



Joint Tech Internship Community Program

Team Name	Hello_World
Team Members	Vaitheeshwari J(22ALR108) Sivaharini K (22ALR093) Karunya S (22CSR090)
College Name	Kongu Engineering College
Assignment No.	2

Assignment Title: Voice-Based Flight Ticket Booking

PROBLEM DESCRIPTION :

The application is designed to create a voice-activated flight booking system using Flask. Users upload an audio file containing flight preferences (such as origin and destination), which is converted to text using Google Speech Recognition. The extracted text is then processed with spaCy and Cohere to identify essential details like the user's name, flight origin, destination, and preferences. The system retrieves flight availability from a third-party API and presents options to the user. Upon selecting a flight and providing an email, the user receives a booking confirmation via email with the flight details. This handles audio file uploads, text processing, flight retrieval, and email notifications.

Overview of Core Libraries and Terminologies

1. Flask: In this application, Flask serves as the core web framework, managing the routing, handling HTTP requests, and rendering HTML templates. It facilitates user interactions such as uploading audio files, selecting flights, and viewing confirmation pages, ensuring a smooth and interactive user experience.

2. SpeechRecognition: The SpeechRecognition library is utilized to convert uploaded audio files into text using the Google Speech Recognition API. This transcribed text forms the basis for extracting essential user information like name, flight origin, and destination, enabling the application to process voice commands effectively.

3. SoundFile: SoundFile is employed to read and convert various audio file formats into WAV format. This conversion is crucial for compatibility with the SpeechRecognition library, ensuring that the audio data can be accurately transcribed into text regardless of the original file format.

4. Dotenv: Dotenv is used to load sensitive environment variables from a .env file. This includes credentials for sending emails and API keys for services like Cohere. By externalizing these configurations, Dotenv ensures that sensitive information remains secure and is not exposed within the codebase.

5. smtplib: The smtplib library facilitates the sending of confirmation emails to users after successful flight bookings. By leveraging the SMTP protocol, the application can send personalized emails containing flight details and user preferences, ensuring users receive timely and accurate booking confirmations.

6. Cohere: Cohere's Natural Language Processing (NLP) capabilities are integrated to enhance the extraction of user data from transcribed text. By utilizing Cohere's language models, the application can perform more sophisticated text analysis, improving the accuracy and reliability of information extraction compared to rule-based methods alone.

7. email.mime.text.MIMEText & email.mime.multipart.MIMEMultipart: These modules from Python's email library are used to construct well-formatted MIME (Multipurpose Internet Mail Extensions) email messages. MIMEText handles the plain text content of the email, while MIMEMultipart allows for the combination of multiple parts (such as text and HTML) within a single email, ensuring that confirmation emails are both informative and professionally formatted.

8. Werkzeug's secure_filename: The secure_filename function from Werkzeug is utilized to sanitize and secure uploaded file names. This prevents potential security vulnerabilities by

ensuring that filenames do not contain malicious content or path traversal characters, thereby safeguarding the server from unauthorized access or file manipulation.

9. spaCy: spaCy is a powerful NLP library used to perform advanced text processing and entity recognition on the transcribed audio text. It identifies and categorizes entities such as names, dates, and locations, which are essential for accurately extracting user information and preferences. spaCy enhances the application's ability to understand and interpret natural language inputs effectively.

10. Session Management in Flask: Flask's session management is leveraged to store user data, available flights, and itinerary details across different routes within the application. This ensures that user information persists throughout the booking process, enabling seamless transitions between uploading audio, selecting flights, and receiving confirmations without losing context.

11. File Handling with Flask: The application uses Flask's built-in file handling capabilities to manage the upload and storage of audio files. By configuring upload folders and enforcing file size limits, the application ensures efficient and secure handling of user-uploaded content, maintaining server performance and integrity.

Comprehensive Workflow for a Voice-Activated Flight Booking System

1. Audio File Upload:

- The user accesses the homepage and uploads an audio file (in formats like WAV, MP3, or M4A) containing a spoken flight booking request.
- The file is securely stored in the server's upload folder and is ready for further processing.

2. Audio Conversion to WAV Format:

- To ensure compatibility with the speech-to-text engine, the uploaded audio file is converted to the WAV format using the SoundFile library. This step ensures the application can consistently process various audio formats.

3. Speech-to-Text Conversion:

- The converted audio file is processed using the SpeechRecognition library, which leverages Google Speech Recognition API to convert the user's spoken words into text. This step captures the booking details the user provided verbally.

4. Data Extraction Using spaCy and Cohere:

- The transcribed text is analyzed with spaCy and Cohere to extract essential user information such as name, flight details, and preferences. The `extract_user_data` function combines spaCy and Cohere to extract user data.

5. Flight Availability Check via External API:

- Using the extracted flight origin and destination, the system makes a request to an external API(postman API) to retrieve available flights for the user's desired route. Only the relevant flights matching the user's request are displayed.

6. Flight Selection and Email Submission:

- The user is presented with a list of available flight options and prompted to select a preferred flight. Additionally, the user provides their email address, which will be used for booking confirmation and communication.

7. Sending Confirmation Email:

- The `smtplib` library is used to send a personalized confirmation email to the user, containing all the details of their booking: flight number, departure date, seat preference, and meal preference. The email credentials are securely retrieved from environment variables using the `dotenv` library, ensuring sensitive information remains protected.

8. Final Booking Confirmation:

- After the email is sent, the user is directed to a confirmation page where they can review the details of their flight and the email address to which the confirmation was sent. This provides a final validation step for the user.

Design Architecture

You can explore the complete architecture of our project by visiting [Design Architecture](#). It provides a detailed overview of the system's components and workflow, showcasing how the voice-based flight ticket booking process is structured.

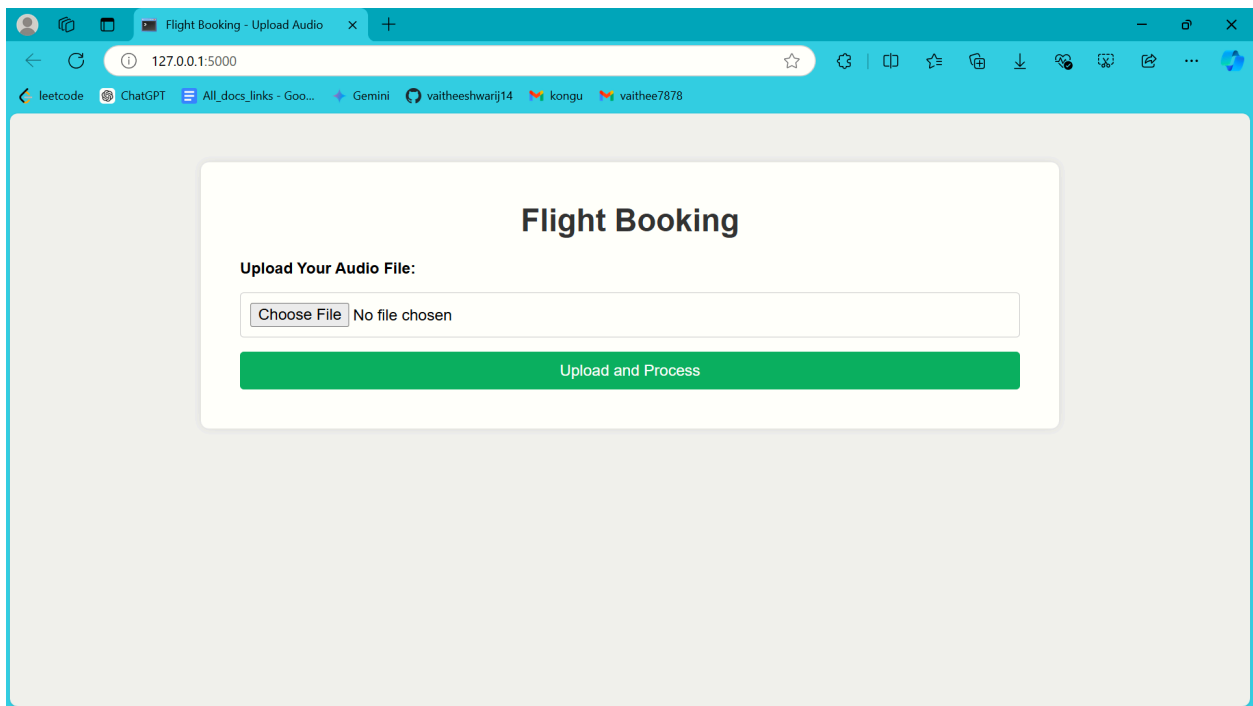
Git link

Use this repository URL for the reference of code and execution and follow the instructions in the README.md

Repository URL: https://github.com/vaitheeshwarij14/Flight_ticket_booking.git

Screenshots

Run “python app.py”



Choose the audio file and click on the upload and process.

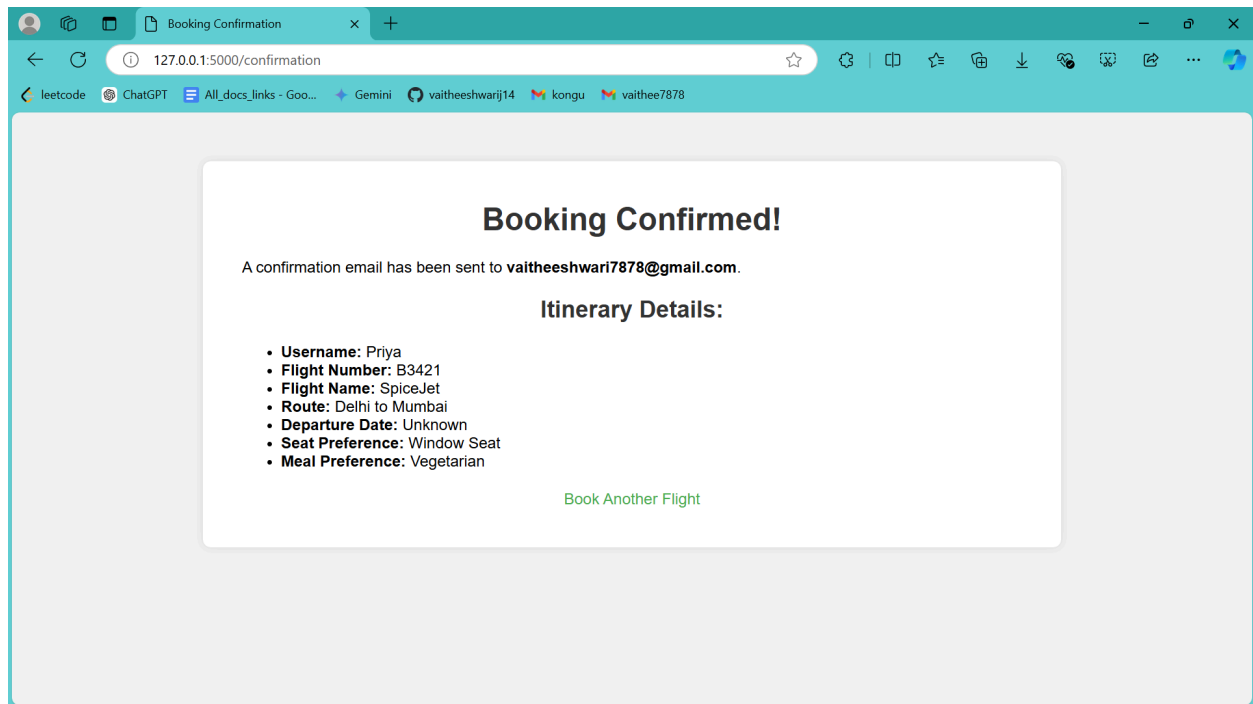
The screenshot shows a web browser window with the title "Flight Booking - Upload Audio". The address bar displays "127.0.0.1:5000". The browser's tab bar includes "leetcode", "ChatGPT", "All_docs_links - Goo...", "Gemini", "vaitheeshwari14", "kongu", and "vaithee7878". The main content area features a white card with the heading "Flight Booking". Below this heading is the instruction "Upload Your Audio File:". A file selection box contains a "Choose File" button and the text "audio_file.mp3". Below the file box is a green button labeled "Upload and Process".

Choose a flight from the available options and provide the recipient's email address and click on the book flight and send confirmation.

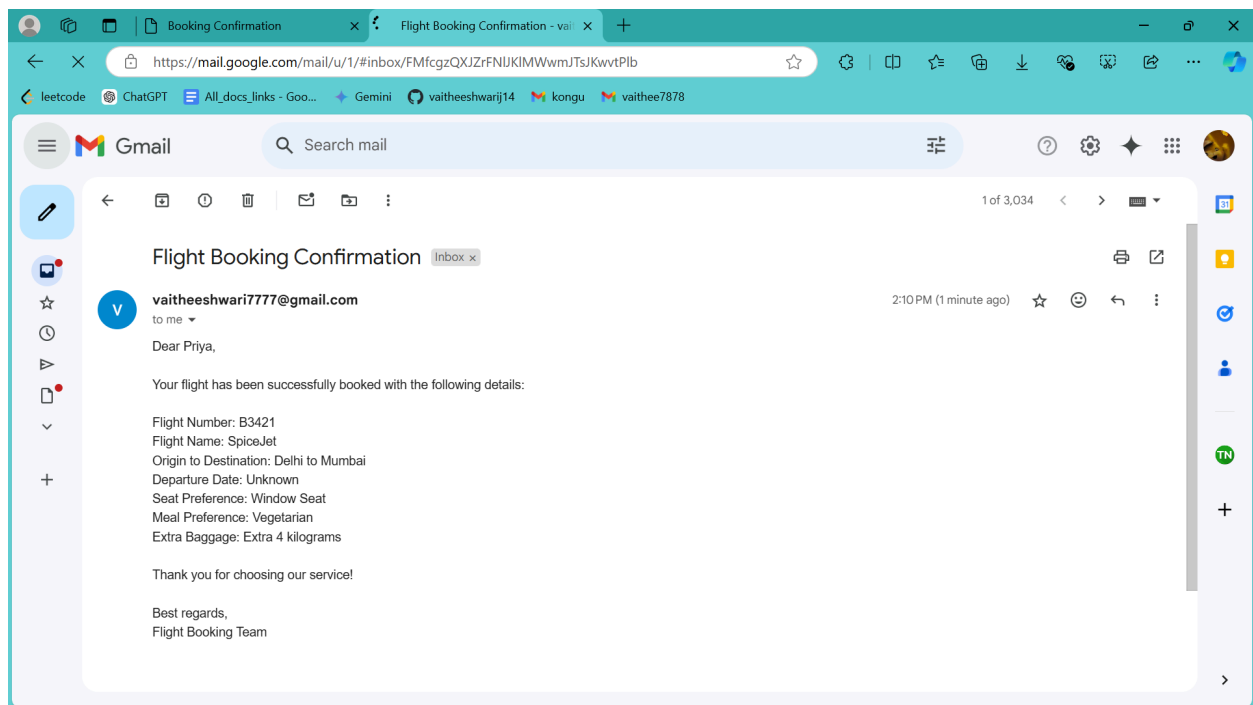
The screenshot shows a web browser window with the title "Available Flights". The address bar displays "127.0.0.1:5000/select_flight". The browser's tab bar includes "leetcode", "ChatGPT", "All_docs_links - Goo...", "Gemini", "vaitheeshwari14", "kongu", and "vaithee7878". The main content area features a white card with the heading "Available Flights from Delhi to Mumbai". Below this heading is the instruction "Converted Text:". The converted text reads: "hi I would like to book a flight ticket my name is Priya and I am born on May 20th 1998 I planned to travel from Delhi to Mumbai by passport number is 1234567 and I prefer a window seat I would like to have a vegetarian meal during the flight additionally I would like to carry extra 4 kilograms of baggage can you help me to book a flight ticket thank you". Below the text is the instruction "Available Flights:". A table with three columns: "Select", "Flight Number", and "Flight Name". The table contains two rows: the first row has a selected radio button, "B3421", and "SpiceJet"; the second row has an unselected radio button, "B3421", and an empty cell. Below the table is the instruction "Recipient Email Address:" followed by a text input field containing "vaitheeshwari7878@gmail.". At the bottom is a green button labeled "Book Flight and Send Confirmation".

Select	Flight Number	Flight Name
<input checked="" type="radio"/>	B3421	SpiceJet
<input type="radio"/>	B3421	

This window confirms that your booking is complete and reminds you to check your email for the confirmation details.



This is the confirmation email that you will receive.



Conclusion :

In conclusion, this voice-activated flight booking system provides a smooth and efficient way for users to book flights using voice commands. Starting with the upload of an audio file, the system converts speech into text and extracts important details like flight origin, destination, and preferences. It checks for available flights through an external API, allows users to select a flight, and then sends a confirmation email using secure credentials. With technologies like SpeechRecognition, cohere, spaCy, and smtplib working together, the system ensures a user-friendly, secure, and reliable booking process from start to finish.