INTERNSHIP REPORT

Submitted By

SANDHIYA.A

Reg.No:511822104038

in partial fulfilment of the requirements for the Award of Degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

PODHIGAI COLLEGE OF ENGINEERING AND TECHNOLOGY

TIRUPATTUR-635601

ANNA UNIVERSITY: CHENNAI-600 02

JUNE-2025

ANNA UNIVERSITY: CHENNAI-600 025

Department of Computer Science and Engineering
PODHIGAI COLLEGE OF ENGINEERING AND TECHNOLOGY

(Approved by AICTE New Delhi| Affiliated to Anna University, Chennai)

TIRUPATTUR-635601

July 2025

BONAFIDE CERTIFICATE

This is to certify that summer internship report is the Bonafide work of **SANDHIYA. A (511822104038)** in partial fulfilment of the award of the Degree of Computer Science and Engineering of

Podhigai College of Engineering and Technology during the year 2025 - 2026.

SUPERVISOR **HEAD OF THE**
(Mr.M.KESAVAN) **DEPARTMENT**

Submitted for the viva voce examination held on

____

DECLARATION

I affirm that the Summer Internship reoprt from the company of

DiyanTech Soluctions Pvt Ltd submitted in partial fulfilment for the award of **Podhigai College of Enginnering and Technology-Tirupattur,Computer Science and Engineering degree** is the original work carried out by me.

(Signature of the candidate)

SANDHIYA A (511822104038)

I certify that the declarations made above by the candidate are

(Signature of the Guide)

Mr.KESAVAN,B.E.,

ACKNOWLEDGEMENT

I wish to express my sincere thanks to all those who were involved in the completion of this project.

I also express my profound thank to our Chairman **Thiru.Rtn.K.C.EZHILARASAN** for their support and encouragement.

I offer my sincere thanks to our Principal **Dr.M.PRABAGARAN,M.E.,Ph.D.,**for their valuable suggestions which edified us with zest to carry out the work.

I offer my sincere thanks to **Mr.A.RAJ GANESH , M.E.,** our beloved Head of the Department of Computer Science and Engineering , **Podhigai College Of Engineering And Technology-Tirupattur (635601)**, for his encouragement.

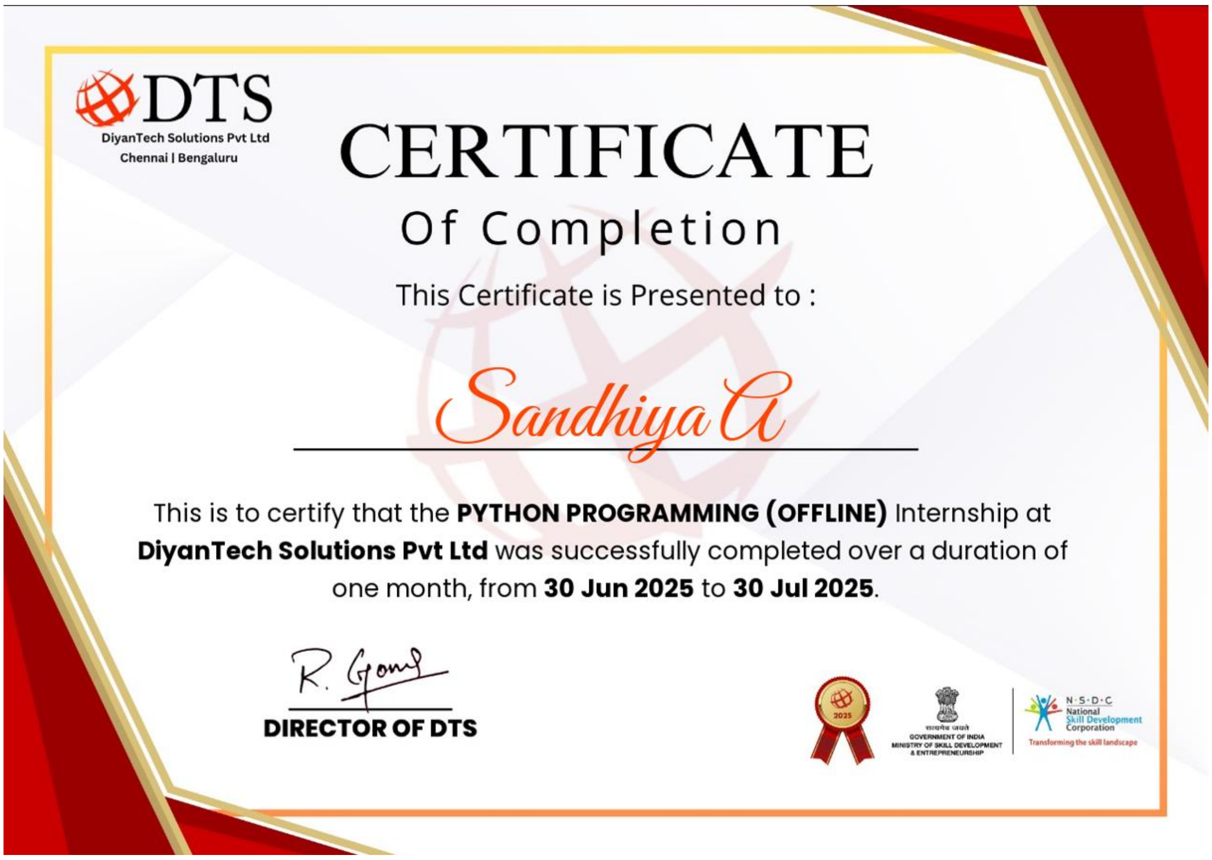
I especially thanks **Mr.M.KESAVAN B.E.,** our guide for assisting us

with his valuable suggestions and technical guidance to complete this Internship.

I express my deepest gratitude to **DIYAN TECH SOLUTOINS PVT LTD** for providing me with the opportunity to work as an internship.

My sincere thanks to my mentors and colleagues, whose guidance and support great contributed to the successful completion of this internship.

COMPANY CERTIFICATE



DTS
DiyanTech Solutions Pvt Ltd
Chennai | Bengaluru

CERTIFICATE
Of Completion

This Certificate is Presented to :

Sandhiya A

This is to certify that the **PYTHON PROGRAMMING (OFFLINE)** Internship at **DiyanTech Solutions Pvt Ltd** was successfully completed over a duration of one month, from **30 Jun 2025** to **30 Jul 2025**.

[illegible]
DIRECTOR OF DTS

2025

GOVERNMENT OF INDIA
MINISTRY OF SKILL DEVELOPMENT & ENTREPRENEURSHIP

N·S·D·C
National Skill Development Corporation
Transforming the skill landscape

INTERNSHIP COMPLETION CERTIFICATE

DTS

DiyanTech Solutions Pvt Ltd

SY NO 35/2,2nd Floor, Reliance Smart Bazaar Building, opp. E City, Phase II, Bengaluru, Karnataka,560100

CIN NO:U62099TN2023PTC164764

Reg : INT/2025/595
Place : Bengaluru
Date : 30 July, 2025

Internship Completion Certificate

This is to certify that **Sandhiya A**, Reg No.**511822104038**, a student of **B.E. Computer Science and Engineering**, **Podhigai College of Engineering and Technology**, has successfully completed an **Offline** internship at our organization in the role of **Python Programming Intern**.

The internship period was from **30.06.2025 to 30.07.2025**, during which the student worked under my supervision and actively participated in various tasks and learning activities. Throughout the internship, the student exhibited a strong commitment to learning, and demonstrated good understanding of **Python Programming** fundamentals and their practical applications.

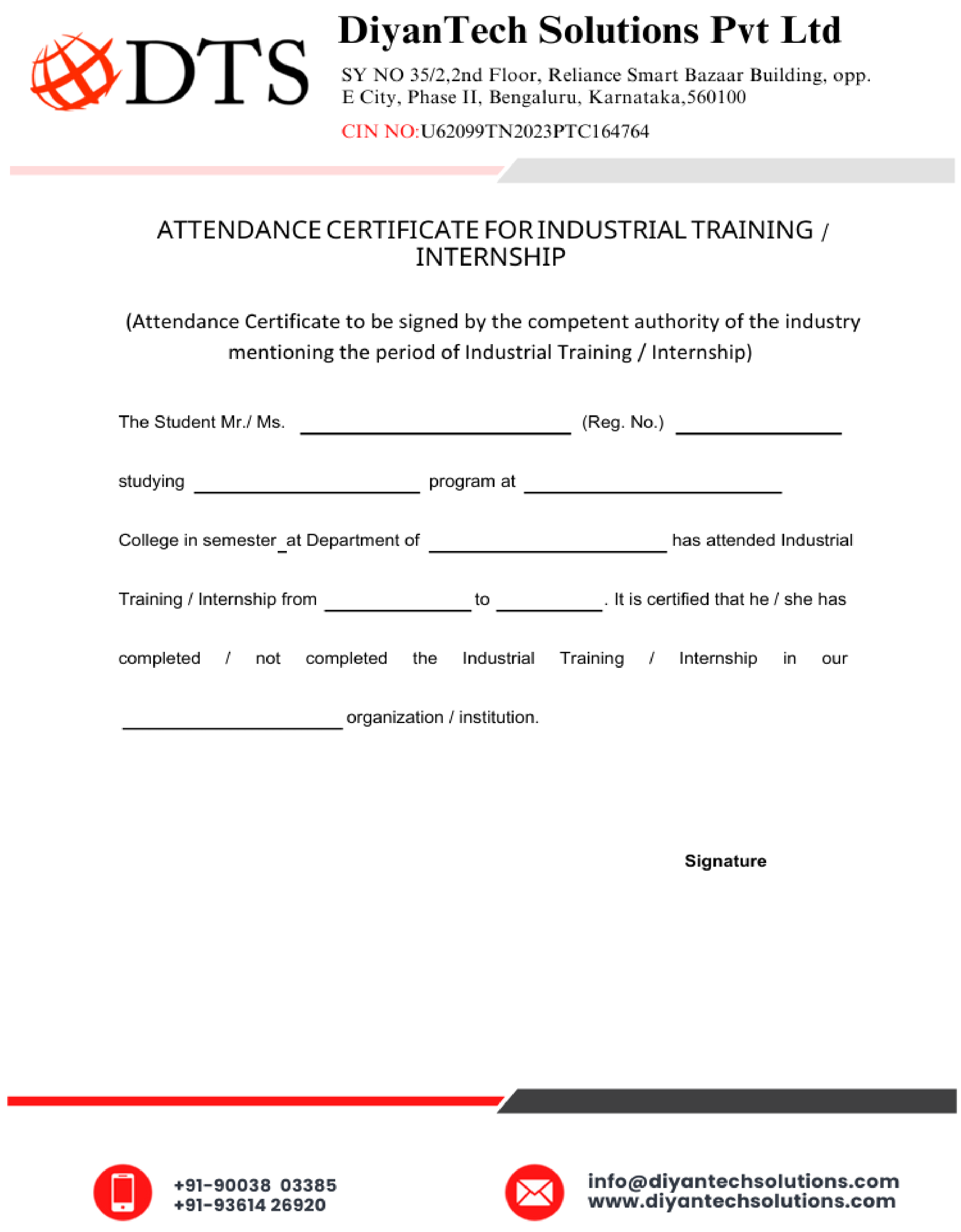
We appreciate the student's contribution and wish them success in all future endeavors.

DIRECTOR OF DTS

+91-90038 03385
+91-93614 26920

info@diyantechsolutions.com
www.diyantechsolutions.com

ATTENDANCE CERTIFICATE FOR INDUSTRIAL TRAINING/ INTERNSHIP



DTS

DiyanTech Solutions Pvt Ltd

SY NO 35/2,2nd Floor, Reliance Smart Bazaar Building, opp.
E City, Phase II, Bengaluru, Karnataka,560100

CIN NO:U62099TN2023PTC164764

ATTENDANCE CERTIFICATE FOR INDUSTRIAL TRAINING / INTERNSHIP

(Attendance Certificate to be signed by the competent authority of the industry mentioning the period of Industrial Training / Internship)

The Student Mr./ Ms. ______________________ (Reg. No.) ____________

studying __________________ program at ____________________

College in semester _at Department of __________________ has attended Industrial

Training / Internship from ____________ to ________. It is certified that he / she has

completed / not completed the Industrial Training / Internship in our

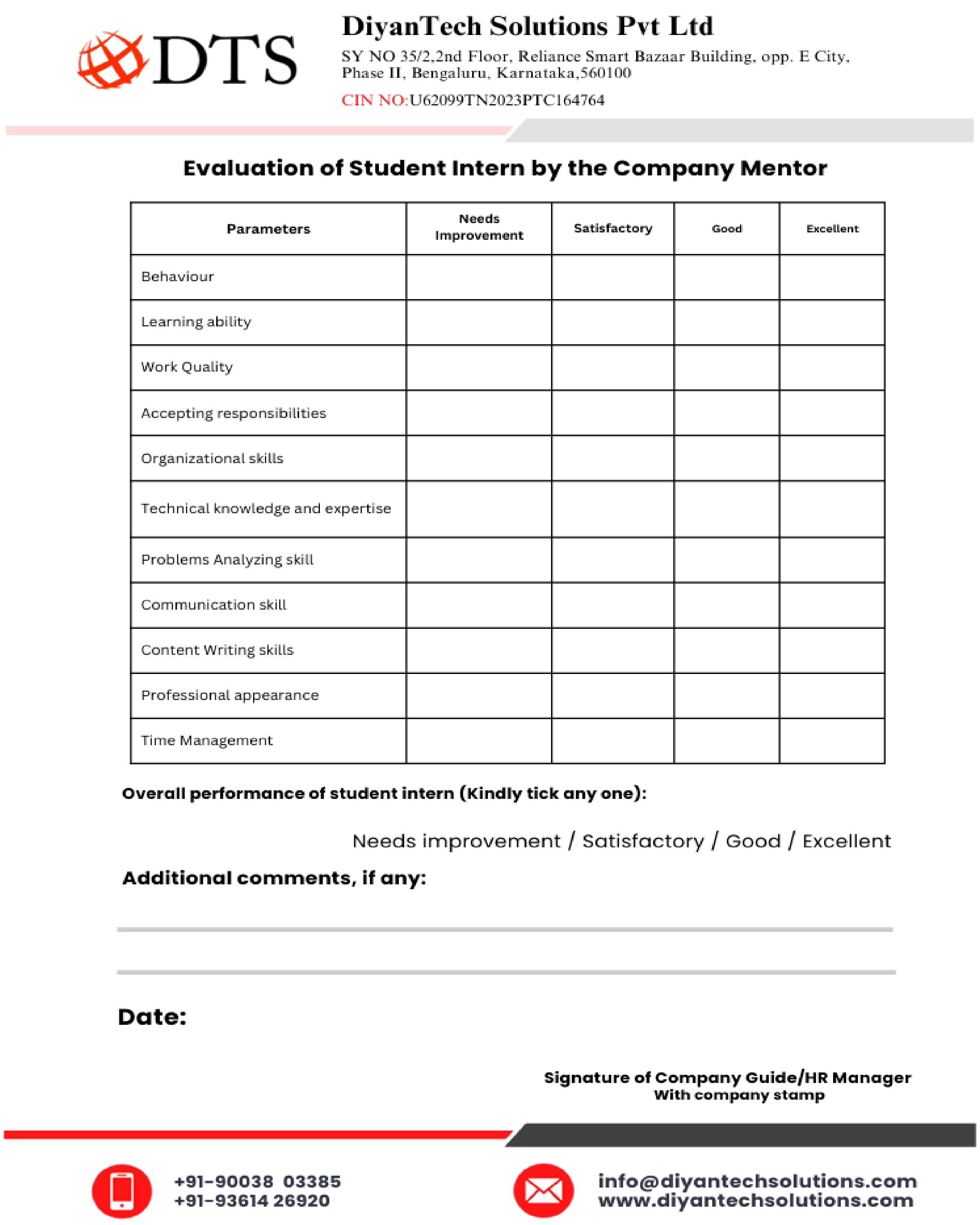
__________________ organization / institution.

Signature

+91-90038 03385
+91-93614 26920

info@diyantechsolutions.com
www.diyantechsolutions.com

EVALUATION OF STUDENT INTERN BY THE COMPANY MENTOR



DTS

DiyanTech Solutions Pvt Ltd

SY NO 35/2,2nd Floor, Reliance Smart Bazaar Building, opp. E City, Phase II, Bengaluru, Karnataka,560100

CIN NO:U62099TN2023PTC164764

Evaluation of Student Intern by the Company Mentor

Parameters	Needs Improvement	Satisfactory	Good	Excellent
Behaviour				
Learning ability				
Work Quality				
Accepting responsibilities				
Organizational skills				
Technical knowledge and expertise				
Problems Analyzing skill				
Communication skill				
Content Writing skills				
Professional appearance				
Time Management				

Overall performance of student intern (Kindly tick any one):

Needs improvement / Satisfactory / Good / Excellent

Additional comments, if any:

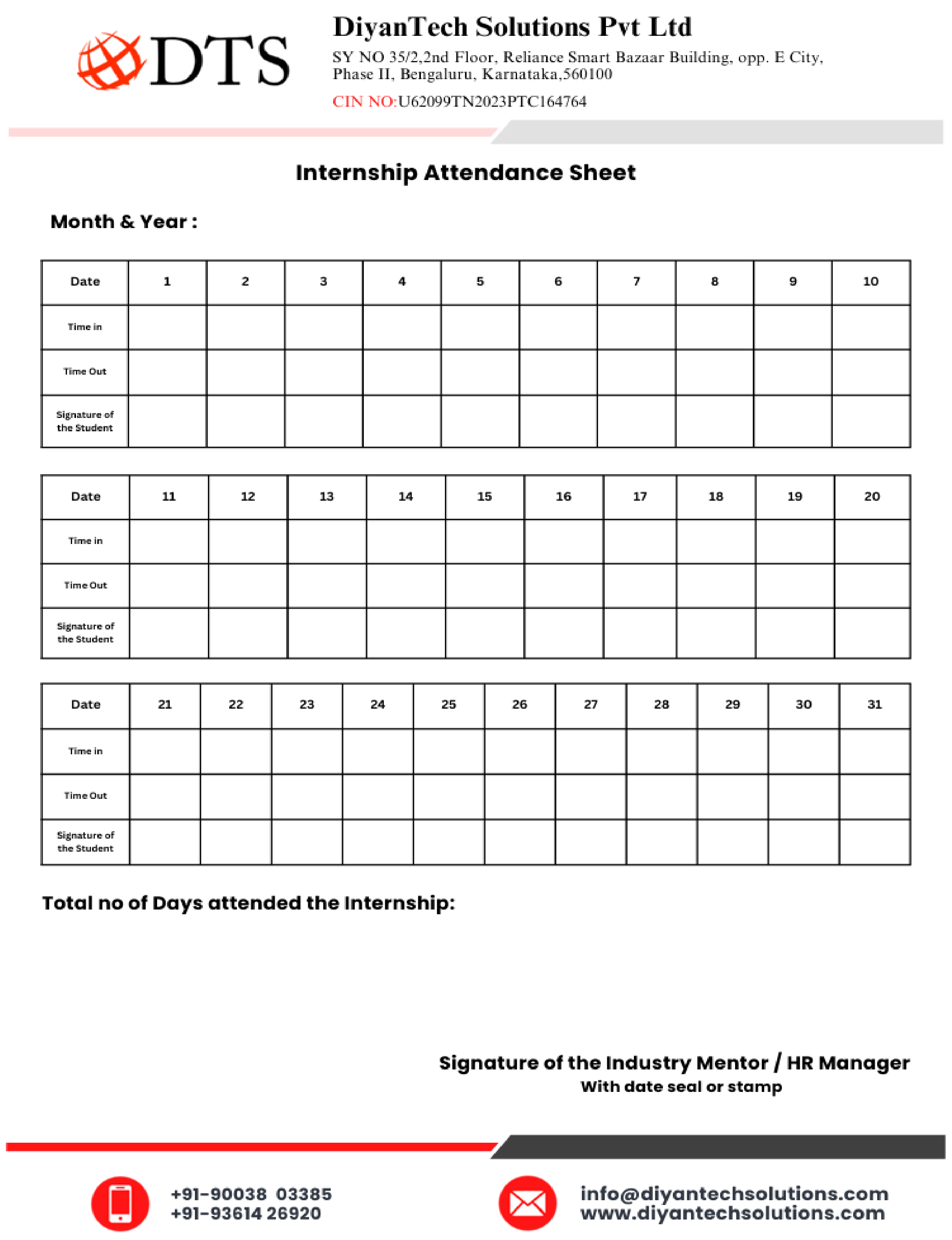
Date:

Signature of Company Guide/HR Manager
With company stamp

+91-90038 03385
+91-93614 26920

info@diyantechsolutions.com
www.diyantechsolutions.com

INTERNSHIP ATTENDANCE SHEET



DTS

DiyanTech Solutions Pvt Ltd
SY NO 35/2,2nd Floor, Reliance Smart Bazaar Building, opp. E City, Phase II, Bengaluru, Karnataka,560100
CIN NO:U62099TN2023PTC164764

Internship Attendance Sheet

Month & Year :

Date	1	2	3	4	5	6	7	8	9	10
Time in										
Time Out										
Signature of the Student										

Date	11	12	13	14	15	16	17	18	19	20
Time in										
Time Out										
Signature of the Student										

Date	21	22	23	24	25	26	27	28	29	30	31
Time in											
Time Out											
Signature of the Student											

Total no of Days attended the Internship:

Signature of the Industry Mentor / HR Manager
With date seal or stamp

+91-90038 03385
+91-93614 26920

info@diyantechsolutions.com
www.diyantechsolutions.com

CONTENT

1. ABSTRACT

The increasing demand for personalized, AI-powered educational tools has led to the development of a web-based Pronunciation Detector application aimed at enhancing users' English language speaking skills. This project leverages Python, Flask, MongoDB, speech recognition libraries, and Text-to-Speech technologies to provide an interactive and engaging language learning experience. The app includes two main learning modes: **Practice Level** (focused on pronunciation and listening) and **Challenge Level** (featuring advanced activities like Speak-the-Word, Audio MCQ, and Fill-in-the-Blank exercises), each categorized across multiple domains and difficulty levels.

Users begin by registering or logging into the app, after which they can explore structured tasks that test both pronunciation and comprehension. The system utilizes browser-based recording (MediaRecorder API) to capture user speech, which is then analyzed using Python's speech processing libraries. MongoDB is used to dynamically serve and retrieve questions, while Flask manages backend operations. The intuitive UI and modular structure support scalability for additional domains and levels.

This report documents the complete software development lifecycle, including system design, implementation, sample code, data storage structure, testing, and screenshots of the application.

2. OBJECTIVE

The primary objective of this project is to design and develop a web-based Pronunciation Detector application that enhances users' spoken English proficiency through interactive practice and challenge activities. This app aims to provide a user-friendly platform that evaluates and improves the user's pronunciation and listening skills using real-time speech recognition and text-to-speech technologies.

The specific objectives include:

1. **To develop a secure login and registration system** for user-based access and personalized usage.
2. **To implement Practice and Challenge levels** categorized by domain (e.g., AI/ML, Web Development, Cybersecurity, etc.) and difficulty (Beginner, Intermediate, Advanced).
3. **To provide structured pronunciation and listening exercises** in the Practice section using Text-to-Speech (TTS) and audio playback.
4. **To evaluate user speech in real time** using speech recognition APIs and the MediaRecorder API.
5. **To store and retrieve questions dynamically** from MongoDB based on domain and level selections.
6. **To offer three challenge types** (Speak-the-Word, Audio MCQ, Fill-in-the-Blank) for advanced evaluation.

7. **To ensure a responsive and intuitive user interface** using HTML, CSS, and Flask-based routing.
8. **To allow users to hear the correct pronunciation, practice repeatedly, and improve gradually.**
9. **To encourage self-assessment** by displaying results and correct answers after quiz completion.
10. **To promote a flexible, scalable language-learning tool** that can be extended with more languages, accents, and voice feedback features.

3. RESOURCE REQUIREMENTS

To successfully develop and deploy the Pronunciation Detector Web Application, the following hardware and software resources are required:

1. Hardware Requirements

Component	**Specification**
Processor	Intel Core i3 or higher
RAM	Minimum 4 GB (Recommended: 8 GB or more)
Hard Disk	Minimum 100 MB free space for project files
Microphone	Built-in or external microphone for voice input
Speaker/Headphones	For listening to audio playback
Internet	Required for initial library installations
Display	13” or larger screen for web UI testing

2. Software Requirements

Component	**Description**
Operating System	Windows 10/11, Linux (Ubuntu), or macOS
Python	Version 3.10 or above
Flask	Web framework for backend development
MongoDB	NoSQL database to store users and questions
HTML/CSS/JavaScript	For frontend design and interactivity
MediaRecorder API	For recording and processing user audio input
pyttsx3	Python library for Text-to-Speech (TTS)
speech_recognition	Python library for recognizing speech input
VS Code or PyCharm	Preferred IDE for code development
MongoDB Compass	GUI for managing MongoDB database
Web Browser	Google Chrome / Firefox for app testing

4.MODEL OVERVIEW

The Pronunciation Detector Web Application is designed to help users improve their pronunciation and listening skills through interactive tasks. The application leverages speech recognition and text-to-speech (TTS) technologies to analyze spoken input, evaluate accuracy, and provide feedback.

System Architecture

The system architecture follows a client-server model with the following core components:

1. User Interface (Frontend)

Developed using **HTML, CSS, and JavaScript**, the UI presents an intuitive and accessible layout. Users can:

- Navigate between different modules (Text-to-Voice, Voice-to-Text, Practice Level, Challenge Level)
- Select domains and levels
- Record and submit voice inputs
- Receive audio or text-based feedback

2. Backend (Flask Framework)

The Flask server handles all:

- User Authentication (Login/Register)
- Data Routing (Serving questions, saving responses)
- Audio Evaluation (Processing submitted recordings)
- Interaction with Database (MongoDB)

3. Speech Recognition Module

Uses Python's speech_recognition library to:

- Convert spoken audio into text
- Compare user input with expected answers
- Evaluate accuracy for Challenge Level task.

4. Text-to-Speech Module

Built with pyttsx3, this module:

- Converts text to audio for listening tasks
- Helps users understand correct pronunciation by hearing it

5. Database (MongoDB)

Stores:

- User credentials and login sessions
- Practice and Challenge Level questions (categorized by domain and level)
- Audio file references and evaluation results (optional)

6. Audio Recording (MediaRecorder API)

The MediaRecorder API in JavaScript is used to:

- Record audio in the browser (client-side)
- Submit user responses in .webm format to the backend for evaluation

5.MODEL WORKFLOW

1. User logs in or registers
2. Selects a mode (Practice / Challenge) and a domain/level
3. Hears a prompt (via TTS) or reads a word
4. Records a response
5. Backend processes and evaluates the response using SR
6. Score/feedback is displayed

WORKFLOW

The workflow of the Pronunciation Detector Web Application describes the step-by-step process that a user experiences while interacting with the system. It outlines how each module works together to deliver an efficient pronunciation learning environment.

1. User Registration and Login

- **Step 1.1**: User opens the web application.
- **Step 1.2**: If the user is new, they register by providing a username and password.
- **Step 1.3**: Existing users log in using their credentials.
- **Step 1.4**: After successful login, user sessions are initiated and managed through Flask.

2. Home Page Navigation

- **Step 2.1**: Once logged in, the user is redirected to the home page.
- **Step 2.2**: The home page provides the following primary options:
  - Text-to-Voice
  - Voice-to-Text
  - Practice Level
  - Challenge Level
  - Additional options: Help, Avatar Bar, Sound Toggle, Suggestion Box

3. Text-to-Voice Mode

- **Step 3.1**: User enters text in an input field.
- **Step 3.2**: The pyttsx3 module converts the text into audio.
- **Step 3.3**: Audio is played back to help the user hear proper pronunciation.

4. Voice-to-Text Mode

- **Step 4.1**: User clicks to start recording.
- **Step 4.2**: The user's speech is recorded using the MediaRecorder API.

- **Step 4.3**: The audio is sent to the Flask backend.
- **Step 4.4**: Python's speech_recognition module converts speech to text and displays it on screen.

5. Practice Level Module

- **Step 5.1**: User selects a domain (e.g., AI, Web Development) and level (Beginner, Intermediate, Advanced).
- **Step 5.2**: A practice question is shown for either:
  - Listening Task: User listens to the audio and selects the correct answer.
  - Pronunciation Task: User sees a word and must pronounce it correctly.
- **Step 5.3**: Audio is played via TTS or user is prompted to speak.
- **Step 5.4**: Pronunciation is optionally compared to expected audio and feedback is provided.

6. Challenge Level Module

- **Step 6.1**: User selects domain and level.
- **Step 6.2**: Three types of tasks are presented randomly or sequentially:
  - Speak-the-Word

    - Audio MCQ
    - Fill-in-the-Blank

- **Step 6.3**: User listens, speaks, or chooses answers as per question type.
- **Step 6.4**: After submission:
    - Score is displayed
    - Answers button appears to review correct responses

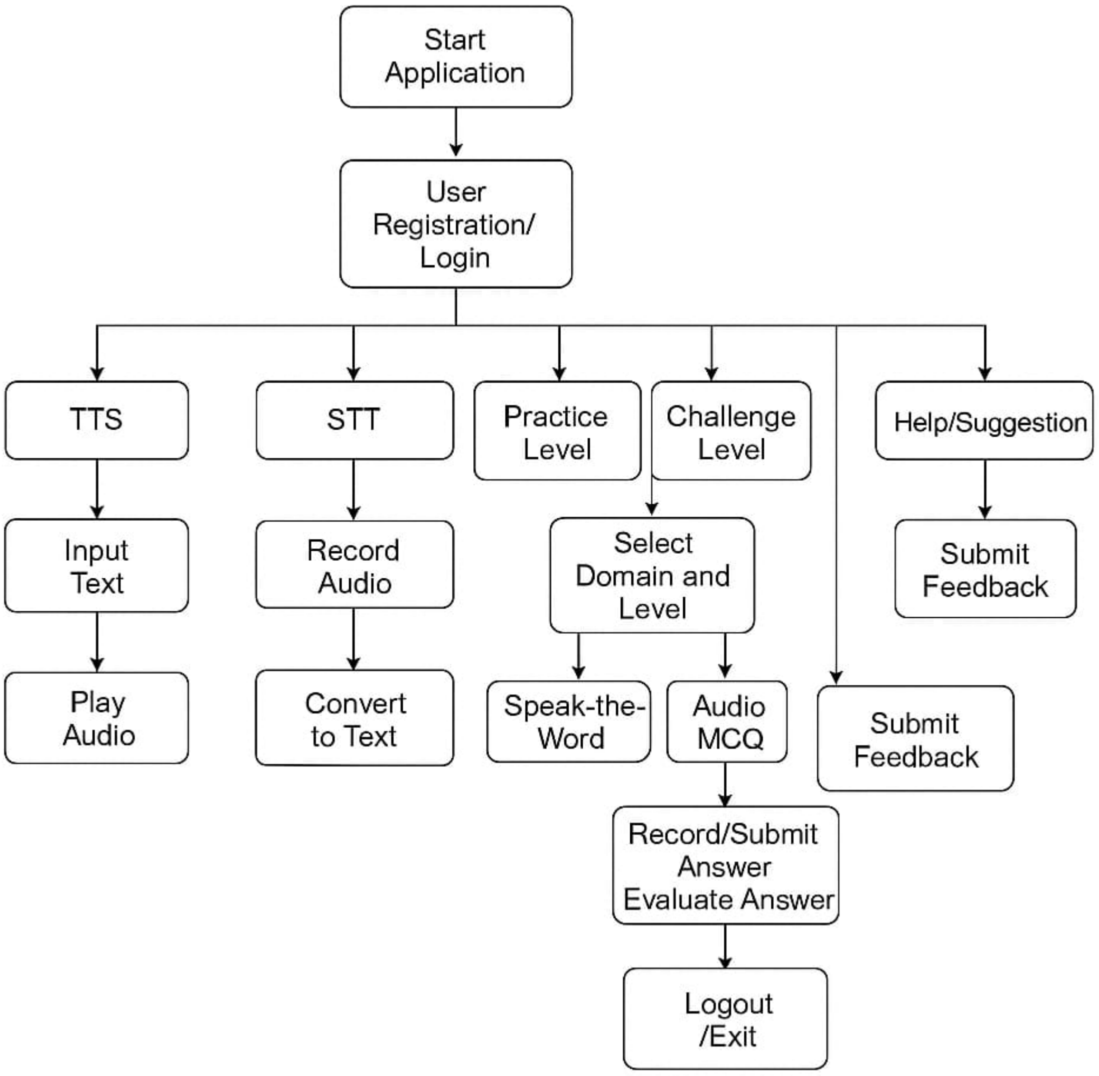
7. Audio Recording and Evaluation

- **Step 7.1**: For speaking tasks, audio is recorded in .webm format.
- **Step 7.2**: Flask server receives the audio and passes it to the recognition engine.
- **Step 7.3**: Recognized text is compared to the expected answer.
- **Step 7.4**: Score is calculated and shown to the user.

8. Suggestion Box and Help

- Users can submit feedback using the Suggestion Box.
- Help page includes tutorials and voice-over guidance using pre-recorded audio.

FLOWCHART FOR WORKFLOW



Overall Workflow

6.CODE IMPLEMENTATION

1.FILE STRUCTURE

```
pronunciation_app/
│
├── app.py
├── usermodel.py
├── templates/
│   ├── base.html
│   ├── login.html
│   ├── register.html
│   ├── home.html
│   ├── practice_level.html
│   ├── challengelevel.html
│   ├── ...
├── static/
│   ├── audio/
│   │   ├── webdev_beg1.webm
│   │   ├── ...
│   └── js/
│       └── recorder.js
├── uploads/
└── requirements.txt
```

2.APP.PY(SAMPLE CODE)

```
from flask import Flask, render_template, request,
redirect, session, url_for

from pymongo import MongoClient

from usermodel import get_user, create_user

import os

app = Flask(_name_)

app.secret_key = 'your_secret_key'

client = MongoClient("mongodb://localhost:27017/")

db = client['pronunciation_app']

practice_collection = db['practice_questions']

@app.route('/')

def index():

    return redirect('/login')

@app.route('/login', methods=['GET', 'POST'])

def login():

    if request.method == 'POST':

        user  =  get_user(request.form['username'],
request.form['password'])

        if user:
            session['username'] = user['username']
            return redirect('/home')
        else:
            return    render_template('login.html',
error="Invalid credentials")
    return render_template('login.html')
@app.route('/register', methods=['GET', 'POST'])
def register():
    if request.method == 'POST':
        result = create_user(request.form)
        if result == "exists":
            return render_template('register.html',
error="User already exists")
        return redirect('/login')
    return render_template('register.html')
@app.route('/home')
def home():
    if 'username' not in session:
        return redirect('/login')
```

```
            return      render_template('home.html',
username=session['username'])
@app.route('/practice-level')
def practice_level():
    if 'username' not in session:
        return redirect('/login')
    domain    =    request.args.get('domain',    'Web
Development')
    level = request.args.get('level', 'Beginner')
    questions                                      =
list(practice_collection.find({'domain':    domain,
'level': level}))
    return   render_template('practice_level.html',
questions=questions, domain=domain, level=level)
if _name_ == '_main_':
    app.run(debug=True)
```

3.USER.PY

```
from pymongo import MongoClient
client = MongoClient("mongodb://localhost:27017/")
```

```
db = client['pronunciation_app']

user_collection = db['users']

def get_user(username, password):

    return    user_collection.find_one({"username":
username, "password": password})

def create_user(data):

    existing                                      =
user_collection.find_one({"username":
data['username']})

    if existing:

        return "exists"

    user = {

        "username": data['username'],

        "password": data['password'],

        "dob": data.get('dob', ''),

        "place": data.get('place', ''),

        "qualification":  data.get('qualification',
''),

        "skills": data.get('skills', '')

    }
    user_collection.insert_one(user)

    return "created"
```

4.CHALLENGE_LEVEL.HTML

```
{% extends "base.html" %}

{% block content %}

<h2>{{ domain }} - {{ level }} Practice</h2>

{% for q in questions %}

<div class="card">

  <h4>{{ loop.index }}. {{ q.word }}</h4>

  <audio controls>

    <source        src="{{        url_for('static',
filename='audio/'      ~      q.audio_file)      }}"
type="audio/webm">

    Your browser does not support the audio element.

  </audio>

</div>

{% endfor %}

{% endblock %}
```

5.MONGODB

```
{
  "domain": "Web Development",
  "level": "Beginner",
  "mode": "Listening",
  "word": "HTML",
  "audio_file": "webdev_beg1.webm"
}
```

6.LOGIN.HTML

```
{% extends "base.html" %}
{% block content %}
<h2>Login</h2>
<form method="POST">
  Username: <input type="text" name="username"><br>
  Password:         <input          type="password"
name="password"><br>
  <button type="submit">Login</button>
</form>
```

```
<a href="/register">Don't have an account? Register here</a>
{% endblock %}
```

7.REGISTER.HTML

```
{% extends "base.html" %}
{% block content %}
<h2>Register</h2>
<form method="POST">
  Username: <input type="text" name="username"><br>
  Password: <input type="password" name="password"><br>
  <button type="submit">Register</button>
</form>
{% endblock %}
```

8.HOME.HTML

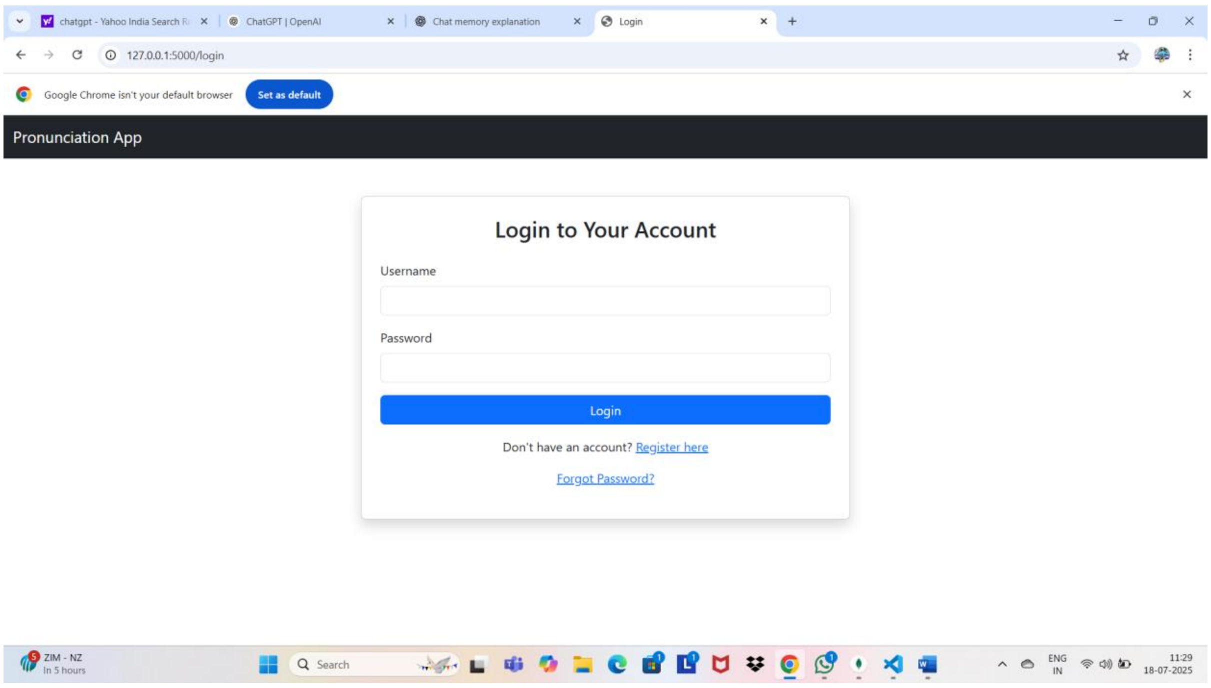
```
{% extends "base.html" %}
{% block content %}
<h2>Welcome {{ username }}</h2>
```

```
<a href="/practice-level?domain=AI&level=Beginner">Practice</a><br>
<a href="/challenge-level?domain=AI&level=Beginner">Challenge</a>
{% endblock %}
```

9.PRACTICE_LEVEL.HTML

```
{% extends "base.html" %}
{% block content %}
<h2>Practice - {{ domain }} ({{ level }})</h2>
{% for question in questions %}
  <div>
    <p>{{ question.text }}</p>
    <audio controls>
      <source src="{{ url_for('static', filename='audio/' ~ question.audio_file) }}" type="audio/webm">
    </audio>
  </div>
{% endfor %}
{% endblock %}
```

7.SAMPLE OUTPUT(SCREENSHOT)



Pronunciation App

Login to Your Account

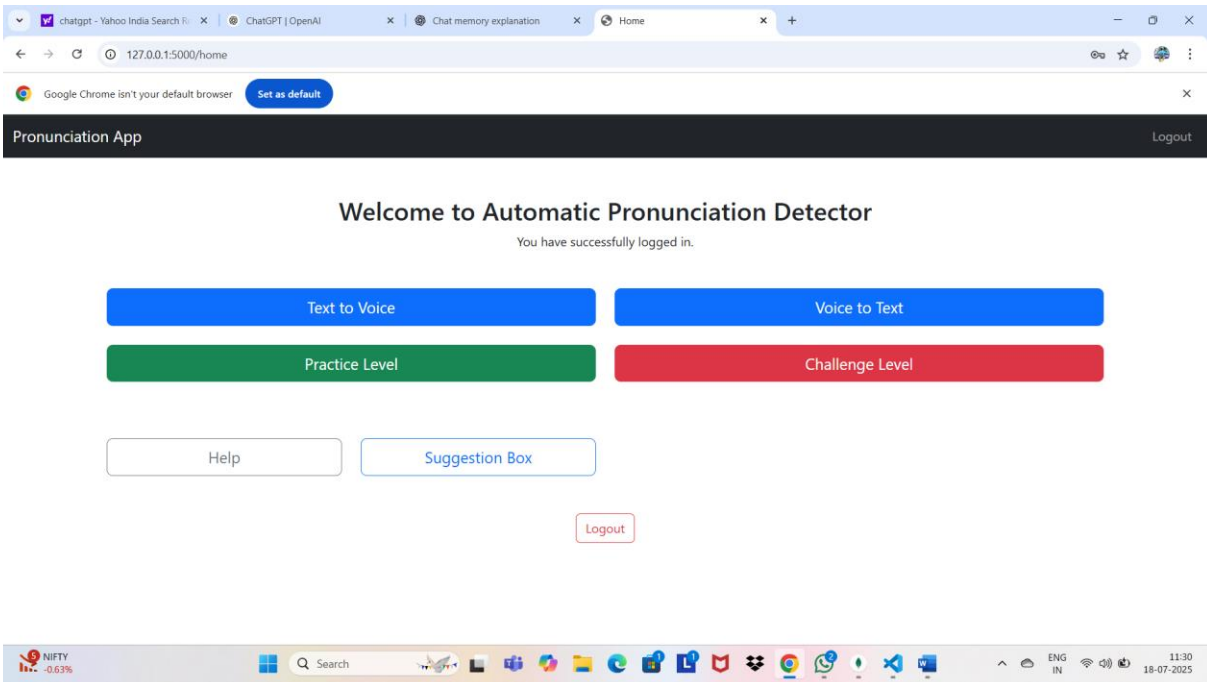
Username

Password

Login

Don't have an account? Register here

Forgot Password?



Pronunciation App

Logout

Welcome to Automatic Pronunciation Detector

You have successfully logged in.

Text to Voice

Voice to Text

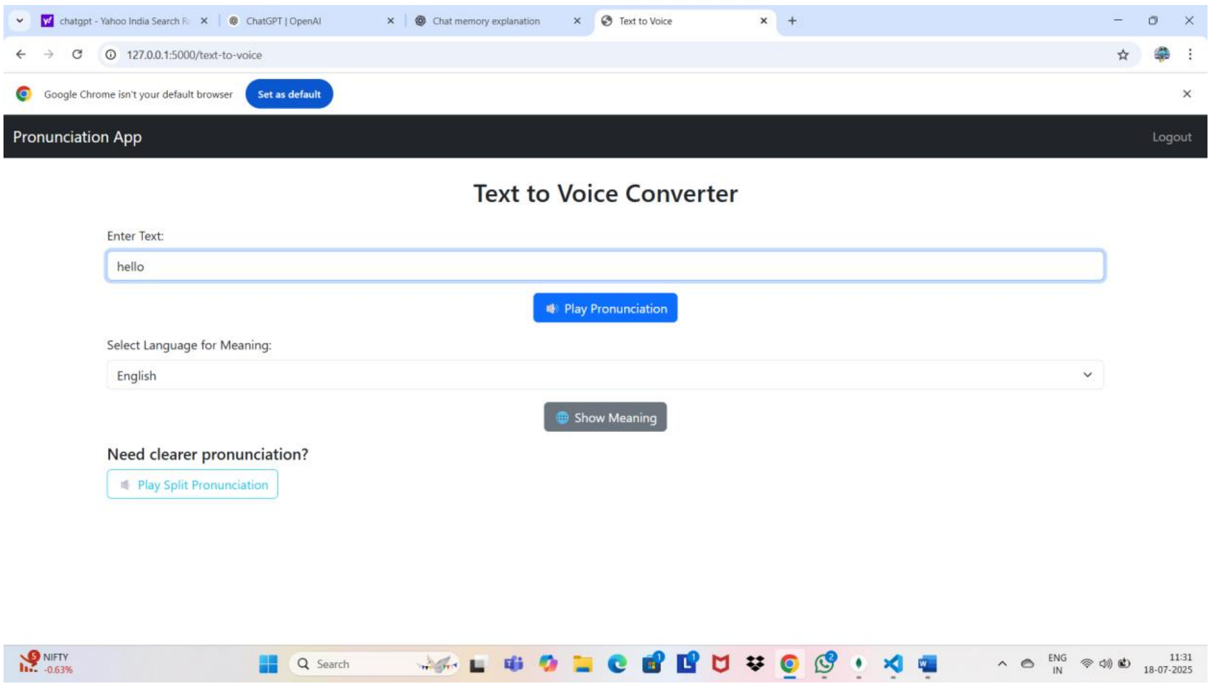
Practice Level

Challenge Level

Help

Suggestion Box

Logout


chatgpt - Yahoo India Search R
ChatGPT | OpenAI
Chat memory explanation
Text to Voice
127.0.0.1:5000/text-to-voice
Google Chrome isn't your default browser
Set as default

Pronunciation App

Logout

Text to Voice Converter

Enter Text:

hello

Play Pronunciation

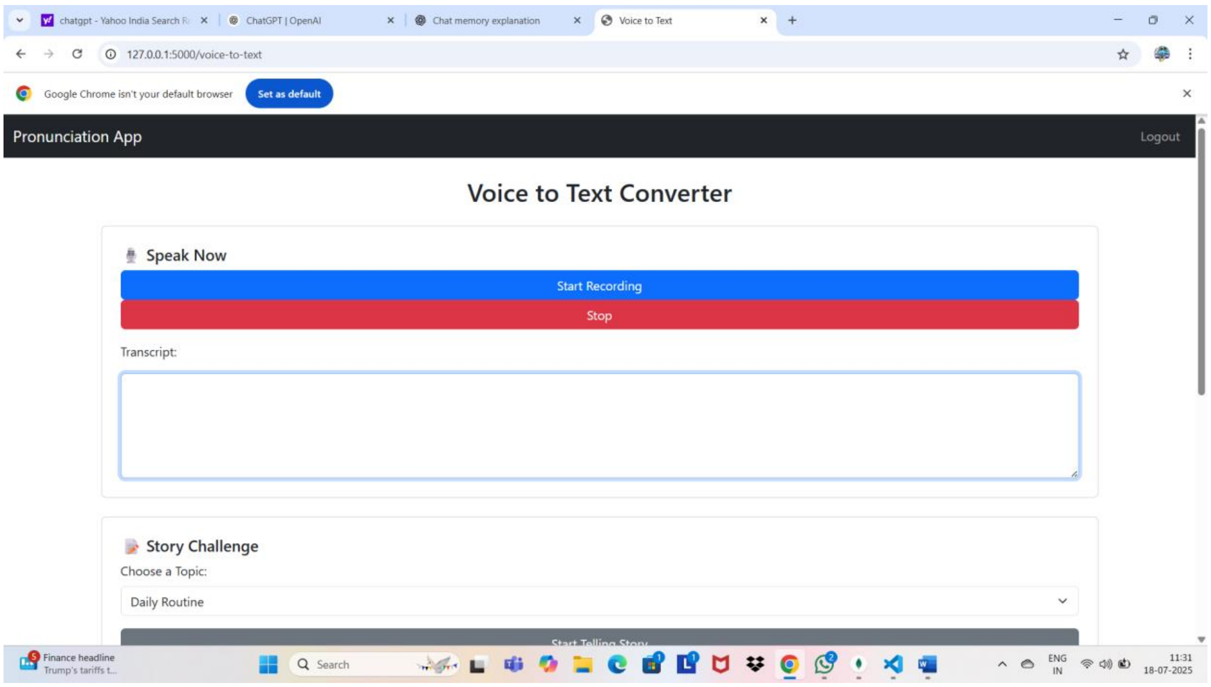
Select Language for Meaning:

English

Show Meaning

Need clearer pronunciation?

Play Split Pronunciation


chatgpt - Yahoo India Search R
ChatGPT | OpenAI
Chat memory explanation
Voice to Text
127.0.0.1:5000/voice-to-text
Google Chrome isn't your default browser
Set as default

Pronunciation App

Logout

Voice to Text Converter

Speak Now

Start Recording

Stop

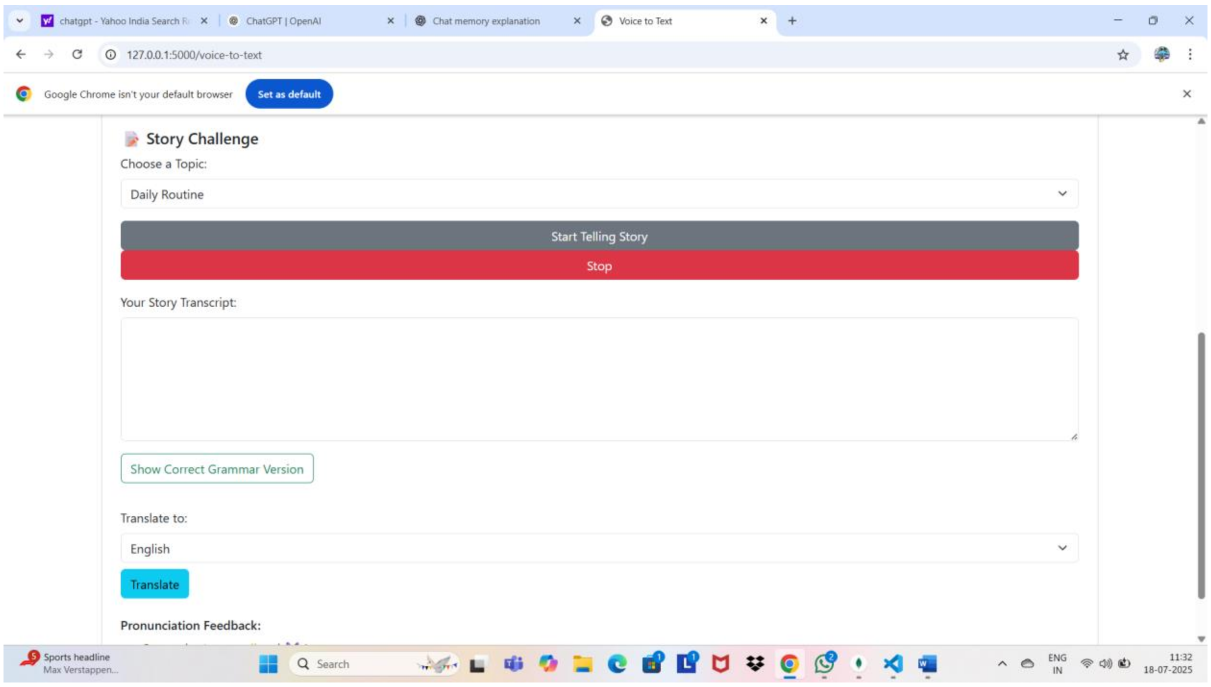
Transcript:

Story Challenge

Choose a Topic:

Daily Routine

Start Telling Story


chatgpt - Yahoo India Search Re
ChatGPT | OpenAI
Chat memory explanation
Voice to Text
127.0.0.1:5000/voice-to-text
Google Chrome isn't your default browser
Set as default

Story Challenge

Choose a Topic:

Daily Routine

Start Telling Story

Stop

Your Story Transcript:

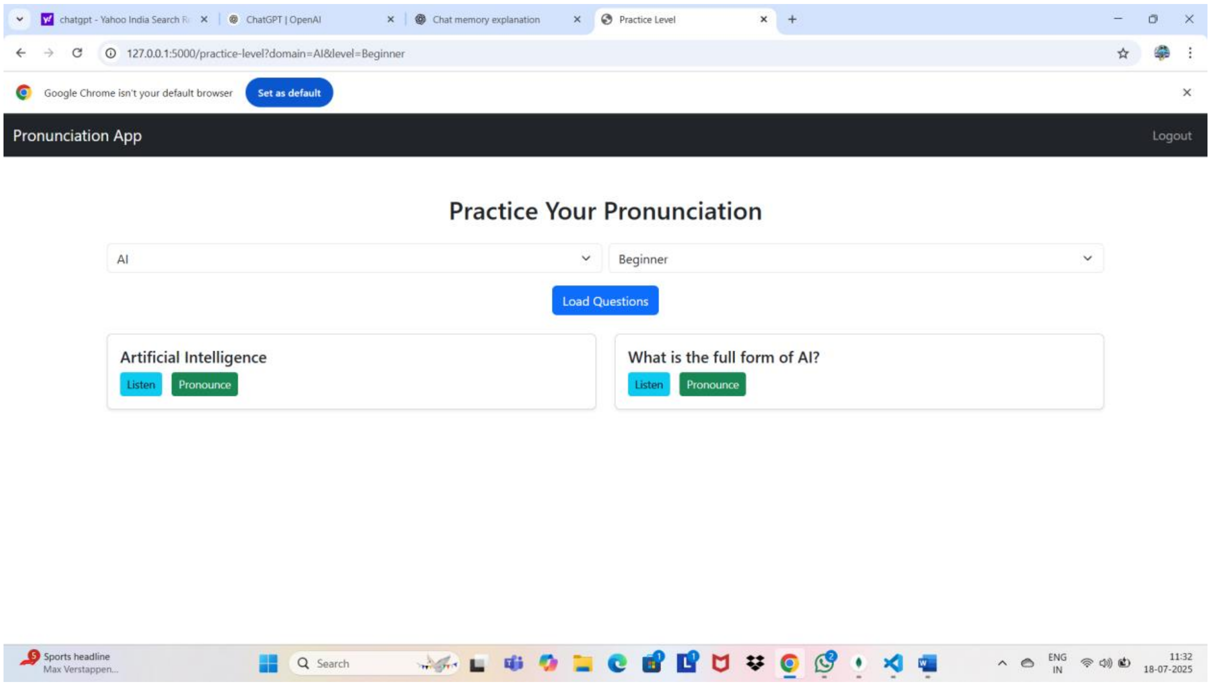
Show Correct Grammar Version

Translate to:

English

Translate

Pronunciation Feedback:


chatgpt - Yahoo India Search Re
ChatGPT | OpenAI
Chat memory explanation
Practice Level
127.0.0.1:5000/practice-level?domain=AI&level=Beginner
Google Chrome isn't your default browser
Set as default

Pronunciation App

Logout

Practice Your Pronunciation

AI

Beginner

Load Questions

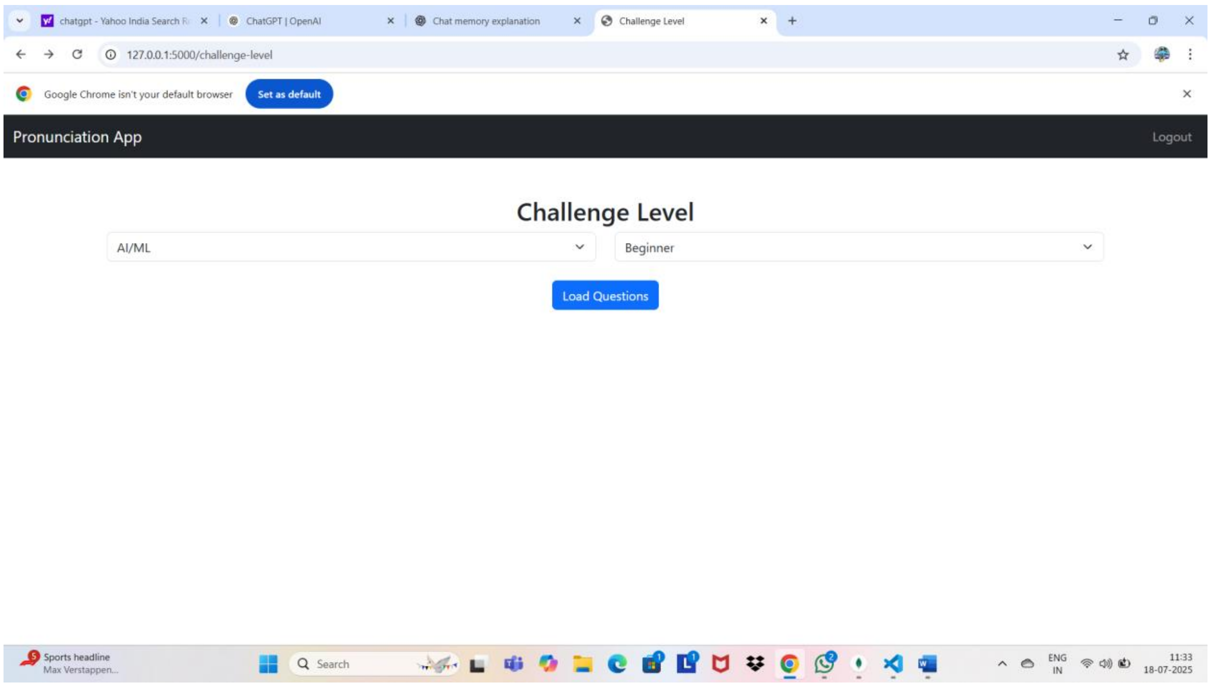
Artificial Intelligence

Listen Pronounce

What is the full form of AI?

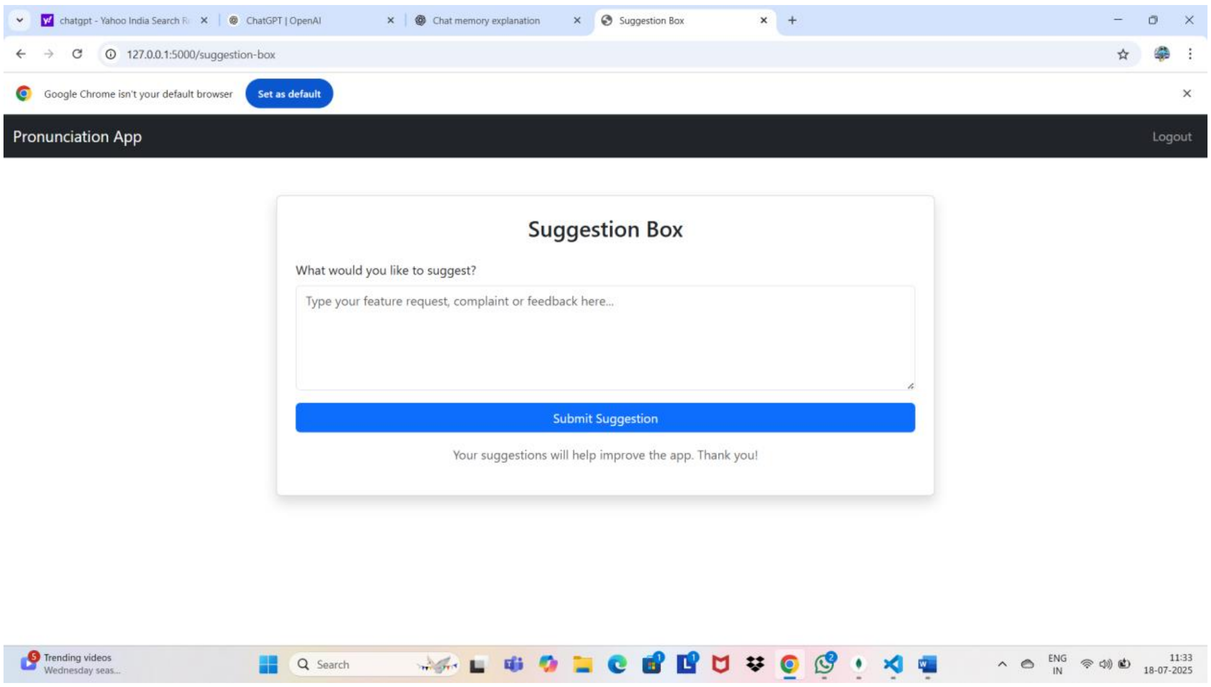
Listen Pronounce

Sports headline
Max Verstappen...
Search
ENG
IN
11:32
18-07-2025


127.0.0.1:5000/challenge-level

Pronunciation App

Challenge Level


127.0.0.1:5000/suggestion-box

Pronunciation App

Suggestion Box

What would you like to suggest?

Type your feature request, complaint or feedback here...

Your suggestions will help improve the app. Thank you!

8.AUDIO HANDLING

In the Pronunciation Detector App, audio handling is a core component that bridges user interaction with speech processing. The app allows users to record their pronunciation through a browser-based interface using the MediaRecorder API and sends the recorded .webm audio to the server for analysis.

For playback and practice questions, the app preloads audio files corresponding to the pronunciation and listening tasks. These audio files are stored in the /static/audio/ directory of the Flask project and mapped to MongoDB question entries. Each file is named and referenced in the HTML dynamically based on the selected domain and level.

The recorded files from the user are saved in the uploads/ directory with a timestamp-based filename. These recordings are then evaluated using Python libraries for further comparison.

9.EVALUATION LOGIC (PYTHON SNIPPET)

To assess the user's pronunciation, the app uses the speech_recognition library to convert the spoken audio into text. Then it compares the recognized text with the expected correct word or sentence using fuzzy matching or string similarity techniques.

```
import speech_recognition as sr
from difflib import SequenceMatcher

def evaluate_pronunciation(audio_file,
expected_text):
    recognizer = sr.Recognizer()
    with sr.AudioFile(audio_file) as source:
        audio_data = recognizer.record(source)
        try:
            user_text =
recognizer.recognize_google(audio_data)
            similarity = SequenceMatcher(None,
user_text.lower(), expected_text.lower()).ratio()
            score = round(similarity * 100, 2)
            return {"user_text": user_text, "similarity_score": score}
        except sr.UnknownValueError:
            return {"error": "Could not understand the audio"}
        except sr.RequestError:
            return {"error": "Service is unavailable"}
```

This function returns a similarity score which is displayed on the evaluation page after submission.

10.CHALLENGES FACED

Developing this pronunciation app posed several challenges:

1.**Browser Compatibility**: Ensuring the MediaRecorder API worked consistently across Chrome, Firefox, and Edge.

2.**Audio Quality Variability**: Background noise and microphone differences impacted speech recognition accuracy.

3.**Real-Time Audio Evaluation**: Handling large audio files and processing them without latency required optimization.

4.**Flask and MongoDB Integration:** Dynamically loading questions and storing user sessions required careful structuring of the backend.

5.**Frontend UI Complexity**: Building an intuitive and clean interface to support multiple levels and modes without overwhelming the user.

11.FUTURE ENHANCEMENT

Several enhancements are planned for future versions of the app:

1.**AI-based Accent Analysis:** Use deep learning models to assess accents and give tailored feedback.

2.**Gamification:** Add badges, scores, and leaderboards to motivate users.

3.**User Progress Tracking**: Store user evaluations and provide visual graphs over time.

4.**Mobile Compatibility:** Optimize the UI and audio input for mobile devices.

5.**Multilingual Support:** Extend beyond English to include regional Indian languages for broader accessibility.

12.CONCLUSION

The Pronunciation Detector App serves as a foundational tool for language learners, enabling them to practice and receive feedback on their spoken English. With interactive modules like Listening Mode, Pronunciation Practice, and Challenge Level, the app supports both beginners and advanced learners. The integration of Flask, MongoDB, speech recognition, and audio handling in the browser makes this project a robust and scalable web-based solution. It fulfills the primary goal of combining education and technology for effective self-assessment in pronunciation.

13.REFERENCE

1. Python SpeechRecognition Library – https://pypi.org/project/SpeechRecognition/

2. Flask Web Framework – https://flask.palletsprojects.com/

3. MongoDB Documentation – https://www.mongodb.com/docs/

4. Web MediaRecorder API –/https://developer.mozilla.org/en-US/docs/Web/API/MediaRecorder

5. Pyttsx3 Text-to-Speech Engine – https://pypi.org/project/pyttsx3/

6. HTML5 Audio Tag Guide – https://developer.mozilla.org/en-US/docs/Web/HTML/Element/audio