```
#!/usr/bin/env python

import csv
import email
from email import policy
import imaplib
import logging
import os
import ssl

from bs4 import BeautifulSoup


credential_path = "credentials.txt"
csv_path = "mails.csv"

logger = logging.getLogger('imap_poller')

host = "imap.gmail.com"
port = 993
ssl_context = ssl.create_default_context()


def connect_to_mailbox():
    # get mail connection
    mail = imaplib.IMAP4_SSL(host, port, ssl_context=ssl_context)

    with open(credential_path, "rt") as fr:
        user = fr.readline().strip()
        pw = fr.readline().strip()
        mail.login(user, pw)

    # get mail box response and select a mail box
    status, messages = mail.select("INBOX")
    return mail, messages


# get plain text out of html mails
def get_text(email_body):
    soup = BeautifulSoup(email_body, "lxml")
    return soup.get_text(separator="\n", strip=True)


def write_to_csv(mail, writer, N, total_no_of_mails):

    for i in range(total_no_of_mails, total_no_of_mails - N, -1):
        res, data = mail.fetch(str(i), "(RFC822)")
```

```
response = data[0]
if isinstance(response, tuple):
    msg = email.message_from_bytes(response[1], policy=policy.default)

    # get header data
    email_subject = msg["subject"]
    email_from = msg["from"]
    email_date = msg["date"]
    email_text = ""

    # if the email message is multipart
    if msg.is_multipart():
        # iterate over email parts
        for part in msg.walk():
            # extract content type of email
            content_type = part.get_content_type()
            content_disposition = str(part.get("Content-Disposition"))
            try:
                # get the email email_body
                email_body = part.get_payload(decode=True)
                if email_body:
                    email_text = get_text(email_body.decode('utf-8'))
            except Exception as exc:
                logger.warning('Caught exception: %r', exc)
            if (
                content_type == "text/plain"
                and "attachment" not in content_disposition
            ):
                # print text/plain emails and skip attachments
                # print(email_text)
                pass
            elif "attachment" in content_disposition:
                pass

    else:
        # extract content type of email
        content_type = msg.get_content_type()
        # get the email email_body
        email_body = msg.get_payload(decode=True)
        if email_body:
            email_text = get_text(email_body.decode('utf-8'))

    if email_text is not None:
        # Write data in the csv file
        row = [email_date, email_from, email_subject, email_text]
        writer.writerow(row)
    else:
        logger.warning('%s:%i: No message extracted', "INBOX", i)
```

```
def main():
    mail, messages = connect_to_mailbox()

    logging.basicConfig(level=logging.WARNING)

    total_no_of_mails = int(messages[0])
    # no. of latest mails to fetch
    # set it equal to total_no_of_emails to fetch all mail in the inbox
    N = 2

    with open(csv_path, "wt", encoding="utf-8", newline="") as fw:
        writer = csv.writer(fw)
        writer.writerow(["Date", "From", "Subject", "Text mail"])
        try:
            write_to_csv(mail, writer, N, total_no_of_mails)
        except Exception as exc:
            logger.warning('Caught exception: %r', exc)


if __name__ == "__main__":
    main()
```