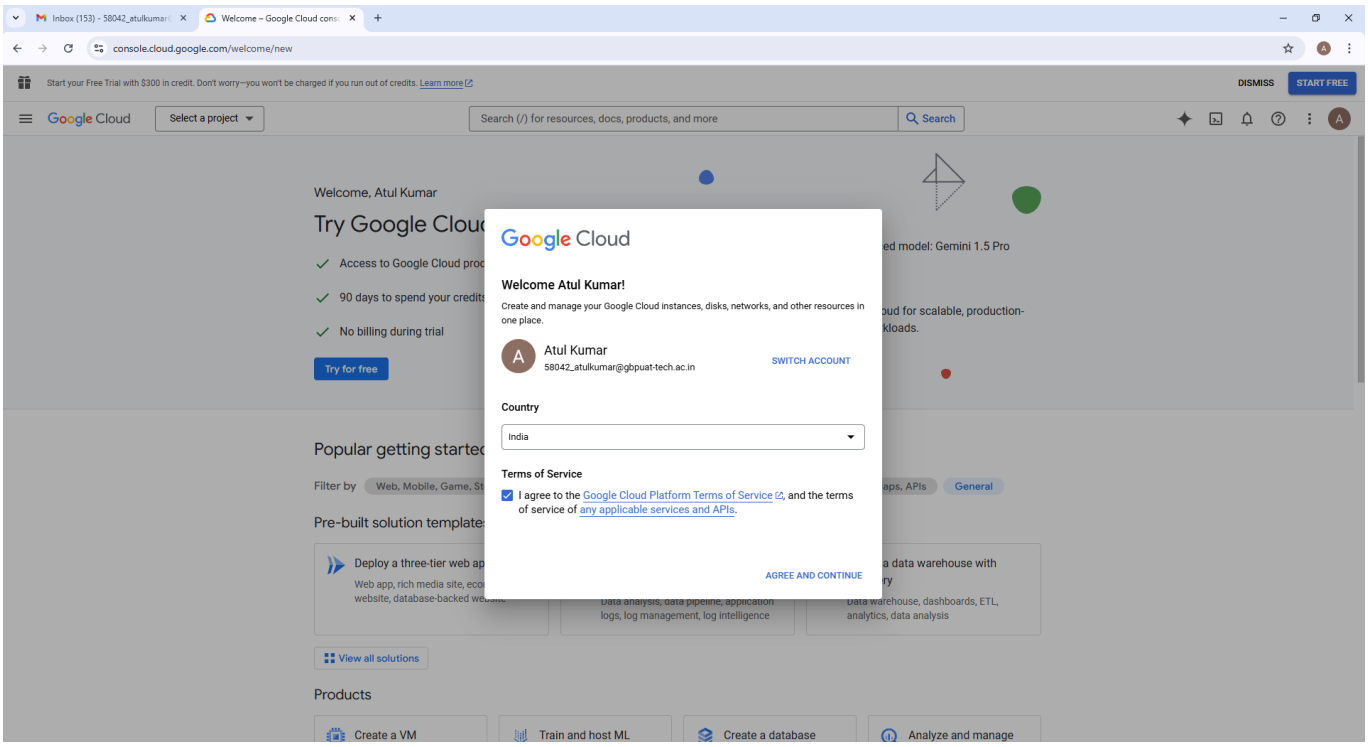
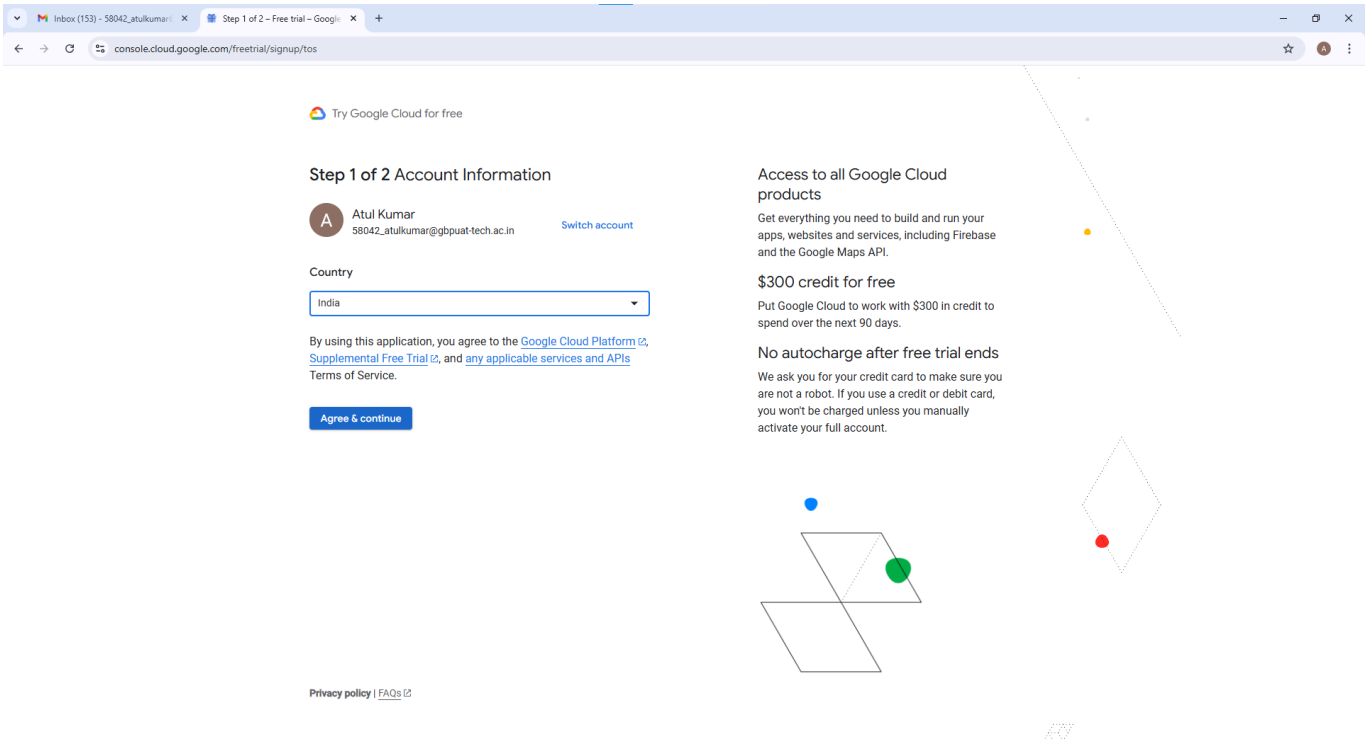
Sending Mail via GMail API Guide

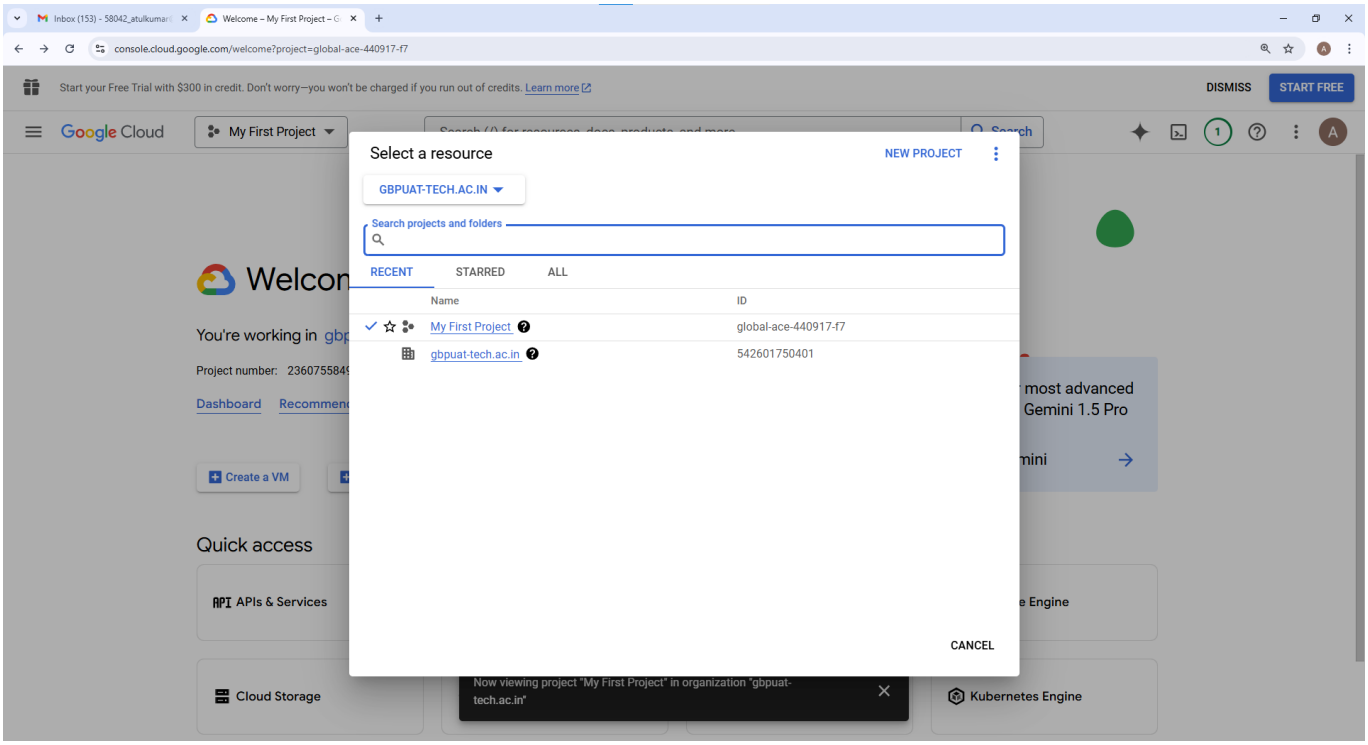
Goto *console.cloud.google.com* and create an account

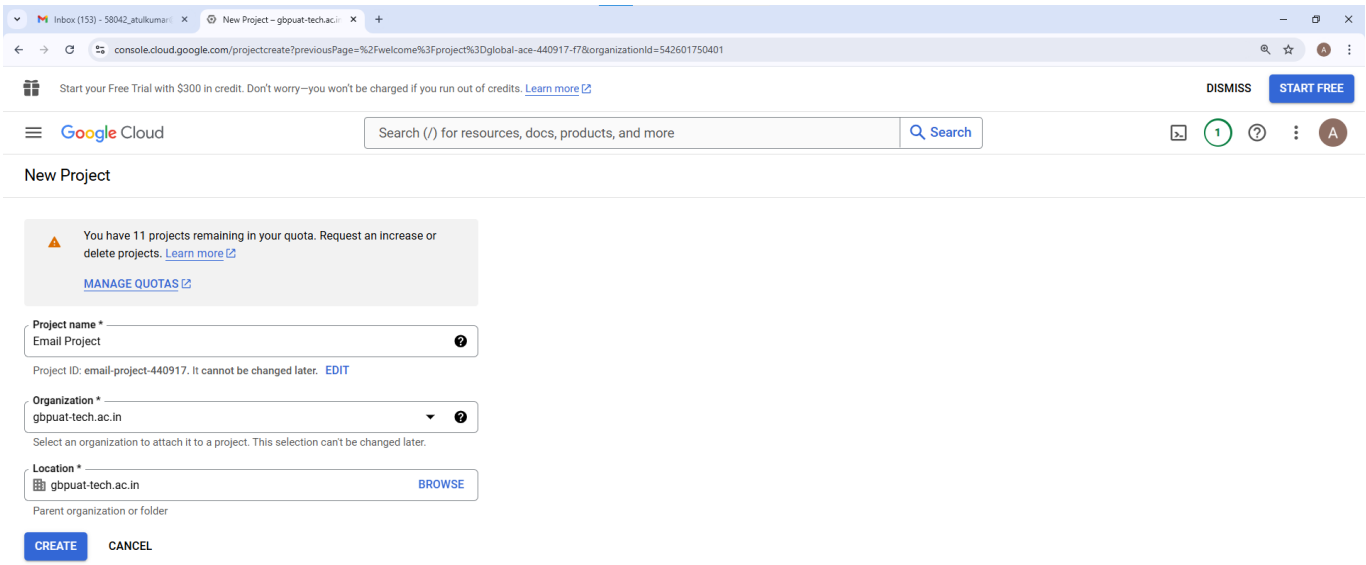


Complete the setup of account

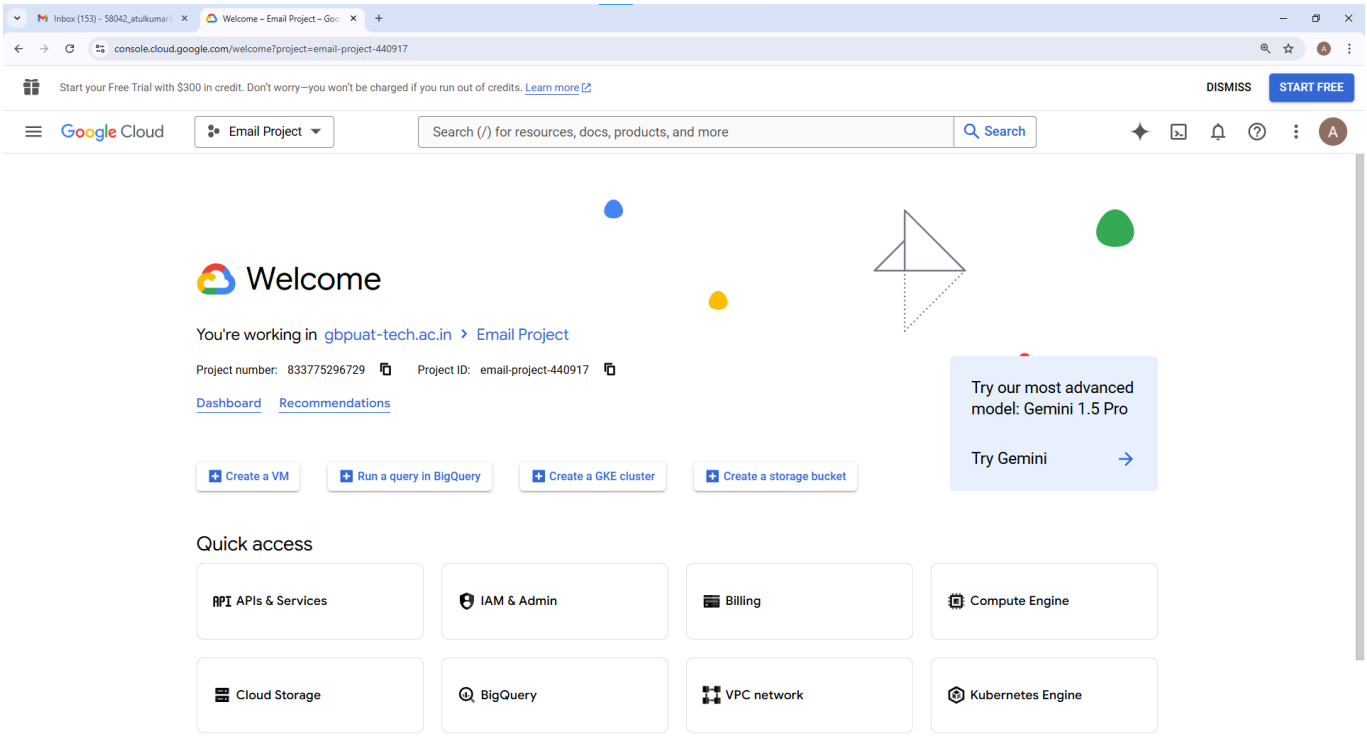


Create a **New Project** by navigating the top pane

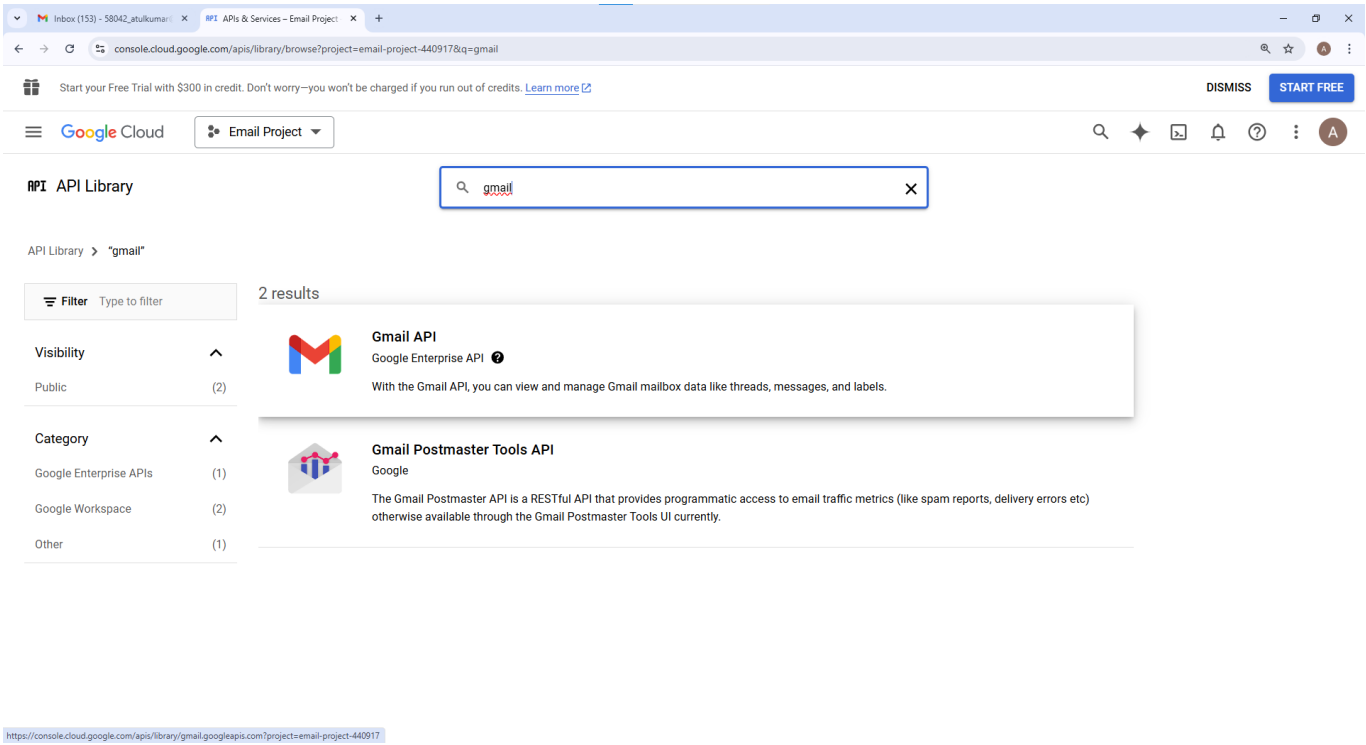




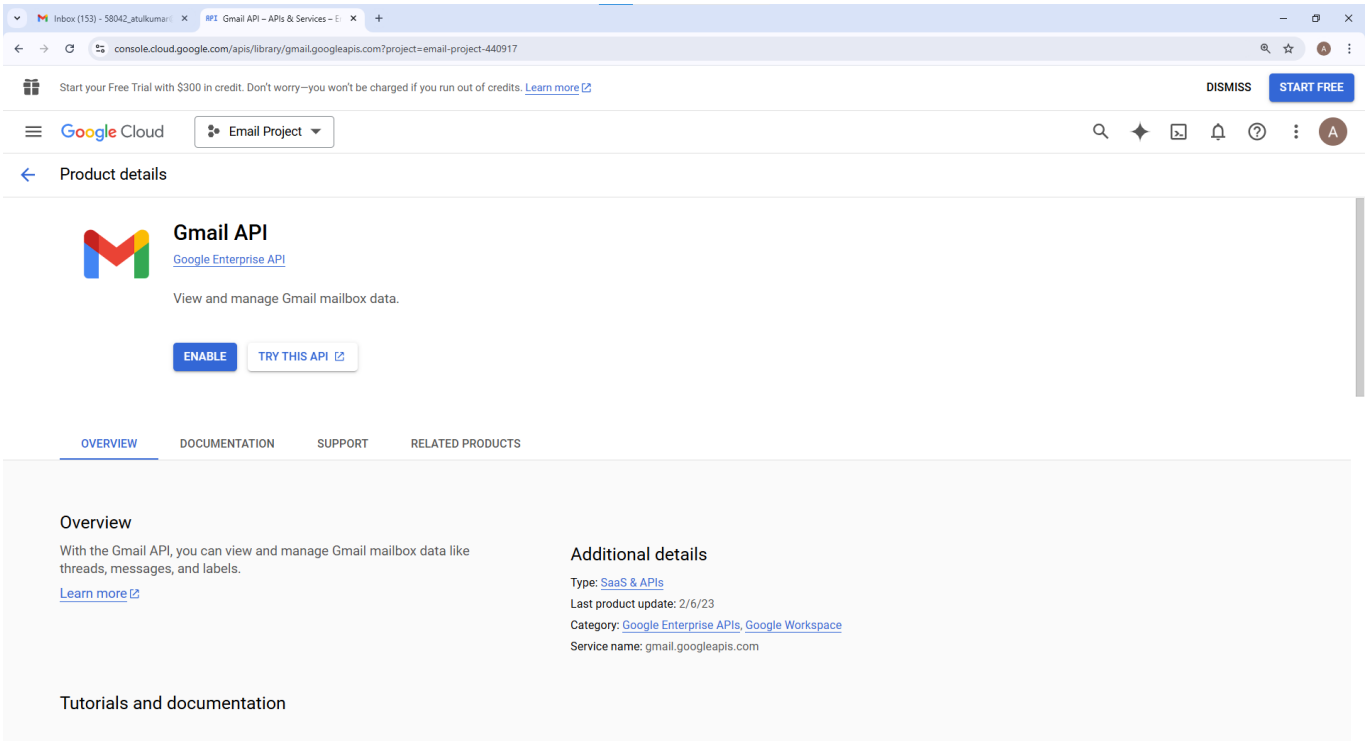
Change the project to new project from top pane



Goto **APIs & Services** from the left pane then select **Enabled API & Services**.

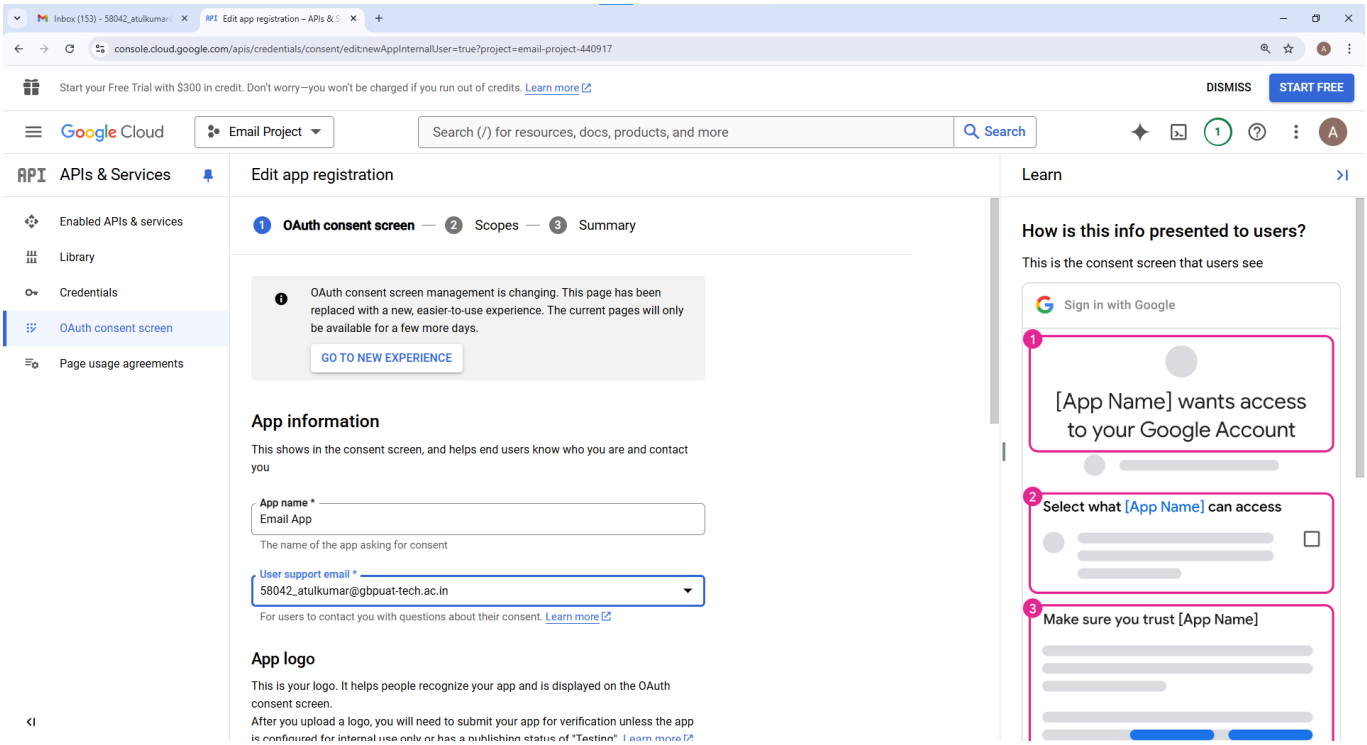


Search for Gmail * Then select **Enable**.

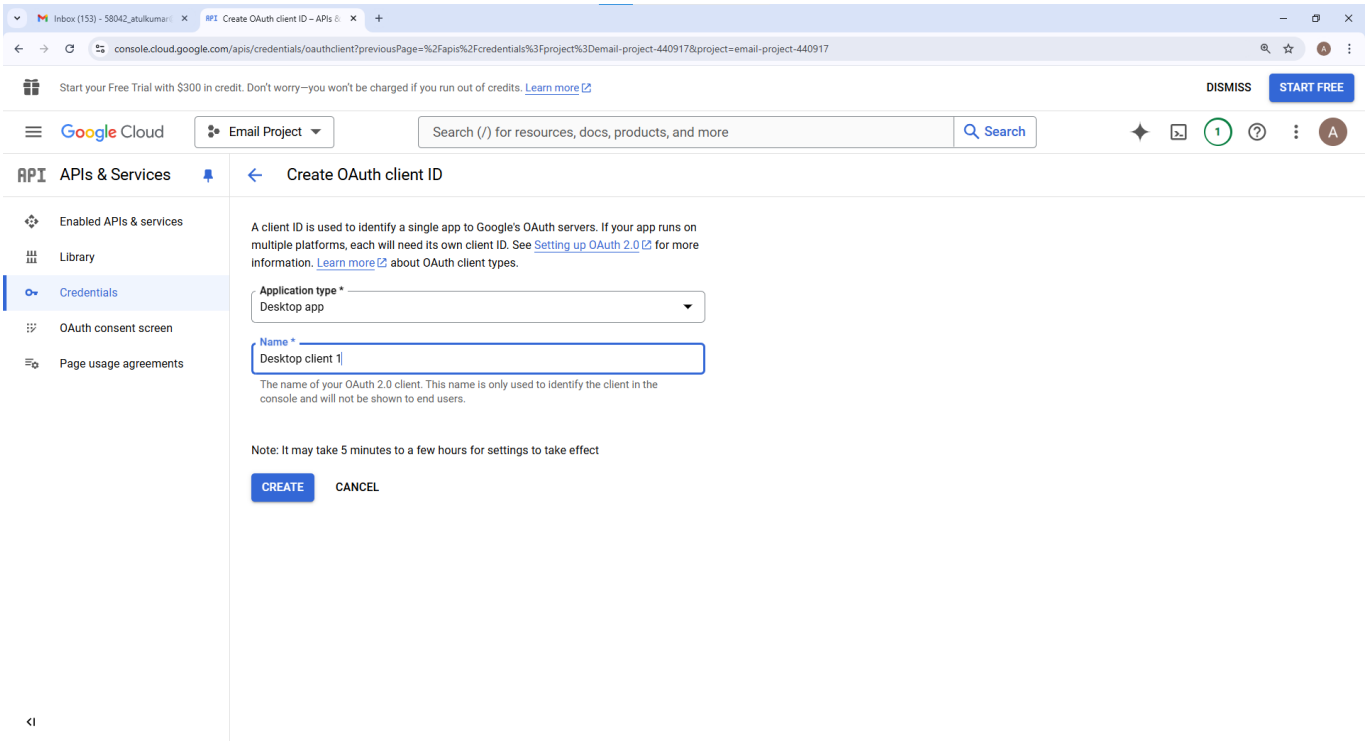


Click Credentials → then click **Create Credentials**

- Create consent screeen



- Create OAuth credendials



- Select Application type → Destop app
- Give the app a relevant name or choose the default name. Click on Create
- Download the client secret in .json format and rename it to client_secrets.json . **The client_secret.json file contains critical account information and should not be shared with anyone.**

Create a project directory and change directory

```
mkdir gmail_api
cd gmail_api
```

Create a python virtual environment and activate it

```
python -m venv venv/          # Create environment
source venv/bin/activate      # Activate environment Linux
.\venv\Scripts\activate       # Activate environment Windows
```

Install the gmail library inside the virtual environment

```
pip install --upgrade google-api-python-client google-auth-httplib2 google-auth-
oauthlib
```

Move the client secret (client_secrets.json) downloaded in step 8 to gmail_api/ directory and **do not share this file anywhere else** since your gmail account can be accessed by anyone using these credentials.

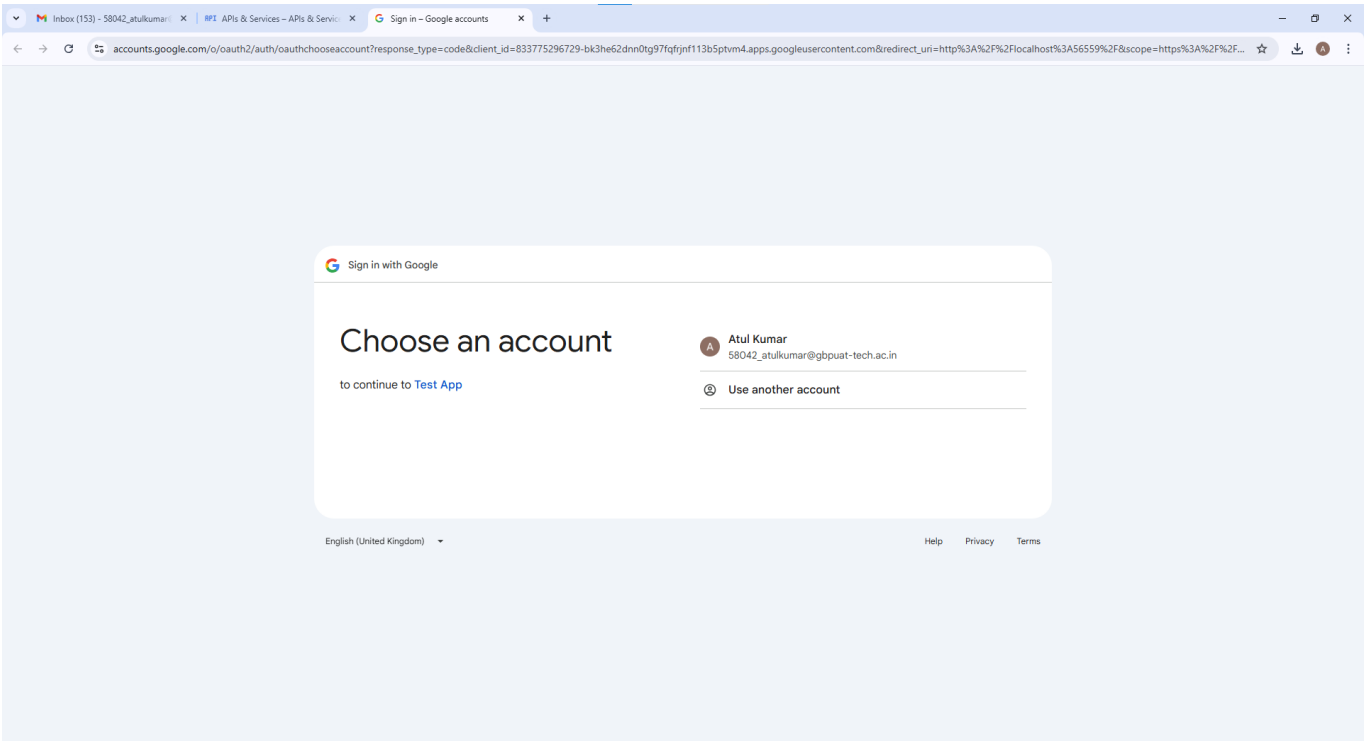
Create a new python file inside **gmail_api/** titled **send_mail.py**:

```
'''
This module sends emails with attachments to the participants Reference -
https://developers.google.com/gmail/api/quickstart/python
'''

import os
from google.auth.transport.requests import Request
from google.oauth2.credentials import Credentials
from google_auth_oauthlib.flow import InstalledAppFlow
from googleapiclient.discovery import build
from googleapiclient.errors import HttpError
from email.mime.text import MIMEText
import base64


# If modifying these scopes, delete the file token.json.
SCOPES = ['https://www.googleapis.com/auth/gmail.send']

def aunthentication():
    creds = None
# The file token.json stores the user's access and refresh tokens, and is
# created automatically when the authorization flow completes for the first
# time.
    if os.path.exists('token.json'):
        creds = Credentials.from_authorized_user_file('token.json', SCOPES)
    # If there are no (valid) credentials available, let the user log in.
    if not creds or not creds.valid:
        if creds and creds.expired and creds.refresh_token:
            creds.refresh(Request())
        else:
            flow = InstalledAppFlow.from_client_secrets_file(
'client_secrets.json', SCOPES)
            creds = flow.run_local_server(port=0)
            # Save the credentials for the next run
            with open('token.json', 'w') as token:
                token.write(creds.to_json())
    return creds




def prepare_and_send_email(recipient, subject, message_text):
    """Prepares and send email with attachment to the participants """
    creds = aunthentication()

    try:
        # Call the Gmail API
        service = build('gmail', 'v1', credentials=creds)
        sender='58042_atulkumar@gbpuat-tech.ac.in'
        #create message
```

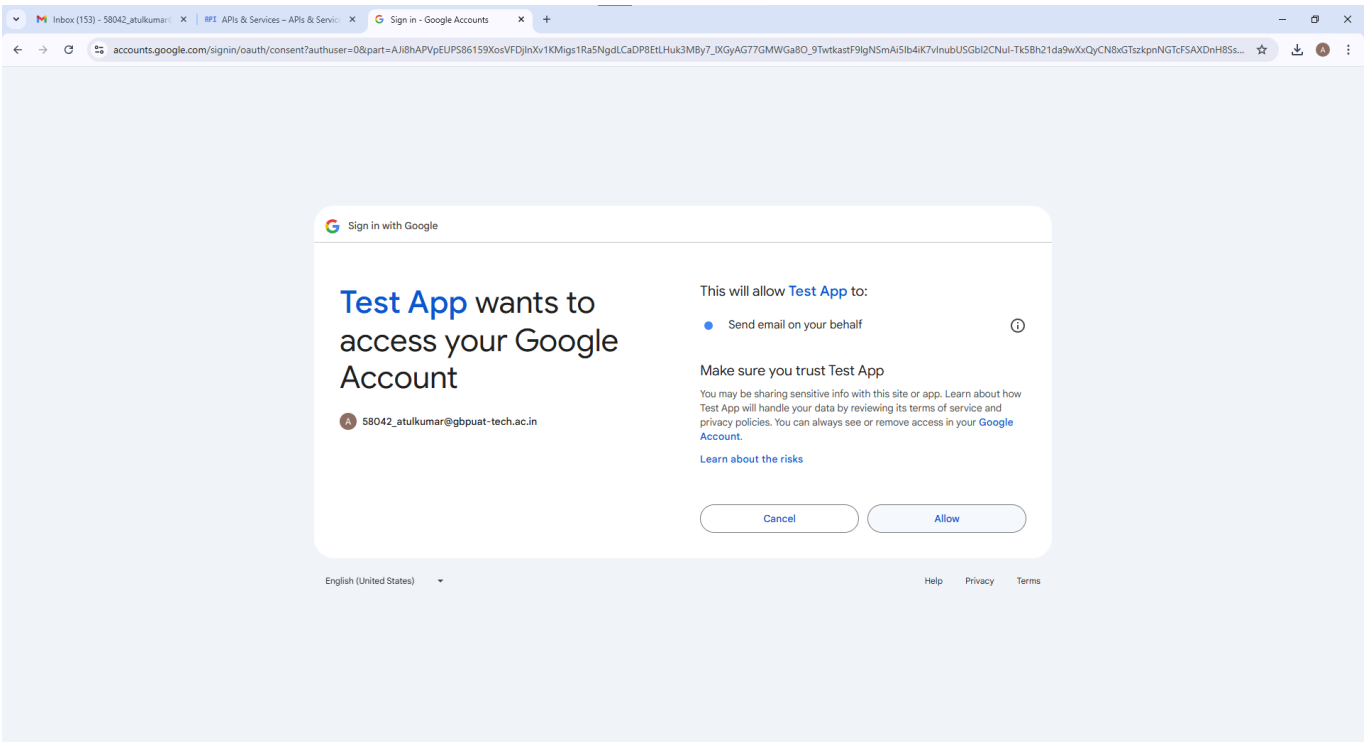
```
        msg = create_message(sender, recipient, subject, message_text)
        send_message(service, 'me', msg)

    except HttpError as error:
    #TODO(developer) - Handle errors from gmail API.
        print(f'An error occurred: {error}')



def create_message(sender, to, subject, message_text):
    """Create a message for an email.

    Args:
    sender: Email address of the sender. to: Email address of the receiver.
    subject: The subject of the email message. message_text: The text of the email
message.

    Returns:
    An object containing a base64url encoded email object. """
    message = MIMEText(message_text)
    message['from'] = sender
    message['to'] = to
    message['subject'] = subject
    return {'raw':
base64.urlsafe_b64encode(message.as_string().encode()).decode()}


def send_message(service, user_id, message):
    """Send an email message.

    Args:
    service: Authorized Gmail API service instance. user_id: User's email address.
The special value "me" can be used to indicate the authenticated user. message:
Message to be sent.

    Returns:
    Sent Message. """
    try:
        message = (service.users().messages().send(userId=user_id,
body=message).execute())
        print('Message Id: %s' % message['id'])
        return message
    except HttpError as error:
        print('An error occurred: %s' % error)


if __name__ == '__main__':
    prepare_and_send_email('atulakku99@gmail.com', 'Greeting from Global
Infoventures', 'This is a test email for our upcoming app')
```

Run the python file.

- At the first time you will be prompted to enter you Google account credentials.
- **Use the same credentials** that you used to sign up in Google Cloud.



- Allow your app



After completion of authentication you will see a message as

- The authentication flow has completed. You may close this window.