Ollscoil
Teicneolaíochta
an Atlantaigh

Atlantic
Technological
University

Pitch Detection

Project Summary

This Project is a real-time pitch detection application in which analyses audio frequencies for various applications, including and not limited to instrument tuning, note identification, speech analysis, and environmental sound classification.

The application processes audio input from pre-recorded files, live microphone audio, and external frequency tones, it applies Filtering and Fast Fourier Transforms(FFT) analysis and visualises the result in real time.

This project includes a web interface with React and Node.js to provide interactive controls for adjusting frequency ranges and displaying real-time data update.

What My Project Does

•Real-Time Audio Processing:
The system processes various audio inputs in a python environment using a .Wav file, it then applies a band-pass filter and normalizes the signal for consistent volume throughout the audio sample.

•Frequency Detection & Analysis:
Using FFT, the system converts audio to frequency data, detects peak frequencies, and identifies musical notes.

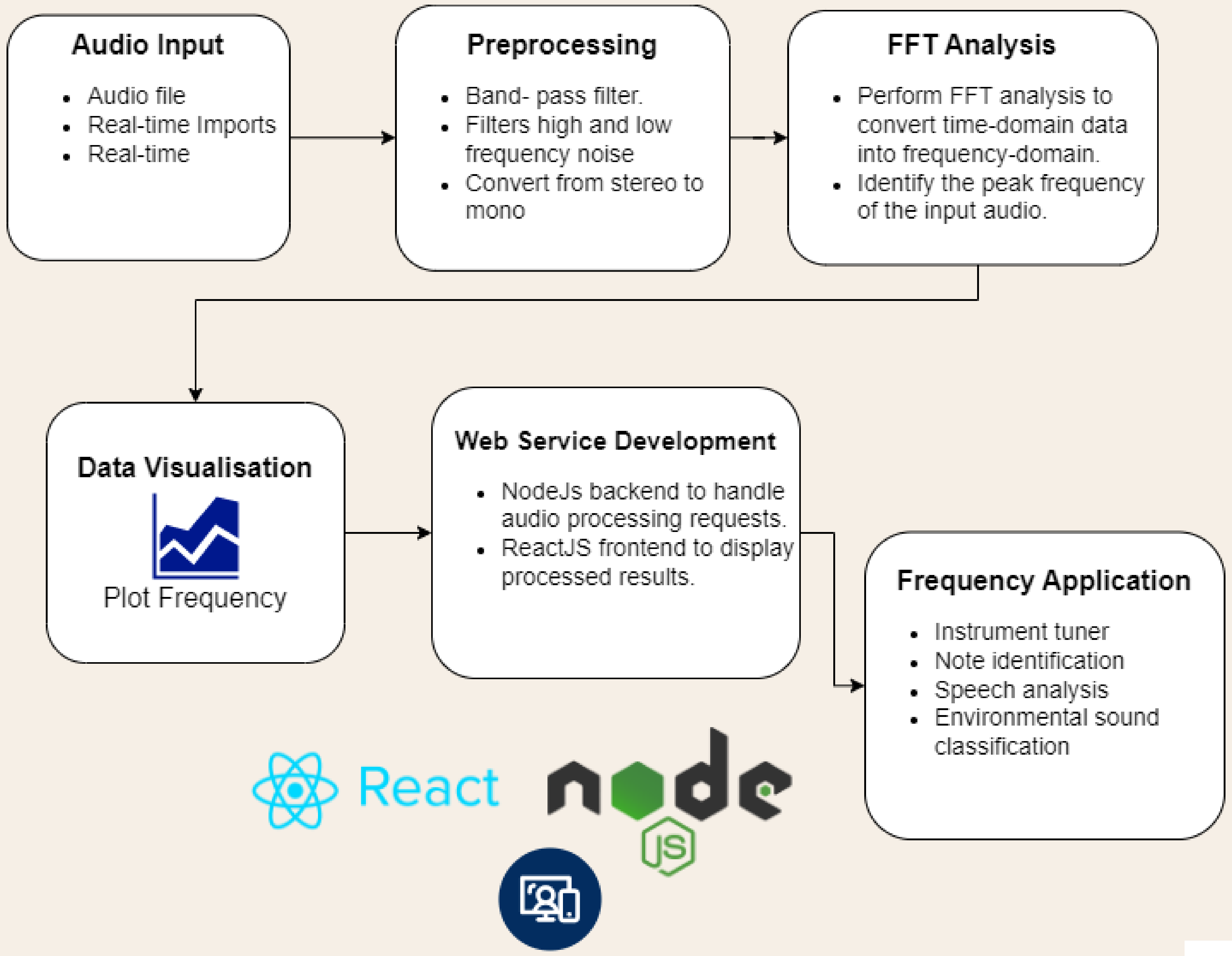
•Visual Real-Time:
An interactive graph displays real-time frequency variations, allowing users to adjust settings like frequency range.

•Web-Based Interface:
The backend of the system is built using Node.js, which processes the data, and React is used for the frontend, providing a user interface. It updates the data in real-time using a communication protocol, allowing users to interact with and visualize the pitch detection results.

Frequency-Based Applications:

- **Instrument Tuner:** Provides real-time tuning information such as 'Sharp', 'Flat'.
- **Note Identification:** Detects musical notes and displays corresponding frequencies.
- **Speech Analysis:** Analyses speech pitch and variation.
- **Sound Classification:** Differentiates between sound sources and classifies notes with machine learning.

Project Architecture



Audio Input
- Audio file
- Real-time Imports
- Real-time

Preprocessing
- Band- pass filter.
- Filters high and low frequency noise
- Convert from stereo to mono

FFT Analysis
- Perform FFT analysis to convert time-domain data into frequency-domain.
- Identify the peak frequency of the input audio.

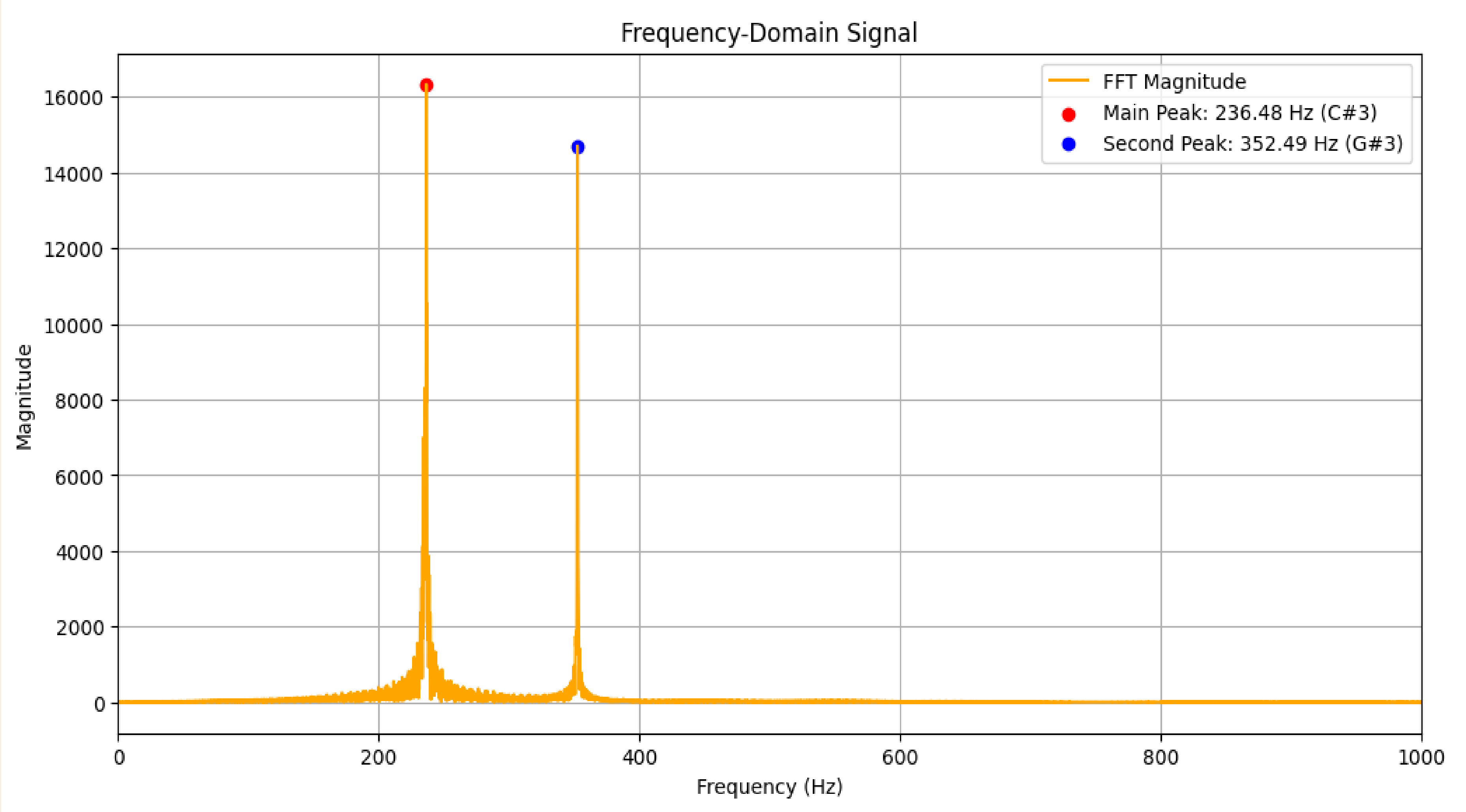
Data Visualisation
Plot Frequency

Web Service Development
- NodeJs backend to handle audio processing requests.
- ReactJS frontend to display processed results.

Frequency Application
- Instrument tuner
- Note identification
- Speech analysis
- Environmental sound classification

Data Visualisation

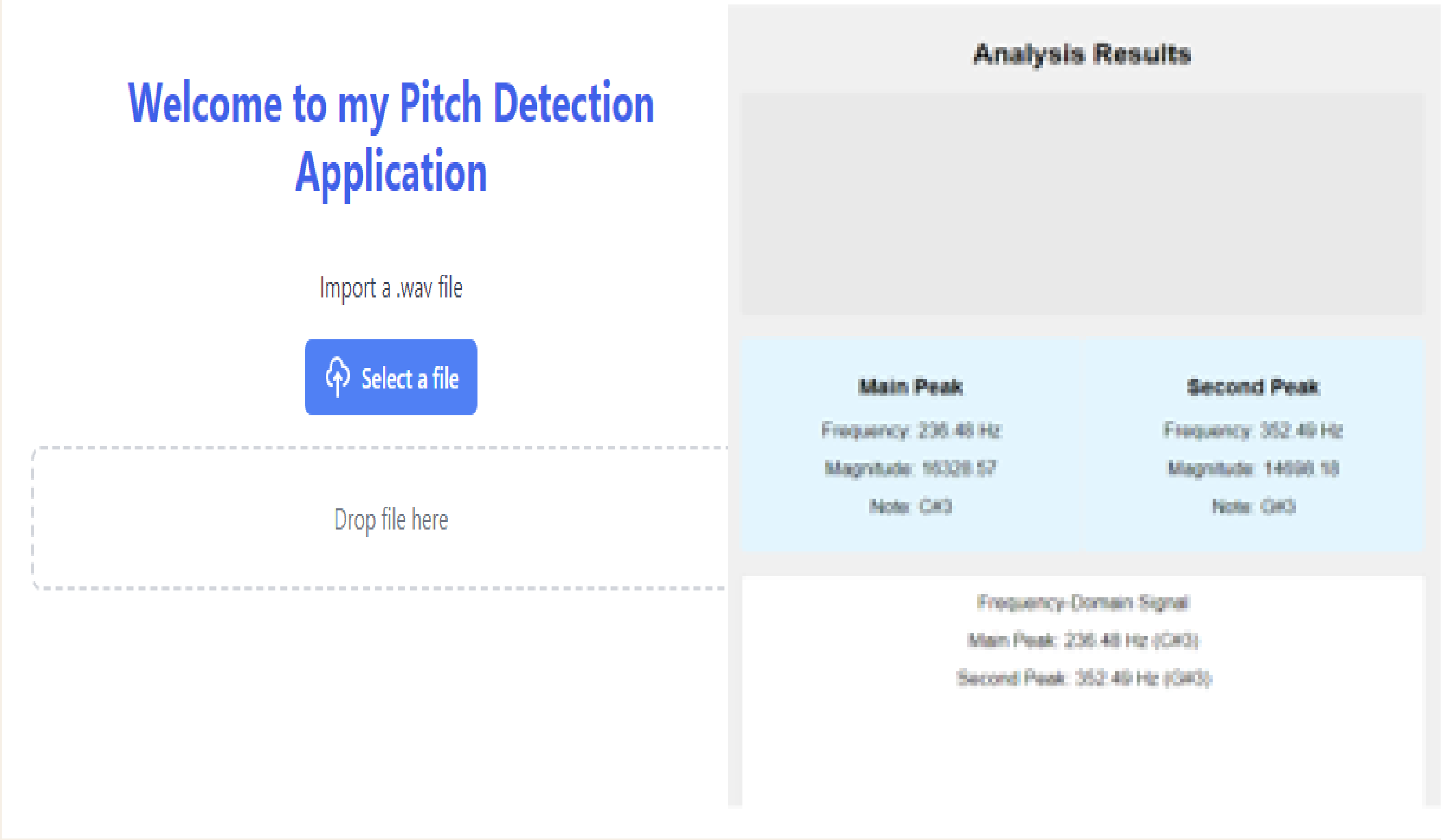
```
Main Peak: Frequency: 236.48 Hz, Magnitude: 16328.57, Note: C#3 (236.47744648318044 Hz)
Second Peak: Frequency: 352.49 Hz, Magnitude: 14698.18, Note: G#3 (352.491876911315 Hz)
```



Technologies and Tools

•Python pandas – for processing data and generating CSV reports.
•Matplotlib – for plotting graphs and visualizing frequency spectrum in Python.
•NumPy – for implementing the FFT algorithm and processing frequency-domain data.
•Librosa – for pitch detection and musical note identification.
•Git – version control system for managing codebase.
•React Native – for building mobile apps.
•FastAPI – backend server to handle requests and audio processing.
•React.js – frontend library for building dynamic user interfaces.

Web Server



Welcome to my Pitch Detection Application

Import a .wav file

Select a file

Drop file here

Analysis Results

Main Peak
Frequency: 236.48 Hz
Magnitude: 16328.57
Note: C#3

Second Peak
Frequency: 352.49 Hz
Magnitude: 14698.18
Note: G#3

Frequency-Domain Signal
Main Peak: 236.48 Hz (C#3)
Second Peak: 352.49 Hz (G#3)

ANALOG DEVICES
AHEAD OF WHAT'S POSSIBLE™