# Project Plan

## Guitar Guide

A guitar song-writing and practice aid

Joseph Xavier Vinit Miranda 3035243778

Hamza Sohail Siddiqui 3035243106

# Project Background

Songwriting is a complex art, requiring not just basic musicianship and suitable comprehension of the various structures and techniques associated with music, but also copious amounts of acumen and creativity. It is decidedly daunting for a budding guitarist to engage in it; beginners, and sometimes intermediate to advanced players, often fall into a rut of learning how to play popular songs from the plethora of instructional videos found on Youtube, without fully contextualizing the musical theory behind said songs. While this does enhance their musicianship, it teaches them little in the way of composing their own works on the guitar. Those that do make an effort to learn theory and make an attempt to memorize the innumerable chord voicings and scale shapes, are soon overwhelmed by the immense influx of information; it is demonstrably difficult to both recall and then to piece together all these fragments of knowledge to actualize a coherent and compelling piece of music.

This project aims to dispel this difficulty; the application we intend to build will provide interactive, real-time support to a guitarist engaged in songwriting. Users of the application will be able to circumvent certain tedious aspects, such as staying in key and locating the most convenient chord voicings and scale positions on the fretboard, giving them full liberty to exercise their creative muscles. The application will further provide users with the opportunity to practise and memorize the aforementioned fretboard ‘shapes’ in an enjoyable and interactive manner, as opposed to the lacklustre conventional method of looking up these shapes on the internet. Moreover, the application will give compositional suggestions, which would be particularly useful to players that are not well-versed in music theory.

In relation to existing guitar education products, there are a few applications, such as Yousician, that are solely focused on teaching specific songs and numerous others that present libraries of chords and shapes in a non-interactive manner. Although these are effective tools for improving technique and speed, they do not explore the actual process of songwriting. The educational approach presented in this proposal is unprecedented, and is expected to be highly beneficial for guitarists looking to undertake their first steps into composing by providing a streamlined songwriting process.

Some features of the application will rely heavily on mapping raw audio signals to the guitar fretboard which involves not only getting the notes that are being played, but where on the fretboard it is being played. Methods that involves Computer Vision have an accuracy of about 70% when it comes to single notes being played and less than 50% when Chord progressions are being played.

# Project Objective

The objective of the project is to help amateur guitar players take their guitar playing to the next level. Often people who pick up the guitar learn how to play the basic open chords and a few songs but give up on playing once they get to the difficult aspects such as improvising, learning advanced harmony and actual music composition. We feel that as guitarists, such a software will greatly help in terms of learning new chord voicings, learning how to play scale shapes all over the fretboard and improving harmony playing.

# Project Methodology

The application will have multiple features that include finding all possible chord voicings anywhere on the fretboard, harmonic suggestions based on a given key, and will also provide scale practice routines for the user. There will also be a feature to track whether the user is playing a melody in the right scale. These features will aid the user in songwriting and would help in getting better at the guitar.

## Guitar Fretboard Mapping

As mentioned before, for certain features of the application, a full fretboard mapping of a raw audio signal to a fret on the guitar fretboard will be needed. As seen in the figure below, simple Pitch Detection Algorithms are ineffective since they are unable to distinguish between the multiple possible positions on the fretboard that vibrate with the same frequency, i.e. that play the same note.

*Octave Colour Code*

| 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|

*Fret Note*

| Open | 1F | 2F | 3F | 4F | 5F | 6F | 7F | 8F | 9F | 10F | 11F | 12F | 13F | 14F | 15F | 16F | 17F | 18F | 19F | 20F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| E4 | F | F#/Gb | G | G#/Ab | A | A#/Bb | B | C | C#/Db | D | D#/Eb | E | F | F#/Gb | G | G#/Ab | A | A#/Bb | B | C |
| B3 | C | C#/Db | D | D#/Eb | E | F | F#/Gb | G | G#/Ab | A | A#/Bb | B | C | C#/Db | D | D#/Eb | E | F | F#/Gb | G |
| G3 | G#/Ab | A | A#/Bb | B | C | C#/Db | D | D#/Eb | E | F | F#/Gb | G | G#/Ab | A | A#/Bb | B | C | C#/Db | D | D#/Eb |
| D3 | D#/Eb | E | F | F#/Gb | G | G#/Ab | A | A#/Bb | B | C | C#/Db | D | D#/Eb | E | F | F#/Gb | G | G#/Ab | A | A#/Bb |
| A2 | A#/Bb | B | C | C#/Db | D | D#/Eb | E | F | F#/Gb | G | G#/Ab | A | A#/Bb | B | C | C#/Db | D | D#/Eb | E | F |
| E2 | F | F#/Gb | G | G#/Ab | A | A#/Bb | B | C | C#/Db | D | D#/Eb | E | F | F#/Gb | G | G#/Ab | A | A#/Bb | B | C |

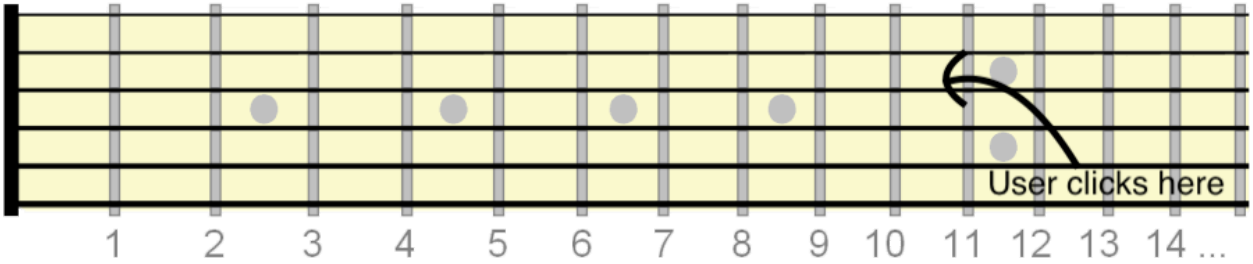
*Fret Frequency (Hz)*

| Open | 1F | 2F | 3F | 4F | 5F | 6F | 7F | 8F | 9F | 10F | 11F | 12F | 13F | 14F | 15F | 16F | 17F | 18F | 19F | 20F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 329.63 | 349.23 | 369.99 | 392.00 | 415.30 | 440.00 | 466.16 | 493.88 | 523.25 | 554.37 | 587.33 | 622.25 | 659.26 | 698.46 | 739.99 | 783.99 | 830.61 | 880.00 | 932.33 | 987.77 | 1046.5 |
| 246.94 | 261.63 | 277.18 | 193.66 | 311.13 | 329.63 | 349.23 | 369.99 | 392.00 | 415.30 | 440.00 | 466.16 | 493.88 | 523.25 | 554.37 | 587.33 | 622.25 | 659.26 | 698.46 | 739.99 | 783.99 |
| 196.00 | 207.65 | 220.00 | 233.08 | 246.94 | 261.63 | 277.18 | 193.66 | 311.13 | 329.63 | 349.23 | 369.99 | 392.00 | 415.30 | 440.00 | 466.16 | 493.88 | 523.25 | 554.37 | 587.33 | 622.25 |
| 146.83 | 155.56 | 164.81 | 174.61 | 185.00 | 196.00 | 207.65 | 220.00 | 233.08 | 246.94 | 261.63 | 277.18 | 193.66 | 311.13 | 329.63 | 349.23 | 369.99 | 392.00 | 415.30 | 440.00 | 466.16 |
| 110.00 | 116.54 | 123.47 | 130.81 | 138.59 | 146.83 | 155.56 | 164.81 | 174.61 | 185.00 | 196.00 | 207.65 | 220.00 | 233.08 | 246.94 | 261.63 | 277.18 | 193.66 | 311.13 | 329.63 | 349.23 |
| 82.41 | 87.3 | 92.49 | 97.99 | 103.83 | 110.00 | 116.54 | 123.47 | 130.81 | 138.59 | 146.83 | 155.56 | 164.81 | 174.61 | 185.00 | 196.00 | 207.65 | 220.00 | 233.08 | 246.94 | 261.63 |

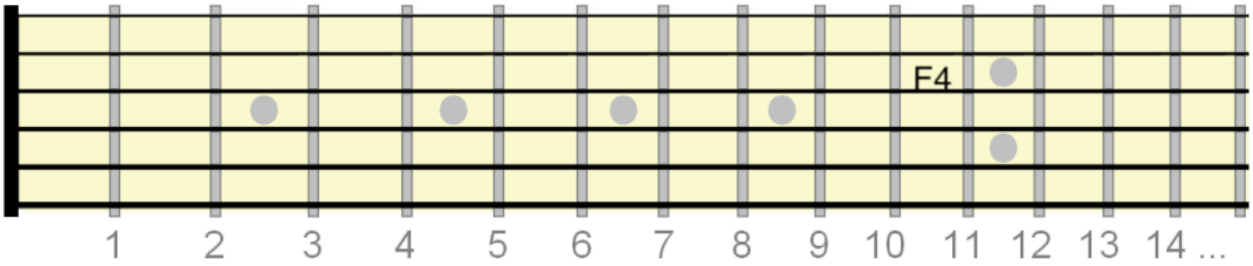
Since some features of the application are position-based, such as the Chord Finder, other techniques need to be used to detect the position of the note that the user is playing. Three such techniques can be used.

1.

**Input:** The user will be given a picture of the fretboard as input. The user will then proceed to click on the note that they wish to play and then press continue



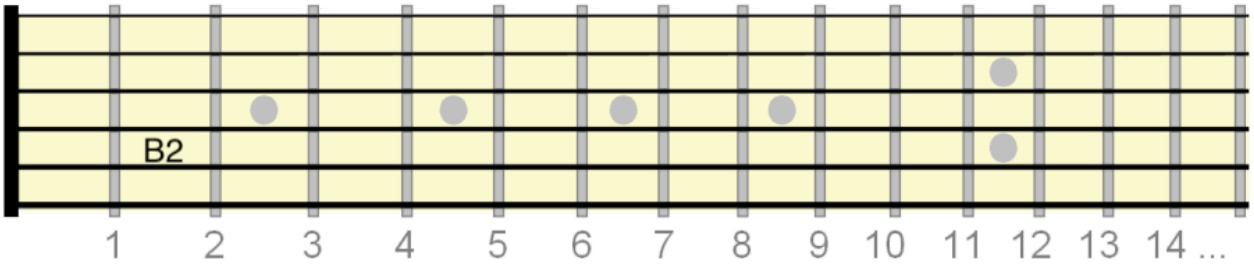

**Output:** The note that the user clicked on will be displayed.




2. The above method is not intuitive and since it is desirable to minimize the input given and also to allow the user to be able to play the note whilst simultaneously hearing it through the sound of his/her own instrument, using a pitch detection algorithm would be a preferred option. This will help in eventual fretboard memorization and would consequently, help in the learning process. A pitch detection algorithm alone can be self-sufficient only if the user inputs the string at which the note is being played, as demonstrated below.

**Input:** User plays a note on the guitar **and** inputs the string that is being played.

**Output:** The note that the user played will be displayed.



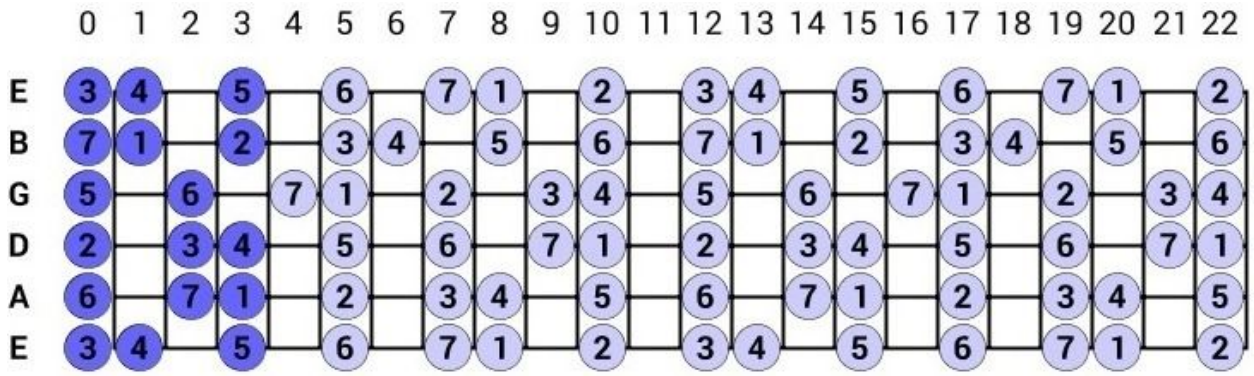

3. The third approach is the most intuitive and minimizes the inputs given by the user.
**Input:** User plays a note on the guitar
**Output:** The note that the user played will be displayed.

This project aims to achieve the third approach of detecting the position being played using a blend of Computer Vision and Pitch Detection algorithms. The note being played can be detected by firstly using a Pitch Detection Algorithm to determine the frequency of the note and then using Computer Vision techniques to rule out which one of the four possible positions on the fretboard corresponds with the user's input. The Pitch Detection Algorithm would involve Digital Signal Processing, specifically by taking the Fast Fourier Transform of the raw signal and noise removal. The key is to detect the positions being played in real time. Advanced machine learning based string detection algorithms that employ markov models to detect the fret position of the note being played have also been used in the past. Based on the requirements of our application, a suitable algorithm will be used.

# Chord Finder

Once audio signals are successfully mapped to the guitar fretboard, work will be done on building the first feature of the application: the chord finder.

Chord voicings are essentially based on numbers of the major scale, for example, a major chord will consist of the 1 3 and 5 positions of the major scale 1 being the root note. Below is the C major scale all over the fretboard. In this case C is the 1 note.




The user has to play the root note of the chord (which can be found using the note recognizer) and select what kind of chord that needs to be played. It can be a major, minor suspended, dominant, or any other suitable chord that satisfies the key. The user also provides an optional input of the number of strings the chord will consist of ranging from 3 to 6. Using an algorithm, the application will detect which shape the guitarist should play for that chord and how it should be played. Since there will be many possible combinations, we need to come up with a suitable way to a) Determine all the possibilities b) Filter from these possibilities to determine those options that are optimum and easy to play.

The application will, subsequently, show the user a few options to play the chord that needs to be played. After this, the chord shape will be added to a chord map. The user will, then, be given the option to input the next chord that he wishes to play. The software will recognize the nearest chord shape for that next chord. This next chord input can also be given by a possible harmony suggester feature. The aforementioned chord map will be the basic drawing board for constructing the entire harmony of the song.

# Harmony Suggester

In order to make logical and rich sounding chord progressions, machine learning techniques can be used to suggest harmony. This can be done by training a machine learning model based off a list of the chord progressions of many songs. Alternatively, concepts of music theory can also be used in the process to generate chord progressions that are likely to be pleasing to the ear. The application, with the incorporation of this feature, will hence serve as an introductory guide to common chord progressions that are used in western music. The feature would provide a good base for amateur guitarists to start writing simple harmony.

## Scale Check

Another feature of the application is scale check, which will be used to check if the note that is played by the user is in a given scale or not. A typical use case would be for musicians who are improvising on the scale. The software will listen to the notes that are being played and using Robust Pitch Detection Algorithms will detect the notes being played in real time. The user will also input the scale that is going to be played. If the user is playing a note that is out of the scale the software will alert the user that the melody if not in key. The challenge here is to make a robust pitch detection algorithm that works in real time with minimum latency. Digital Signal Processing will also be used to reduce the noise from the audio signal of the guitar.

# Project Schedule and Milestones

**October**

1. Market survey
2. Literature Report
3. Work on Simple Pitch Detection in real-time

**November**

1. User testing
2. Build simple Chord Finder interface
3. Start work on the harmony recommender

**December**

1. User testing
2. Start work on the string detection using Computer Vision
3. Work on other features

**January**

1. Revision of Requirements
2. Market Survey
3. Work on other features

**February**

1. Marketing
2. User testing
3. Work on features

**March-April**

1. Work on project paper
2. Revise Requirements
3. Complete Project

# References


1. http://www.hooktheory.com/blog/chord-progression-search-patterns-and-trends/
2. http://joey.scarr.co.nz/pdf/autotab.pdf
3. https://www.idc.ac.il/en/schools/cs/research/Documents/Thesis-Shir%20Goldstein.pdf