```
import socket
import select
import argparse
import sys
import datetime

def start_server(PORT_english, PORT_maori, PORT_german, verbose):
    """
    This function is in essence the main function, it has many sub-functions that are used to get certain values to parse into the
    body of this function, at the bottom of this function you will find the server information, note you can start this function
    with the --verbose flag in order to see the options and arguments, run `server.py --help` for more information`.
    """

    if PORT_english in {PORT_maori, PORT_german} or PORT_maori in {PORT_english, PORT_german} or PORT_german in {PORT_english, PORT_maori}:
        print("Ports cannot be identical")
        return -1

    def check_port(PORT):
        """ Checks the port and makes sure it complies to requirements (1024 - 64000) """
        if PORT < 1024 or PORT > 64000:
            print("The Port number is not within specified range (1024 - 64000)")
            return False
        return True

    def packetCheck(packet):
        """
        checks the integrity of the packet, and makes sure that it is formatted correctly
        """
        info = [packet[i:i+2] for i in range(0, len(packet), 2)]
        MagicNo = int.from_bytes(info[0], 'big')
        PacketType = int.from_bytes(info[1], 'big')
        RequestType = int.from_bytes(info[2], 'big')
        if MagicNo != 0x497E:
            return False
        if PacketType != 0x0001:
            return False
        if RequestType != 0x0001 and RequestType != 0x0002:
            return False
        return True

    def checkRequestType(packet):
        """ Checks to see if the user wants the `date` or `time` """
        info = [packet[i:i+2] for i in range(0, len(packet), 2)]
        RequestType = int.from_bytes(info[2], 'big')
        if RequestType == 0x0001:
            return 'date'
        elif RequestType == 0x0002:
            return 'time'
        else:
            return -1

    def getDate(sock):
        """
        Allows user to form the packet the `date` string to be returned to the client
        """
        months = {
            'english': ["January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December"],
            'maori': ["Kohitatea", "Hui-tanguru", "Poutu Ě-te-rangi", "Paenga-whawha", "Haratua", "Pipiri", "Hongongoi", "Here-turi-koka", "Mahuru", "Whiringa-a-nuku", "Whiringa-a-rangi", "Hakihea"],
            'german': ["Januar", "Februar", "Marz", "April", "Mai", "Juni", "Juli", "August", "September", "Oktober", "November", "Dezember"]
        }
    MagicNo = 0x497E.to_bytes(2, 'big')
    PacketType = 0x0002.to_bytes(2, 'big')
    if sock is s_english:
        LanguageCode = 0x0001
        flag = 'english'
    elif sock is s_maori:
        LanguageCode = 0x0002
        flag = 'maori'
    elif sock is s_german:
        LanguageCode = 0x0003
        flag = 'german'
    date = datetime.datetime.today()
    LanguageCode = LanguageCode.to_bytes(2, 'big')
    year = date.year.to_bytes(2, 'big')
    language_months = months[flag]
    chosen_month = language_months[(date.month - 1)]
    month = date.month.to_bytes(1, 'big')
    day = date.day.to_bytes(1, 'big')
    hour = date.hour.to_bytes(1, 'big')
    minute = date.minute.to_bytes(1, 'big')
    if flag == 'english':
        text = "Today's date is {} {}, {}".format(chosen_month, date.day, date.year)
    elif flag == 'maori':
        text = "Ko te ra o tenei ra ko {} {}, {}".format(chosen_month, date.day, date.year)
    else:
        text = "Heute ist der {} {}, {}".format(chosen_month, date.day, date.year)

    lengthNow = len(text)
    length = lengthNow.to_bytes(1, 'big')

    bytelist = [MagicNo, PacketType, LanguageCode, year, month, day, hour, minute, length]

    out = bytearray()

    for byteset in bytelist:
        out += byteset

    out.extend(text.encode("utf-8"))

    return out

def getTime(sock):
    """
    Server uses this function to form the `time` that will be outputted to the client.
    """
    MagicNo = 0x497E.to_bytes(2, 'big')
    PacketType = 0x0002.to_bytes(2, 'big')
    if sock is s_english:
        LanguageCode = 0x0001
        flag = 'english'
    elif sock is s_maori:
        LanguageCode = 0x0002
        flag = 'maori'
    elif sock is s_german:
        LanguageCode = 0x0003
        flag = 'german'
    date = datetime.datetime.today()
    LanguageCode = LanguageCode.to_bytes(2, 'big')
    year = date.year.to_bytes(2, 'big')
    month = (date.month).to_bytes(1, 'big')
    day = date.day.to_bytes(1, 'big')
    hour = date.hour.to_bytes(1, 'big')
    minute = date.minute.to_bytes(1, 'big')
    if flag == 'english':
        text = f"The current time is {date.hour}:{date.minute}"
    elif flag == 'maori':
        text = f"Ko te wa o tenei wa {date.hour}:{date.minute}"
    else:
        text = f"Die Uhrzeit ist {date.hour}:{date.minute}"

    lengthNow = len(text)
    length = lengthNow.to_bytes(1, 'big')

    bytelist = [MagicNo, PacketType, LanguageCode, year, month, day, hour, minute, length]

    out = bytearray()

    for byteset in bytelist:
        out += byteset

    out.extend(text.encode('utf-8'))

    return out

"""
Here is where we implement the Socket API. This is the main setup for the server to receive and send
the appropriate packets.
"""
IP = socket.gethostbyname('localhost')

s_english = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
s_maori = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
s_german = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

server_sockets = [[s_english, PORT_english], [s_maori, PORT_maori], [s_german, PORT_german]]

for sock, port in server_sockets:
    if check_port(port):
        sock.bind((IP, port))
    else:
        print("Server closed: port {} isnt within range (1024 - 64000)".format(port))
        return -1

sockets = [s_english, s_maori, s_german]

while True:
    try:
        read, write, exception = select.select(sockets, [], [])
        for s in read:
            packet, source = s.recvfrom(48)
            if verbose:
                print(f"Packet recieved from {source}")
            if packetCheck(packet):
                if checkRequestType(packet) == 'date':
                    msg = getDate(s)
                    s.sendto(msg, source)
                elif checkRequestType(packet) == 'time':
                    msg = getTime(s)
                    s.sendto(msg, source)
                else:
                    msg = "Packet discarded"
                    msg = msg.encode("utf-8")
                    s.sendto(bytes(msg))
            else:
                msg = "Packet discarded"
                msg = msg.encode("utf-8")
                s.sendto(bytes(msg))
                s.close()
    except KeyboardInterrupt:
        print("")
        print("Closing Sockets...")
        for s in sockets:
                s.close()
            print("Server Closed")
            sys.exit()
            break


def Main():
    """ Main function: calls to start the server, provides options for the user """
    parser = argparse.ArgumentParser()
    parser.add_argument("PORT_English", help="The Port number to grab date/time in English.", type=int)
    parser.add_argument("PORT_Maori", help="The Port number to grab date/time in Te Aro Maori.", type=int)
    parser.add_argument("PORT_German", help="The port number to grab date/time in German.", type=int)
    parser.add_argument("-v", "--verbose", action="store_true", help="verbose output: view client requests and where they are comming from")

    args = parser.parse_args()
    if args.verbose:
        start_server(args.PORT_English, args.PORT_Maori, args.PORT_German, verbose=True)
    else:
        start_server(args.PORT_English, args.PORT_Maori, args.PORT_German, verbose=False)

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