I. A title for your selection.
   A. I'm titling my selection: Mo's Orchestra

II. The group's members.
   A. It's just me, (Mohamed Ahmed, ahmedm19@msu.edu)

III. The format of your score files.
   A. My score files are present in github website in the folder called "Scorefiles"

IV. The actual score file used for your selection.
   A. My score files are present in the github website in the folder called "Scorefiles".
   B. I used *song.SCORE* for the 1-1.5 minute song
      1. This score for this song was **generated by AI**
   C. I used *song - Copy.SCORE* for the shorter pieces like the wavetable synthesizer and effects

V. One page for each component that describes the functioning of that component in detail. Indicate ownership of each component and what grading elements are supported.

<u>WaveTableSynthesizer</u>

I made a wavetable synthesizer. This synthesizer can be heard *wavetablewynth.wav* in the "Scorefiles" folder. It uses a WaveTableSynth Object in my synthesizer that receives a sine wave sample along with the current instrument. The frame is synthesized via WaveTableSynth->Generate(sample) and the frame is manipulated until returned true, just like an instrument. The wavetable itself maintains a wavetable, pretty self-explanatory. It uses it to manipulate the waveform for each iteration through instrument->Generate().

<u>Effects</u>

I implemented four effects: reverb, ring, flange, and noise gate. Each is a subclass of class BaseEffect. Each effect can process audio input by receiving an input frame, processing it, and then giving an output file. Effects are added to an effects list in synthesizer.cpp, which is then iterated over to process input. Each effect is added to the effect list via this line in the XML file. In the XML file, an effects node is included:

```
<effects effect1="reverb" effect2="ring" effect3="flange" effect4="noisegate" >
```

Effects can be played in parallel or serially, I have functions for both, as well wave file demos. See website.