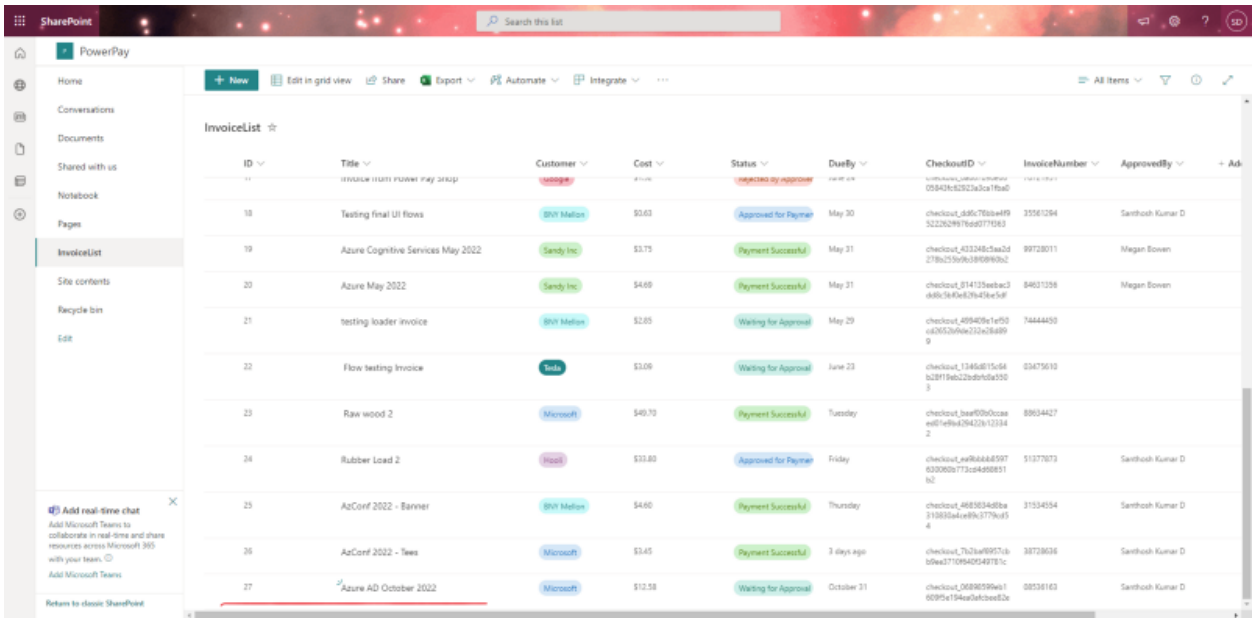
Once you hit send, this goes directly to the selected client and an approval processed is triggered.

Behind the scenes the data is sent to an Azure Function endpoint that talks to the Rapyd Payment servers to create a new checkout id for the requested amount and stores the details in SharePoint list using Microsoft Authentication Library (**msal Python package**).

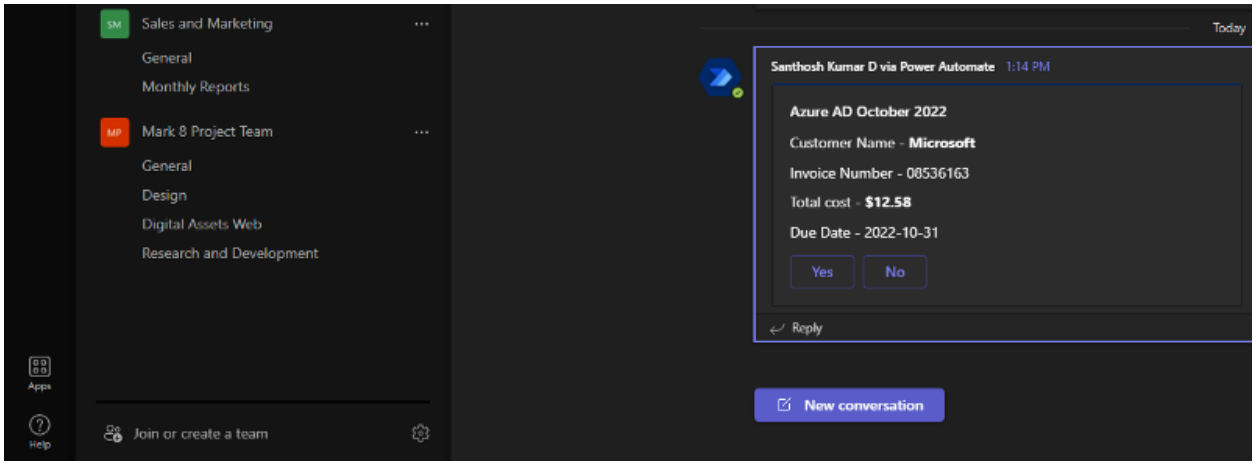

Parse JSON
*Content
Input ×
*Schema
"amount": {
"type": "string"
},
"dueDate": {
"type": "string"
},
"invoiceTitle": {
"type": "string"
},
"companyName": {
Generate from sample
HTTP
*Method
POST
*URI
https://sandy-power-pay-rapyd.azurewebsites.net/api/powerPayCheckout?type=create_invoice
Headers
Enter key
Enter value
Queries
Enter key
Enter value
Body
{
"Title": invoiceTitle ×
"Customer": companyName ×
"Cost": amount ×
"DueBy": dueDate ×
}
Cookie
Enter HTTP cookie
Show advanced options


Parse JSON
0s
HTTP
5s
INPUTS
Show raw inputs
Method
POST
URI
https://sandy-power-pay-rapyd.azurewebsites.net/api/powerPayCheckou
Body
{
"Title": "Azure AD October 2022",
"Customer": "Microsoft",
"Cost": "12.58",
"DueBy": "2022-10-31"
}
OUTPUTS
Show raw outputs
Status code
200
Headers
Key
Value
Transfer-Encoding
chunked
Date
Sun, 23 Oct 2022 07:43:58 G...
Server
Kestrel
Body
{"message": "Successfully created a new item with id - 27"}
Parse JSON 2
0s
Response
0s

Once the Azure Function is triggered and completes successfully, a new item is added to the SharePoint List (in this case id 27 is created)

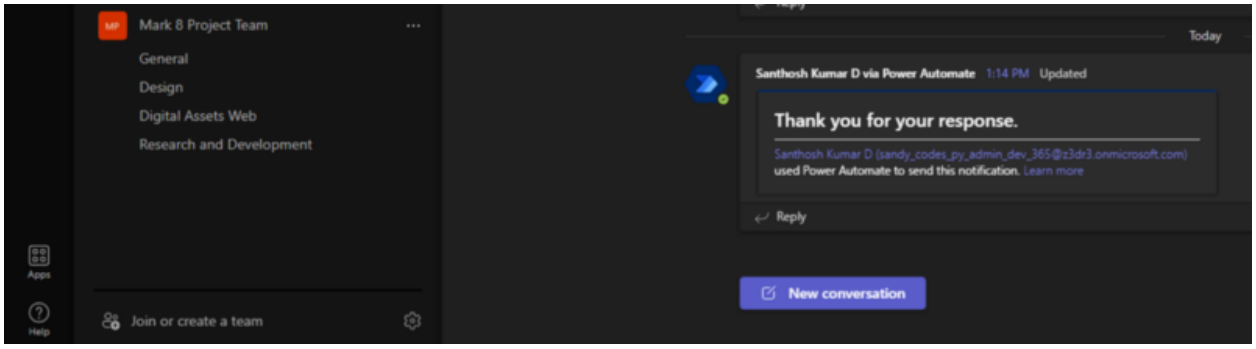


1. There is a Power Automate flow which sends out an adaptive card to the Invoice channel for approval



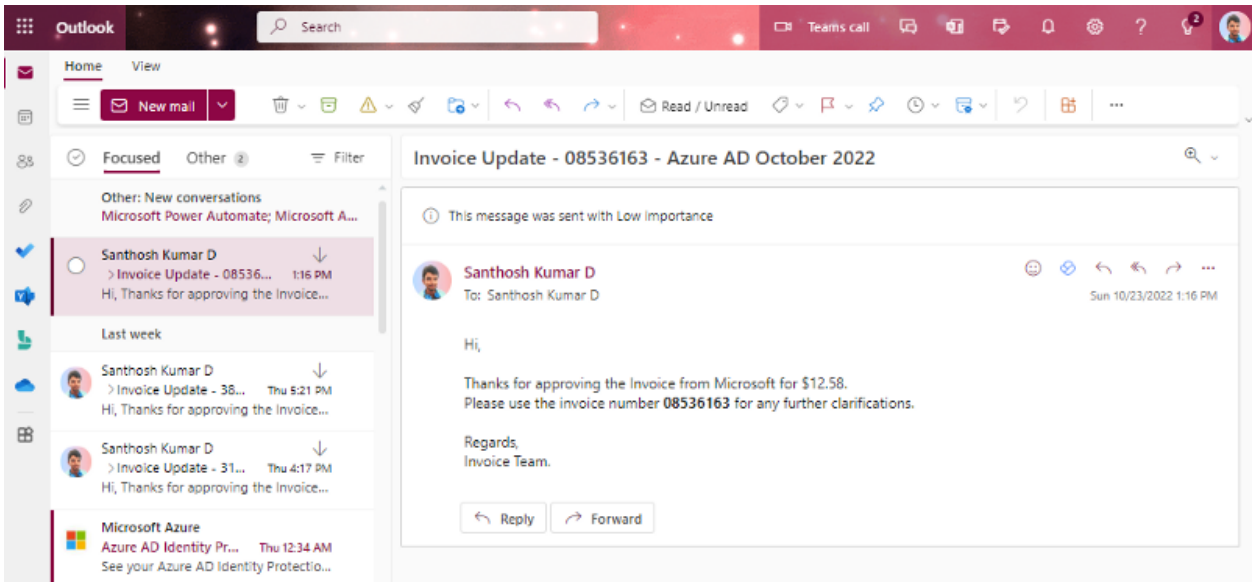
Adaptive Card in Teams Channel — Requested

Once you click on Yes, the below Adaptive card is sent as a response.



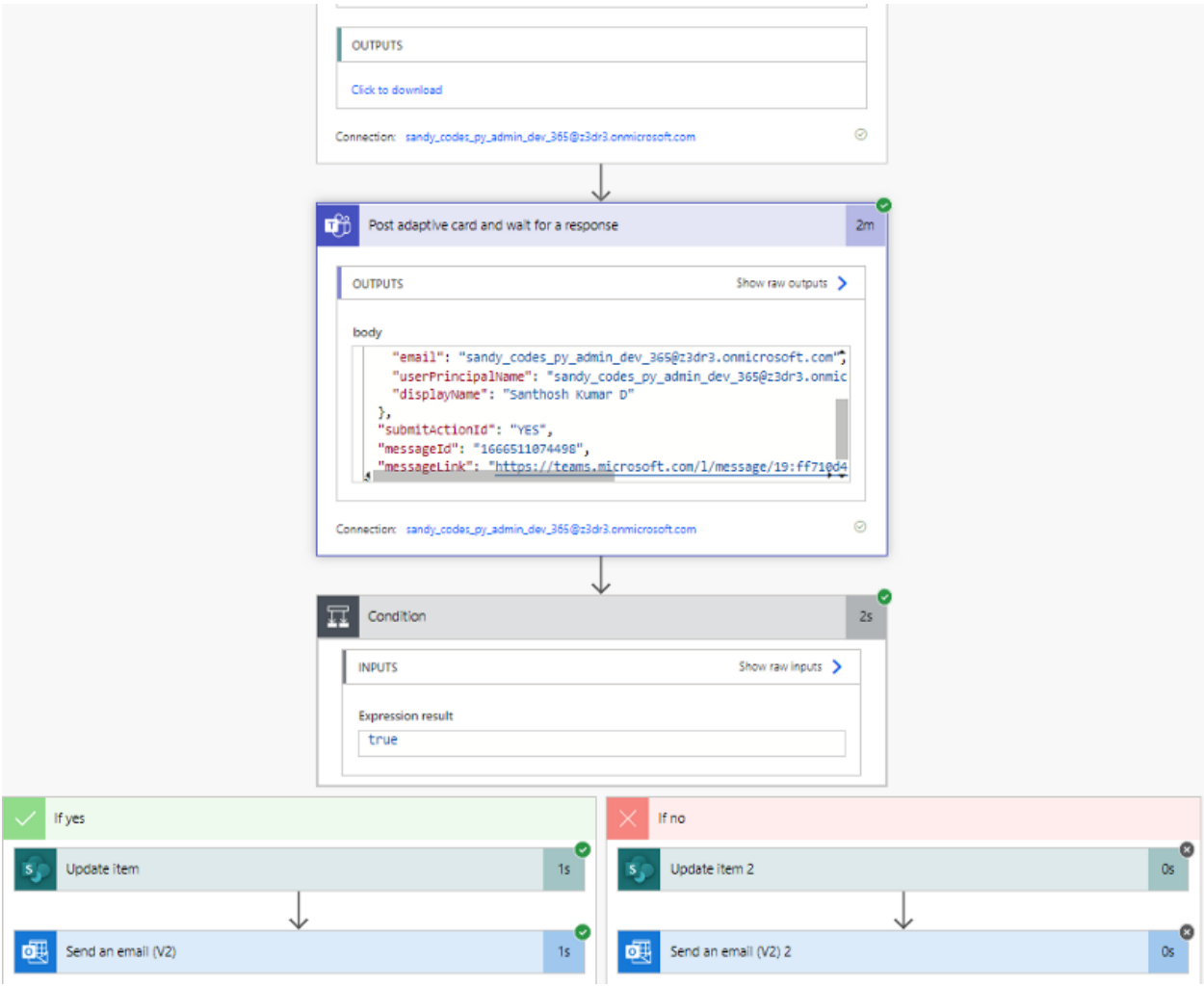
Adaptive Card in Teams Channel — Approved

An email is also triggered once the approval is done



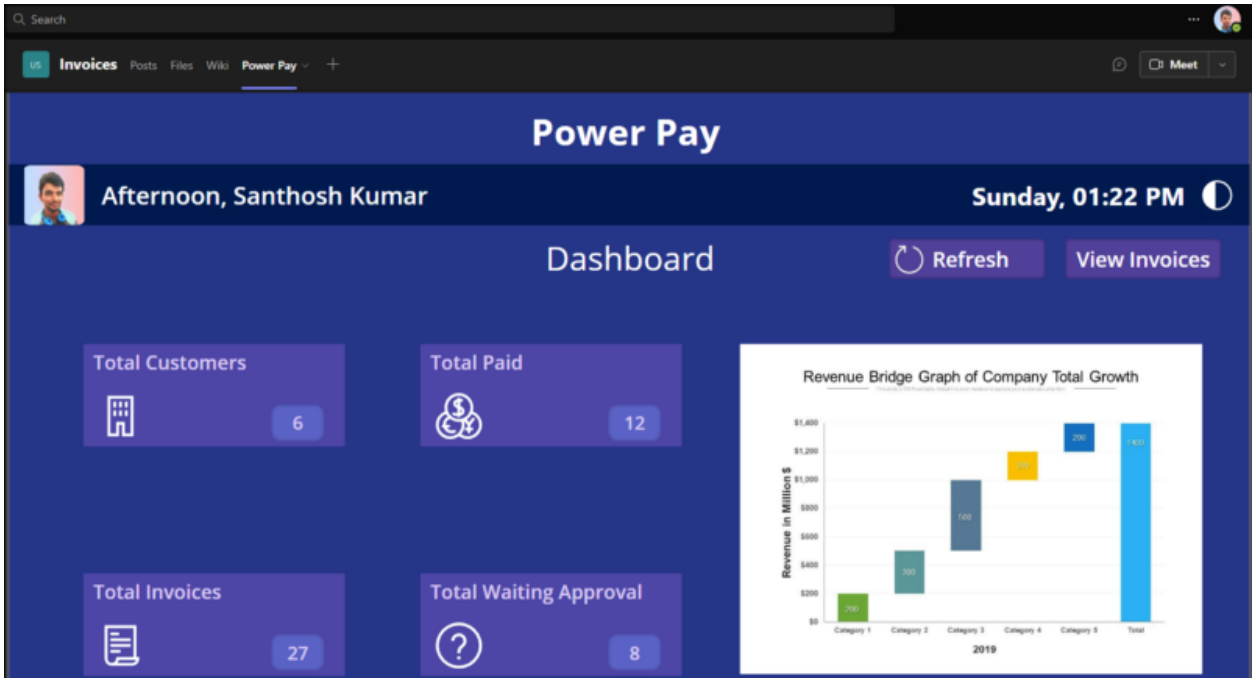
Outlook — Invoice approved acknowledgement

You can see the response received from the Adaptive card with who approved it and at what time, you can also add a logic on who can approve it and have a time cap for approval



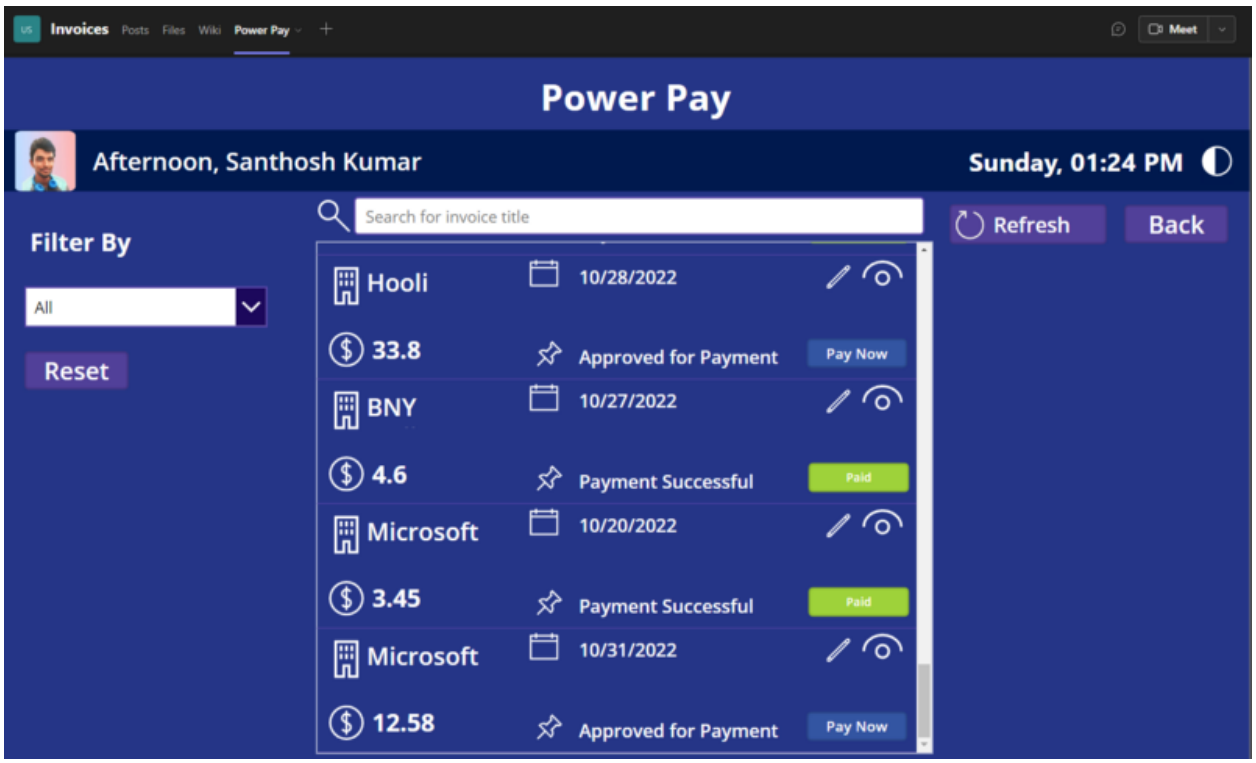
Power Automate — Approval Flow

Now let's look at the Power Apps Dashboard,



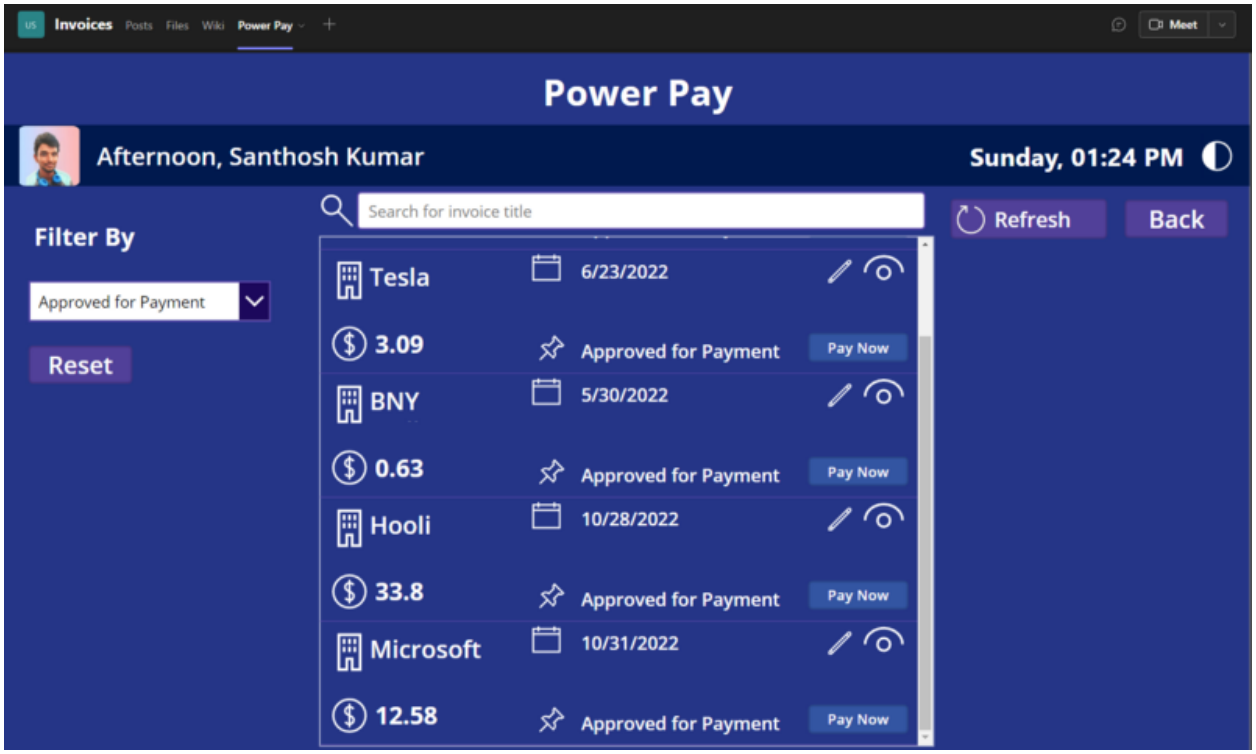
Power Apps — Dashboard

2 You can see all the payments that are made to various vendors



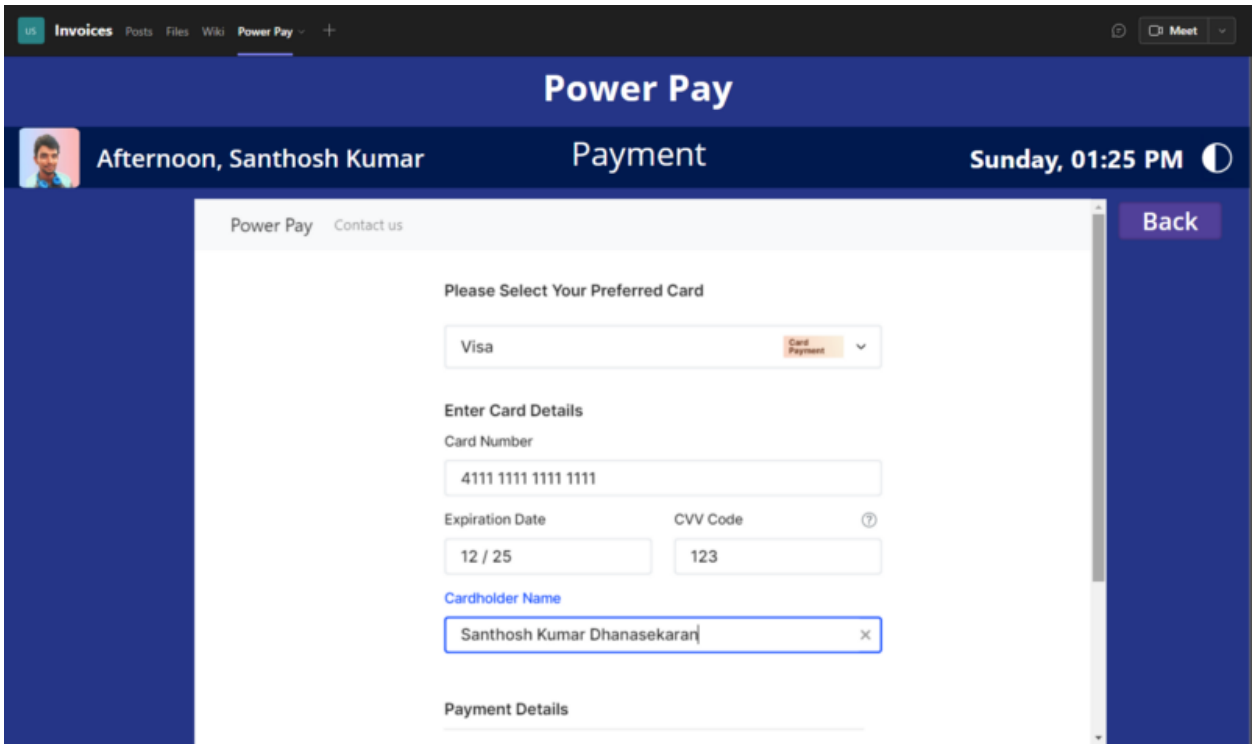
Power Apps — List view

This is a filtered view of only items which are *Approved for Payment*



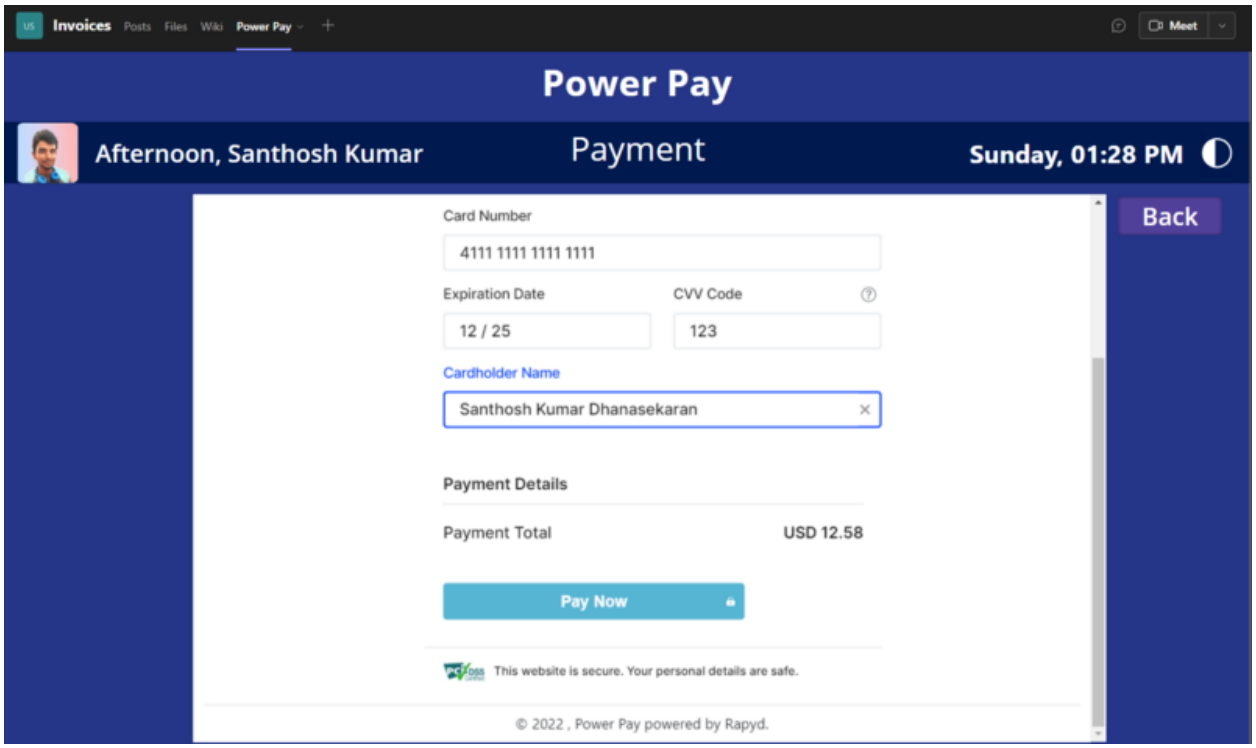
Power Apps — List view — filter

3 Now we are ready to do the payment, once you click on pay, we'll see an embedded checkout page right inside Teams.



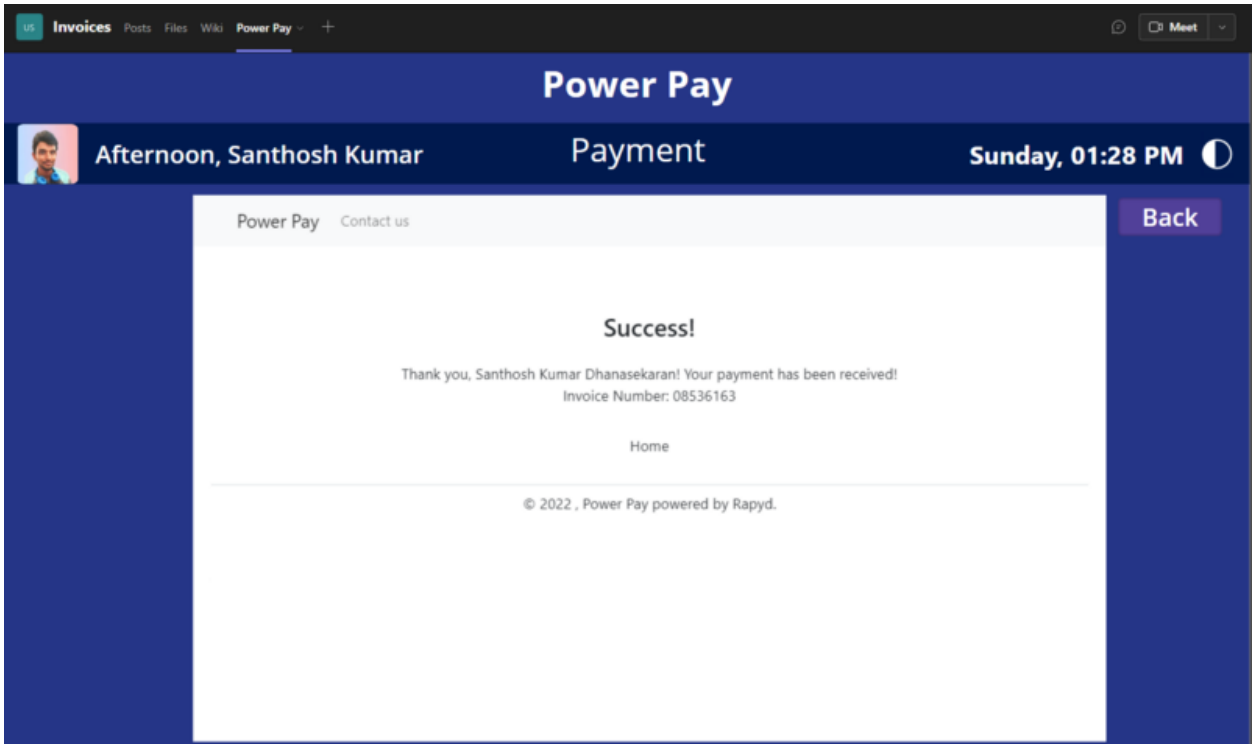
Power Apps — Checkout Page

This is a custom checkout page using Azure Functions



Power Apps — Checkout Page

4 Once the payment is done, we have a **callback URL** to our Azure Functions registered with Payment vendor which send a post request to our Azure Function endpoint, and it will update the SharePoint list item with the payment status of this checkout page



Power Apps — Checkout page — success

We'll now have the status updated for the item 27 in this case in our SharePoint

InvoiceList ☆

ID	Title	Customer	Cost	Status	DueBy	CheckoutID	InvoiceNumber
21	testing loader invoice	BNY Mellon	$2.85	Waiting for Approval	May 29	checkout_499409e1ef50cd2652b9de232e28d899	74444450
22	Flow testing Invoice	Tesla	$3.09	Waiting for Approval	June 23	checkout_1346d815c64b28f19eb22bdbfc8a5503	03475610
23	Raw wood 2	Microsoft	$49.70	Payment Successful	Tuesday	checkout_baaf00b0ccaaed01e9bd29422b123342	88634427
24	Rubber Load 2	Hooli	$33.80	Approved for Paymer	Friday	checkout_ea9bbbb8597630060b773cd4d68651b2	51377873
25	AzConf 2022 - Banner	BNY Mellon	$4.60	Payment Successful	Thursday	checkout_4685834d8ba310830a4ce89c3779cd54	31534554
26	AzConf 2022 - Tees	Microsoft	$3.45	Payment Successful	3 days ago	checkout_7b2baf6957cbb9ee3710f640f349781c	38728636
27	Azure AD October 2022	Microsoft	$12.58	Payment Successful	October 31	checkout_06898599eb1609f5e194ea0afcbee82e	08536163

SharePoint List

IMPORTANT NOTE

Starting June 30th, 2020, we will no longer add any new features to Azure Active Directory Authentication Library (ADAL) and Azure AD Graph. We will continue to provide technical support and security updates, but we will no longer provide feature updates. Applications will need to be upgraded to Microsoft Authentication Library (MSAL) and Microsoft Graph.

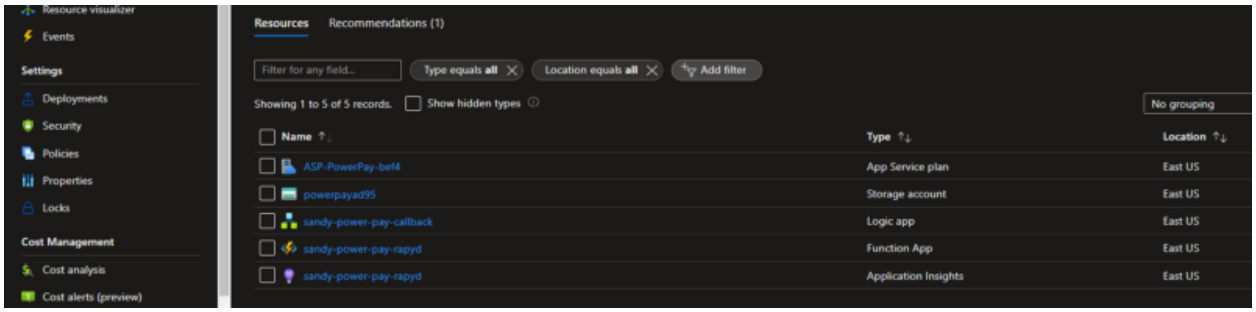
I was trying to use ADAL Python package and can't make it work, use msal Python package and follow the below steps to create a MSFT_CLIENT_SECRET and MSFT_CLIENT_ID.

```
pip install msal
```

1. Go to your Azure Portal
2. Search for Azure Active Directory
3. Click on App Registration from the left-hand sidebar
4. Give it a name and select single tenant for now
5. This will provide you with the client secret, paste it in your .env file
6. use the os.getenv() method to get the values

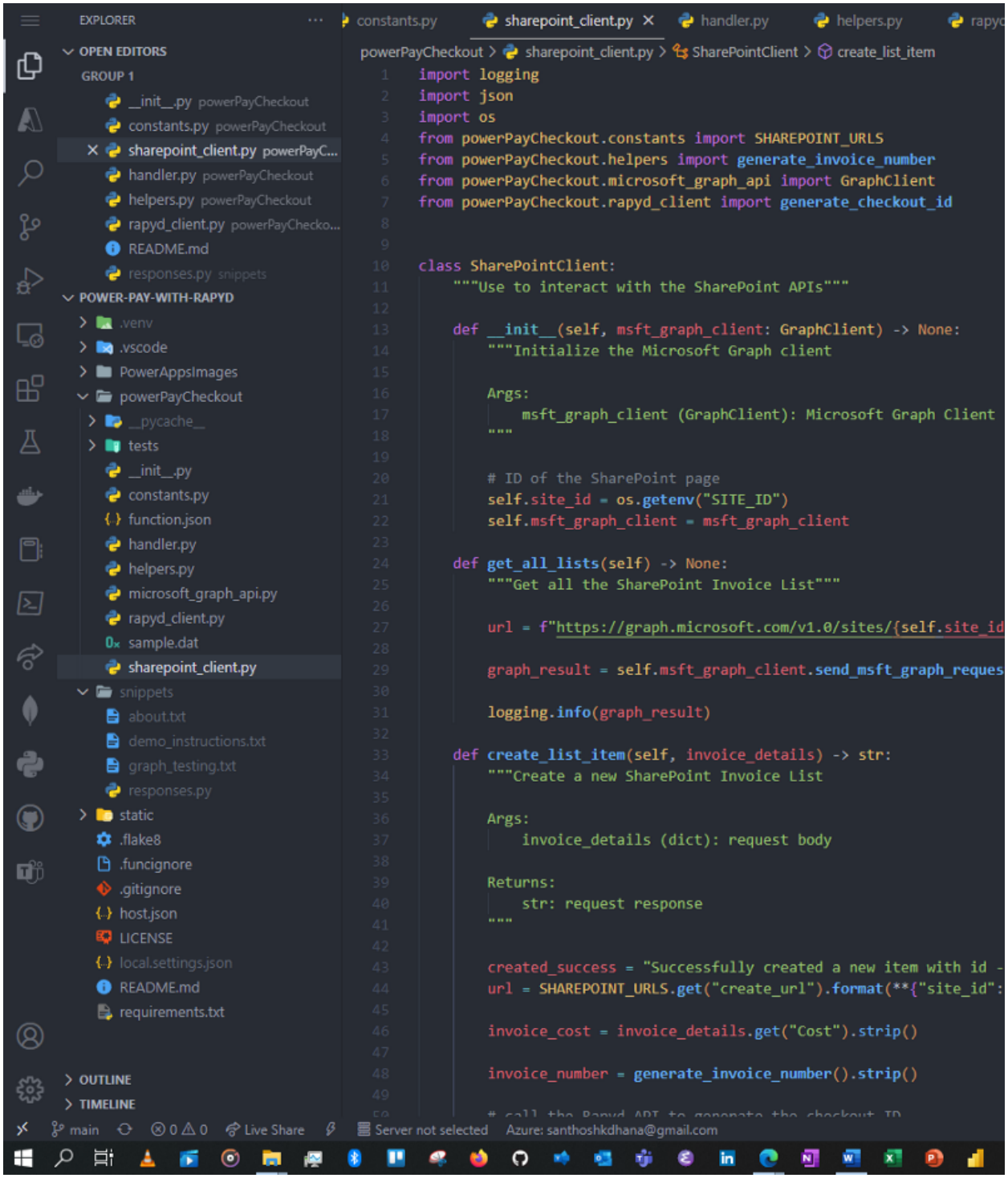
Note: Always create a new venv and install all the dependencies

Peek at the resource group which has all the services,



Azure Portal — Resource Group

Create a custom SharePoint connector to make sure it's working smooth with my style,



VS Code

Use Case/ Benefits

- For small businesses which need supervision on each payment
- For an existing Teams App which need an easy payment solution
- Easy to integrate plugin type component for Power Apps

Future Work/Scope

- Getting approval from Microsoft AppSource to be published as an App that you can add to your Microsoft Teams
- Need to get an MPN (Microsoft Partner Network) ID
- Adding a proper Admin access to the Power Apps
- Adding a proper Admin access to the Power Apps
- Need to implement multi-tenant model for enabling subscription pricing model

Feel free to reach out if you need any further clarification on the implementation. I would be more than happy to answer those. 🤝

Looking forward to hear from you and improvements via Comment or PR (Pull Request in GitHub) are most welcomed. 🙏🙏

Congratulations!! 🎉🎉you have successfully completed reading this huge blog. 🤩🤩

Thanks a lot for reading out till the end. 👓

Let's connect if you want to collaborate on the further work or a quick catch up. 🤝🤝

https://www.linkedin.com/in/santhosh-kumard