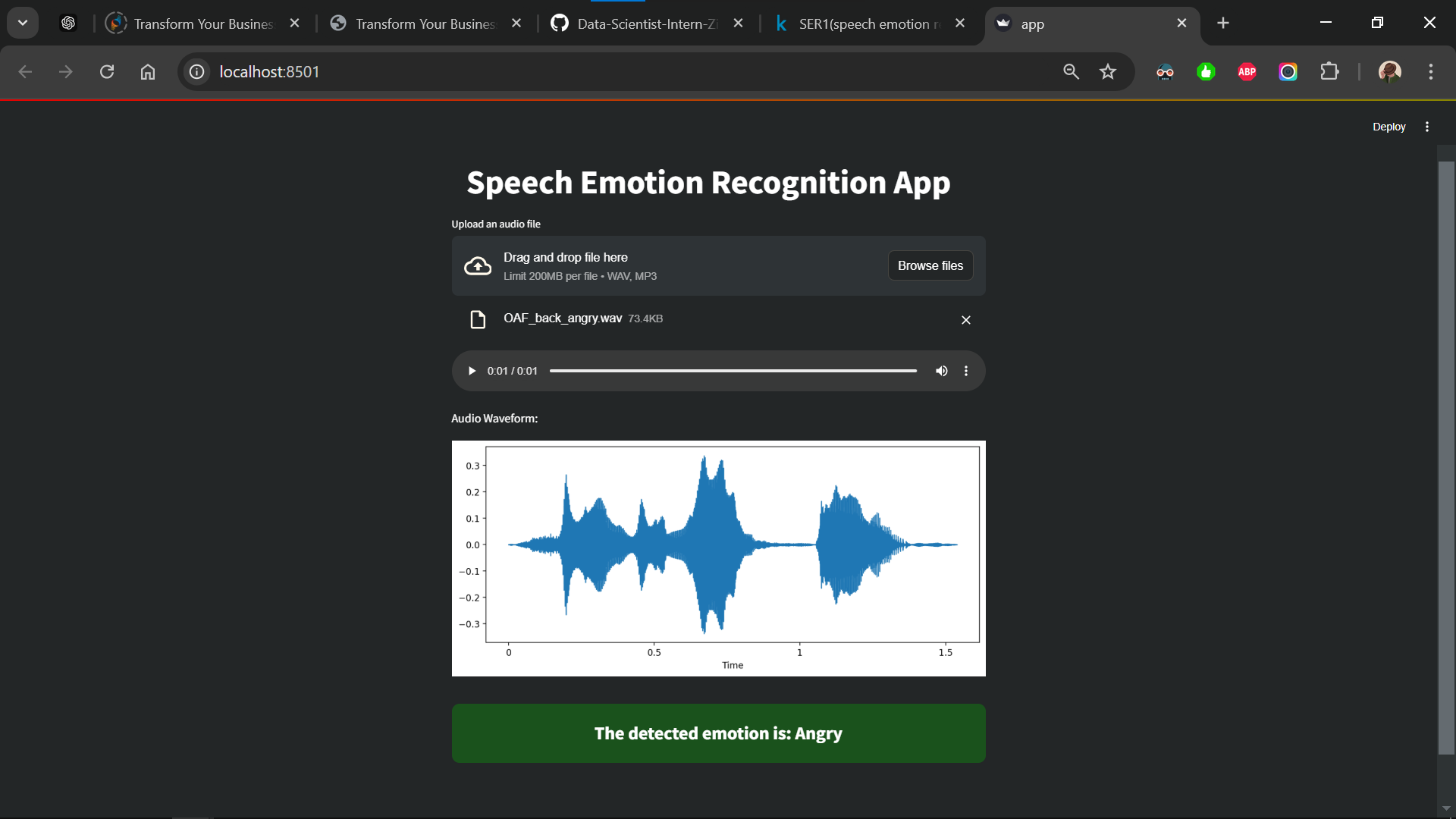
**Speech Emotion Recognition (SER) Deep Learning using LSTM**



**1. Project Overview**

This project aims to classify emotions from speech audio data using advanced machine learning techniques. By leveraging the TESS (Toronto Emotional Speech Set) dataset, the system identifies emotions such as happiness, sadness, anger, and neutrality from audio recordings.

The SER system was developed to address real-world challenges in understanding user sentiments across industries like customer service, mental health, and entertainment.

## ****2. Business Understanding****

### **Business Problem**

### In today’s competitive environment, businesses face challenges in:

* Understanding customer sentiment.
* Enhancing user experience.
* Improving engagement across various platforms.

Traditional methods often overlook emotional nuances in customer interactions, missing opportunities for innovation. SER systems can transform this by offering real-time emotion detection to improve decision-making and customer satisfaction.

### **Business Solution**

### A Speech Emotion Recognition (SER) system was developed using machine learning techniques, including:

* **MFCCs** for feature extraction.
* **LSTM neural networks** for classification.

This automated solution helps businesses enhance customer insights, operational efficiency, and overall engagement.

## ****3. Dataset****

### **TESS (Toronto Emotional Speech Set)**

* **Description**: Contains speech recordings in various emotional tones.
* **Emotions Covered**: Happiness, sadness, anger, fear, and neutrality.
* **Annotations**: Each recording is labeled for accurate training and evaluation.

**4. Tech Stack**

* **Programming Languages**: Python
* **Libraries and Frameworks**: NumPy, Pandas, Matplotlib, Seaborn, TensorFlow, Keras, Librosa
* **Other Tools**: Git, Anaconda, Kaggle Notebook
* **Deployment Tool**: Streamlit

Here’s a structured and detailed documentation template for your Speech Emotion Recognition (SER) project.

## ****3. Dataset****

### **TESS (Toronto Emotional Speech Set)**

* **Description**: Contains speech recordings in various emotional tones.
* **Emotions Covered**: Happiness, sadness, anger, fear, and neutrality.
* **Annotations**: Each recording is labeled for accurate training and evaluation.

## ****4. Tech Stack****

* **Programming Languages**: Python
* **Libraries and Frameworks**: NumPy, Pandas, Matplotlib, Seaborn, TensorFlow, Keras, Librosa
* **Other Tools**: Git, Anaconda, Kaggle Notebook
* **Deployment Tool**: Streamlit

## ****5. Methodology****

### **5.1 Data Preprocessing**

* **Loading Audio Data**: Using Librosa for audio file handling.
* **Label Extraction**: Extracting emotions from file names.
* **Waveforms and Spectrograms**: Visualization of audio signals.
* **Feature Extraction**:
  + Extracted **MFCCs (Mel-Frequency Cepstral Coefficients)**, which represent the power spectrum of the audio signal.

### **5.2 Model Development**

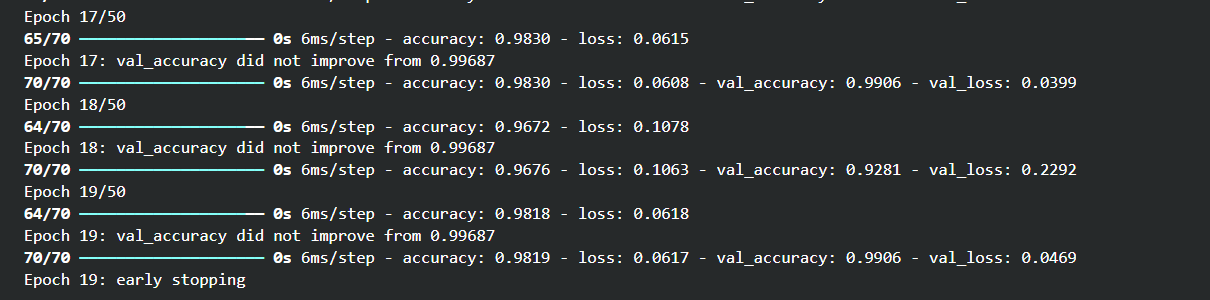
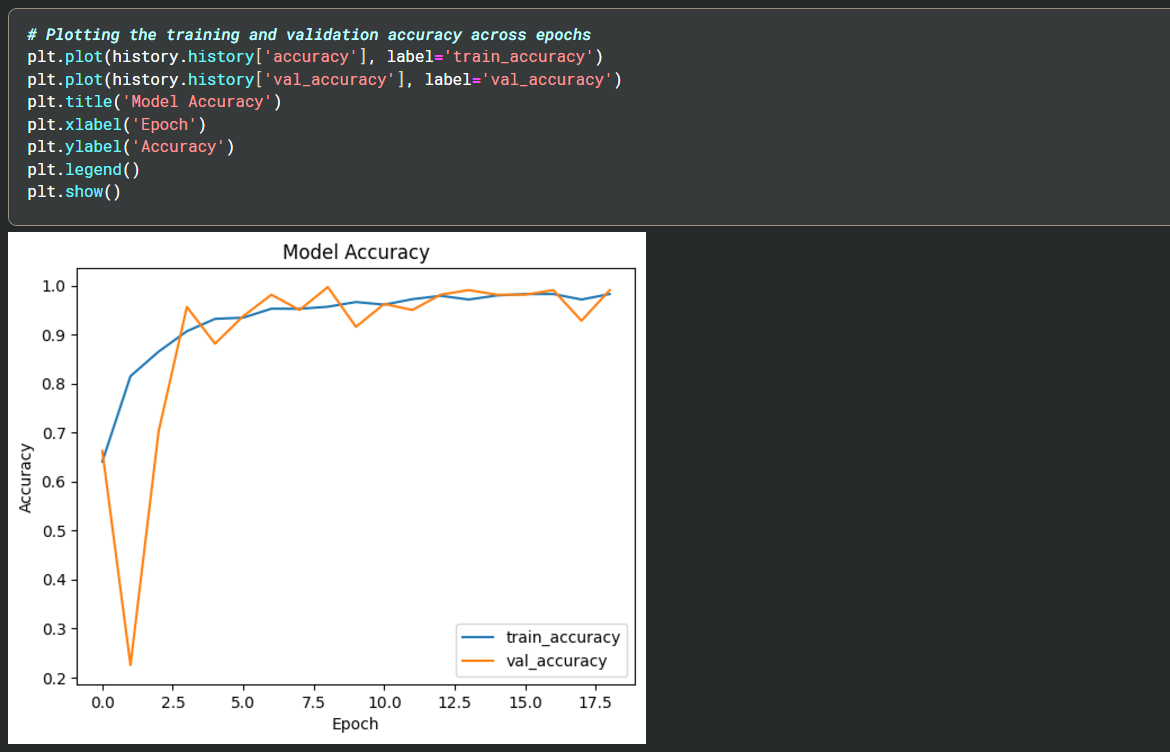
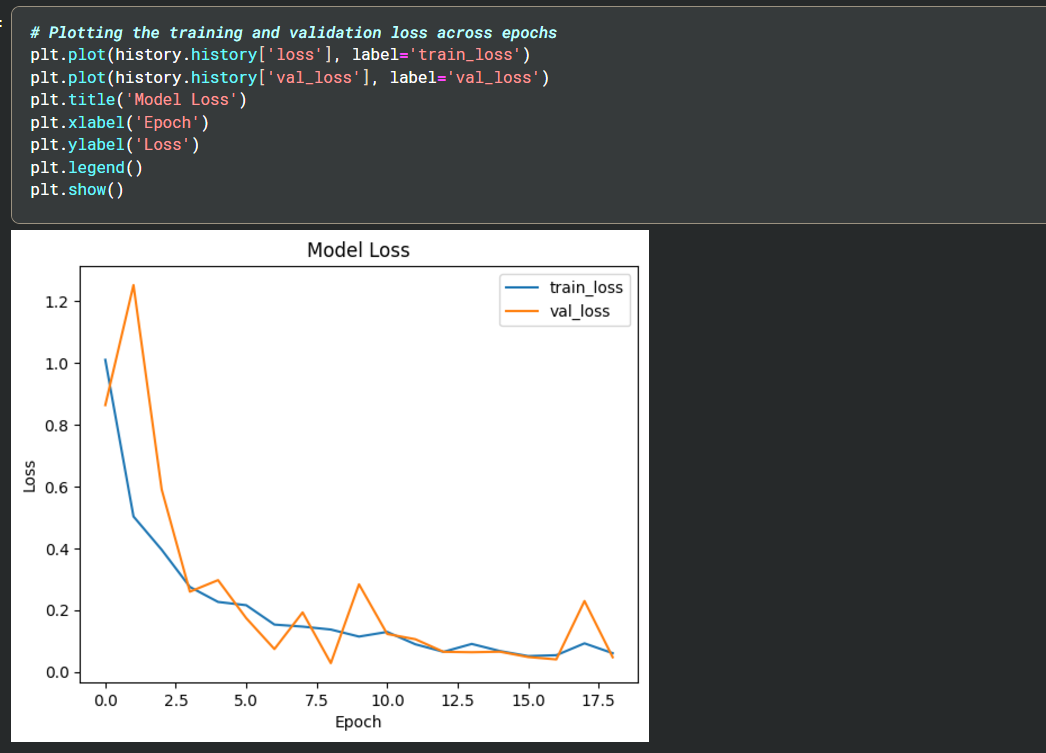
* **Preprocessing**:
  + Data normalization for better performance.
  + One-hot encoding of target labels.
* **Model Architecture**:
  + LSTM-based architecture with dropout layers for regularization.
  + Dense layers for classification.
* **Optimization**:
  + Adam optimizer with a learning rate of 1e-4.
  + Categorical cross-entropy as the loss function.
* **Training**:
  + 50 epochs with a batch size of 64.
  + Validation split to monitor overfitting.

### **5.3 Data Augmentation**

### To improve model generalization:

* **Time Stretching**: Stretching or compressing the audio.
* **Noise Injection**: Adding random noise to the signal.
* **Shifting**: Temporal shifting of audio data.

**6. Results**

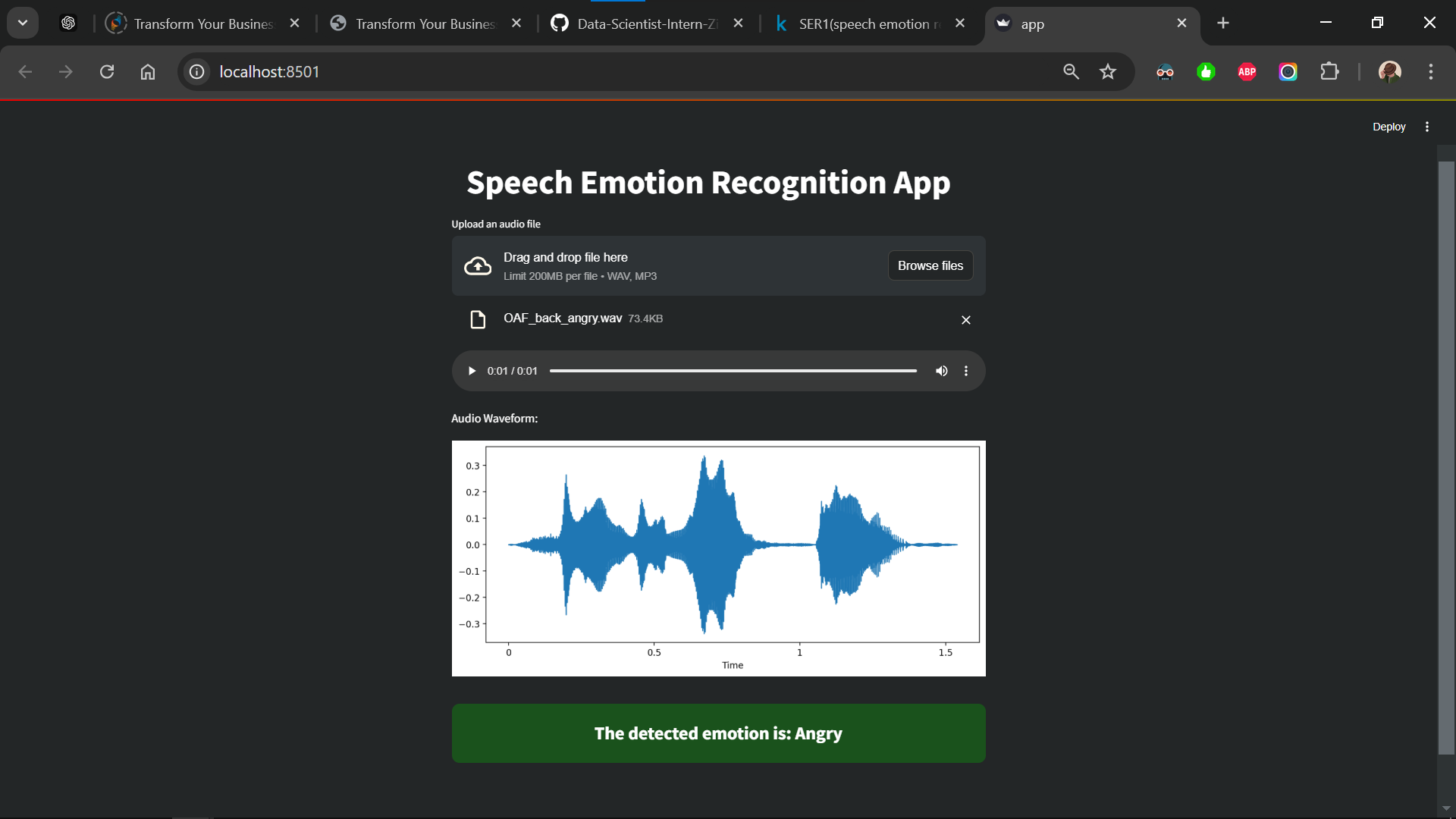
* **Training Accuracy**: ~0.9819%
* **Validation Accuracy**: ~0.9906% (Initial Overfitting Detected) 
* **Training Accuracy and validation accuracy**: Plotting the training and validation accuracy across epoch
* 
* **Training and Testing Error/loss** : Plotting the training and validation loss across epochs 

### **Improvements**

### To mitigate overfitting:

1. Added data augmentation techniques.
2. Reduced model complexity by adjusting dense layers.
3. Increased dropout rates for better regularization.

**7. Deployment**

* The SER model was deployed using **Streamlit**, creating an interactive web interface for real-time emotion detection. 

## ****8. Project Links****

* **GitHub Repository**: <https://github.com/SHINU4RATHOD/Data-Scientist-Intern-Zidio-Development>
* **YouTube Video**: <https://www.youtube.com/watch?v=bngfi7Qy-Rs&list=PLPILFCzuvU7fMzEzxZ4PxeFw1kSQ2bBFr&index=2>

## ****9. Key Learnings****

* **End-to-End Development**: Gained hands-on experience in data preprocessing, model building, and deployment.
* **Audio Signal Processing**: Learned to extract and analyze audio features like MFCCs.
* **Model Optimization**: Implemented techniques to handle overfitting and improve generalization.

## ****10. Future Scope****

* **Real-time Implementation**: Integrate the model with real-time speech systems.
* **Extended Dataset**: Train on larger and more diverse datasets for better accuracy.
* **Multilingual Support**: Adapt the model to recognize emotions in multiple languages.

**Thank you 😊**