1. Prototyping

Seeing how my project went, this one is the most expecting method I've used in my research. I wanted to get answers and insights by programming several prototypes. As a form of tinkering I was practically (trying to) program/code each day. I've had questions which I wanted to research; I chose to do that through programming. How do I do this? Why does something work the way it does? How can I get this thing to work like that? These kinds of questions got me really interested in the subject I chose, and I tried to get my answers by tinkering around with code and making prototypes.

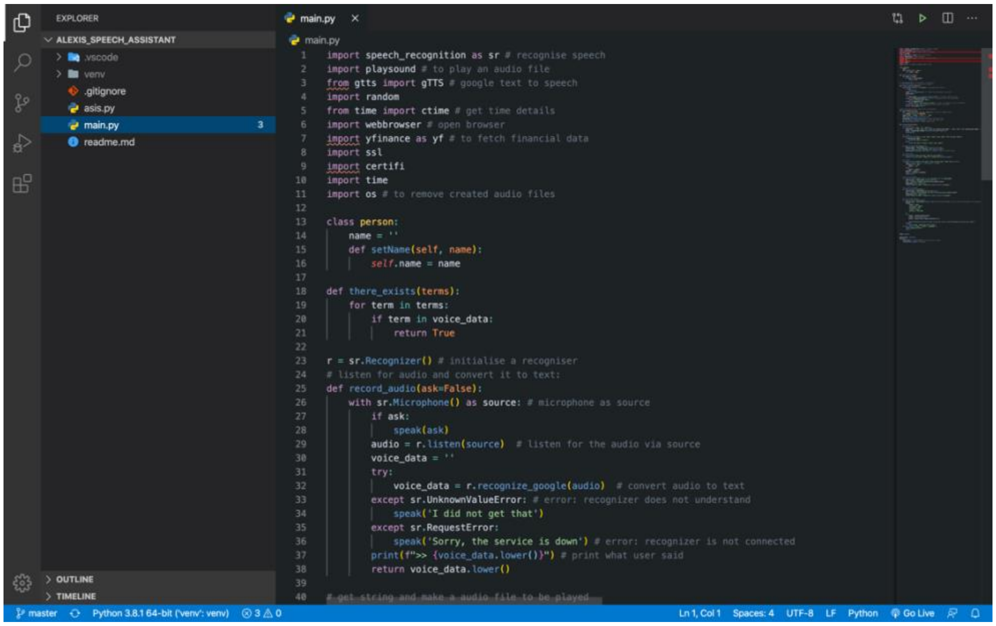
I started programming in the second stage of my project. But I actually started learning things through coding at the start of the lab already. When I look at how my research went, I started by creating a concept, and when I really knew which direction I wanted to go, I started programming.

What I really need for this method are two things. An idea and a text editor to start coding, I've mostly used Visual Studio Code and PyCharm during this research.

Since I chose programming as my main focus, every other method I've used were to be used for my programming. I tried other methods in order to help me with the programming part. Such as, using tutorials, learn to use multiple operating systems and different text editors,

Now that I look back on this method, I'm really glad I chose to do everything the way I did. I did really learn a lot in the past months, and I've gotten some really valuable insights. If I chose to do things any during research.

This method was my primary activity, I used it simultaneously with:
- watching videos online (desk research)
- Looking for information on StackOverflow (desk research)
- Using different work environments (tinkering).



```python
import speech_recognition as sr # recognise speech
import playsound # to play an audio file
from gtts import gTTS # google text to speech
import random
from time import ctime # get time details
import webbrowser # open browser
import yfinance as yf # to fetch financial data
import ssl
import certifi
import time
import os # to remove created audio files

class person:
    name = ''
    def setName(self, name):
        self.name = name

def there_exists(terms):
    for term in terms:
        if term in voice_data:
            return True

r = sr.Recognizer() # initialise a recogniser
# listen for audio and convert it to text:
def record_audio(ask=False):
    with sr.Microphone() as source: # microphone as source
        if ask:
            speak(ask)
        audio = r.listen(source)  # listen for the audio via source
        voice_data = ''
        try:
            voice_data = r.recognize_google(audio)  # convert audio to text
        except sr.UnknownValueError: # error: recognizer does not understand
            speak('I did not get that')
        except sr.RequestError:
            speak('Sorry, the service is down') # error: recognizer is not connected
        print(f">> {voice_data.lower()}") # print what user said
        return voice_data.lower()

# get string and make a audio file to be played
```