

### **1.Salesforce Project :**

- **D-Mart(Grocery Management System)**

### **2.AI with Python Projects:**

- **Speech to Speech Translation in Real Time**
  - **Text to Speech**
  - **Text Summarization**
  - **Facial Emotion Detection**

**मास्टर ऑफ कंप्यूटर अनुप्रयोग पाठ्यक्रम की आंशिक परिपूर्ति हेतु  
परियोजना कार्य  
सत्र :-2023-2025**

**प्रस्तुतकर्ता  
दीपांशु किशोर आजाद**

M.C.A. चतुर्थ छमाही  
नामांकन संख्या :- 2023/10/811/014

**परियोजना निदेशक  
डॉ. हर्षलता पेटकर**

(सहायक प्रोफेसर)  
भाषा प्रौद्योगिकी एवं भाषा अभियांत्रिकी विभाग  
म.गां.अं.हिं.वि.विधा



**भाषा प्रौद्योगिकी एवं भाषा अभियांत्रिकी विभाग  
भाषा विद्यापीठ**

**महात्मा गांधी अंतरराष्ट्रीय हिंदी विश्वविद्यालय**  
(संसद द्वारा पारित अधिनियम 1997, क्रमांक 3 के अन्तर्गत स्थापित केन्द्रीय विश्वविद्यालय)  
पोस्ट : हिंदी विश्वविद्यालय, गांधी हिल, वर्धा – 442001 (महाराष्ट्र) भारत)



**KAVIN**  
Information Technology Services

**KAVIN INDIA PVT LTD**  
[www.kavinindia.com](http://www.kavinindia.com)

Date: 4<sup>th</sup> March, 2025

Candidate name: Dipanshu Kishor Azad

College name: Mahatma Gandhi Antarrashtriya Hindi VishwaVidyalaya, Wardha.

Department: Computer Science, 4<sup>th</sup> Sem.

City: Nagpur

Subject: Internship Selection Confirmation

Dear Dipanshu,

We are pleased to confirm your acceptance of an internship with Kavin India Pvt Ltd.

Below is the details for the same:

- SalesForce CRM

Your first day of internship session will be 5th March, 2025. Internship duration will be of 45 days from March 2025 to May 2025.

Congratulations and welcome to the organization. Further communication would be sent to your registered mail id.

Project Head  
**Kavin India Pvt Ltd**  
(Authorized Signatory)



**CORPORATE OFFICE**

1414, Hunters Glen Dr, Plainsboro New Jersey - 08536, USA  
Tel: +1-443-878-2147, Fax: +1-443-548-0904

**GLOBAL DEVELOPMENT CENTER**

#21, Bonaire, Denise Avenue, Morningside, Sandton, Johannesburg - 2169 SA,  
Tel: +27-730-606939

**HEAD OFFICE**

#P-5/A, 102/201, Akanksha, RPTS Road, Laxmi Nagar, Nagpur - 440022, INDIA  
Tel: +91-712-6464622, Fax: +91-712- 2248422

**BRANCH OFFICE**

Ground Floor, Anand Palace - 11, Opp. Yavant Dada Polytechnic,  
Nandanvan, Nagpur - 09, Tel : 712 - 2712012, Cell : 92256 70888



**IT NetworkZ**  
LETS GET INTERNETWORKED

web : [www.it-networkz.com](http://www.it-networkz.com)  
e-mail : [contact@it-networkz.com](mailto:contact@it-networkz.com)

Date: 24<sup>th</sup> February, 2025

Candidate name: Dipanshu Azad

College name: Mahatma Gandhi Antarrashtriya Hindi Vishwavidyalaya, Wardha

Department: Computer Science, 4<sup>th</sup> Sem.

City: Nagpur

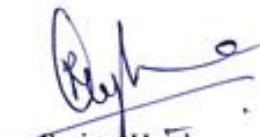
**Subject: Confirmation Letter for IT Internship / Industrial Training**

Dear Dipanshu,

Thank you for showing interest in an IT internship with IT-NetworkZ Infosystems Pvt. Ltd. We are glad to confirm your name for the same. This Internship will proceed in accordance with the approved modules and specifications based on "AI using Python", under the organization's guidance and supervision. Your Internship duration will be of approx. 45 Days, from February - April, 2025.

Congratulations and welcome to the team!

Further communication would be sent to your registered email id.

  
Project Head  
**IT-NetworkZ**  
(Authorized Signatory)



  
Technical Co-ordinator  
**IT-NetworkZ**  
(Authorized Signatory)

**CORPORATE OFFICE**  
1414, Hunters Glen Dr, Plainsboro New Jersey - 08536, USA  
Tel: +1-443-878-2147, Fax: +1-443-548-0904

**HEAD OFFICE**  
#P-5/A,102/201, Akanksha, RPTS Road, Laxmi Nagar, Nagpur - 440022, INDIA  
Tel : 9226063901 / 8799951565

**GLOBAL DEVELOPMENT CENTER**  
#21, Bonaire, Denise Avenue, Morningside, Sandton, Johannesburg - 2169 SA, Tel: +27-730-606939

\* USA \* SA \* INDIA

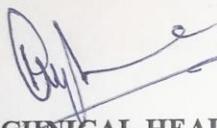
## **ACKNOWLEDGEMENT**

I have taken efforts in the project. However, it would not have been possible without the kind and help of support and help of many individual and organizations would like to extend my sincere thanks to all of them.

I am highly indebted to Mr. Rajiv Verma for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project. I would like to express my gratitude toward my parents & members of IT-NetworkZ Infosystems Pvt. Ltd. for their kind co-operation and encouragement which help me in completion of this project.

I would like to give my special gratitude and thanks to industry persons for giving me such attention and time.

My thanks and appreciation also goes to my colleague in developing the project and people who have willingly helped me out with their abilities.



**TECHNICAL HEAD**

Rajiv Varma



**SUBMITTED TO**

DIPANSHU KISHOR AZAD

# CERTIFICATE OF ACHIEVEMENT

This is to certify that

## Mr Dipanshu Kishor Azad



Has successfully fulfilled the technical requirements and actively contributed to an Industrial Internship in AI Using Python and Salesforce CRM at IT-NetworkZ Infosystems Pvt Ltd under the expert supervision of our technical team.

The Internship tenure was From February 2025 to May 2025.

We acknowledge his sincere efforts and wish his continued success in his career ahead.

Rajiv Verma  
Technical Head

IT-NetworkZ Infosystems Pvt Ltd

Nagpur

Elavarasan Varadharajan  
Director

Kavin India Pvt Ltd



KAVIN INDIA  
PVT. LTD.



**महात्मा गांधी अंतर्राष्ट्रीय हिंदी विश्वविद्यालय**  
Mahatma Gandhi Antarrashtriya Hindi Vishwavidyalaya  
(संसद द्वारा पारित अधिनियम 1997, क्रमांक 3 के अंतर्गत स्थापित केंद्रीय विश्वविद्यालय)  
(A central University Established by Parliament by Act. No. 3 of 1997)

दिनांक...../...../2025

### घोषणा-पत्र

मैं यह घोषणा करता हूँ कि **IT-NetworkZ Infosystem Pvt. Ltd.** में मेरी इंटर्नशिप के दौरान किए गए प्रोजेक्ट का शीर्षक "**D-Mart Grocery Management System**", "**Facial Emotion Detection**", "**Speech to Speech Translation in Real Time**", "**Text Summarization**", "**Text to Speech**" ये रिपोर्ट मेरे द्वारा किए गए मौलिक कार्यों का सच्चा विवरण प्रस्तुत करती है, और यह किसी अन्य डिग्री या डिप्लोमा के लिए पहले कभी प्रस्तुत नहीं की गई है।

यह रिपोर्ट मेरे द्वारा की गई मेहनत और प्राप्त अनुभवों का सही चित्रण करती है।

प्रस्तुतकर्ता  
दीपांशु किशोर आजाद  
नामांकन संख्या: 2023/10/811/014



**महात्मा गांधी अंतर्राष्ट्रीय हिंदी विश्वविद्यालय**  
Mahatma Gandhi Antarrashtriya Hindi Vishwavidyalaya  
(संसद द्वारा पारित अधिनियम 1997, क्रमांक 3 के अंतर्गत स्थापित केंद्रीय विश्वविद्यालय)  
(A central University Established by Parliament by Act. No. 3 of 1997)

डॉ. हर्षलता पेटकर

**Dr. Harshalata Petkar**

सहायक प्रोफेसर, भाषा प्रौद्योगिकी एवं भाषा अभियांत्रिकी विभाग

*Assistant Professor, Language Technology and Language Engineering Department*

म.गा.अं.हिं.वि.वि., वर्धा/ MGAHV, WARDHA

दिनांक...../...../2025

### प्रमाण-पत्र

प्रमाणित किया जाता है कि दीपांशु किशोर आजाद द्वारा एम.सी.ए. पाठ्यक्रम सत्र 2023-25, उपाधि की परियोजना संबंधी आवश्यकता की आंशिक परिपूर्ति हेतु "D-Mart Grocery Management System", "Facial Emotion Detection", "Speech to Speech Translation in Real Time", "Text Summarization", "Text to Speech" विषय पर प्रस्तुत परियोजना कार्य मेरे निदेशन में पूर्ण किया गया है। मेरे संज्ञान में इसे अंशतः या पूर्णतः इस विश्वविद्यालय या किसी अन्य संस्था में किसी उपाधि हेतु प्रस्तुत नहीं किया गया है।

प्रस्तुत परियोजना कार्य को प्रमाणित करते हुए मूल्यांकन हेतु अग्रेषित किया जाता है।

(डॉ. हर्षलता पेटकर)

परियोजना निदेशक

## प्रमाण पत्र

vii



## महात्मा गांधी अंतरराष्ट्रीय हिंदी विश्वविद्यालय

Mahatma Gandhi Antarrashtriya Hindi Vishwavidyalaya

(संसद द्वारा पारित अधिनियम 1997, क्रमांक 3 के अंतर्गत स्थापित केंद्रीय विश्वविद्यालय)

(A central University Established by Parliament by Act. No. 3 of 1997)

प्रो. आनन्द पाटील

**Prof. Anand Patil**

विभागाध्यक्ष, भाषा प्रौद्योगिकी एवं भाषा अभियांत्रिकी विभाग

Head, *Language Technology and Language Engineering Department*

म.गां.अं.हिं.वि.वि., वर्धा/ MGAHV, WARDHA

दिनांक...../...../2025

### प्रमाण-पत्र

प्रमाणित किया जाता है कि दीपांशु किशोर आजाद द्वारा एम.सी.ए. पाठ्यक्रम सत्र 2023-25. उपाधि की परियोजना संबंधी आवश्यकता की आंशिक परिपूर्ति हेतु "D-Mart Grocery Management System", "Facial Emotion Detection", "Speech to Speech Translation in Real Time", "Text Summarization", "Text to Speech" विषय पर प्रस्तुत परियोजना कार्य डॉ. हर्षलता पेटकर के निदेशन में पूर्ण किया गया है। मेरे संज्ञान में इसे अंशतः या पूर्णतः इस विश्वविद्यालय या किसी अन्य संस्था में किसी उपाधि हेतु प्रस्तुत नहीं किया गया है।  
प्रस्तुत परियोजना कार्य को प्रमाणित करते हुए मूल्यांकन हेतु अग्रेषित किया जाता है।

(प्रो. आनन्द पाटील)



**महात्मा गांधी अंतर्राष्ट्रीय हिंदी विश्वविद्यालय**  
Mahatma Gandhi Antarrashtriya Hindi Vishwavidyalaya  
(संसद द्वारा पारित अधिनियम 1997, क्रमांक 3 के अंतर्गत स्थापित केंद्रीय विश्वविद्यालय)  
(A central University Established by Parliament by Act. No. 3 of 1997)

दिनांक: ...../...../202

## आभार

मैंने इस परियोजना में बहुत प्रयास किए हैं हालांकि, कई व्यक्तियों के सहयोग और मदद के बिना यह संभव नहीं होता। मैं सभी को अपना हार्दिक धन्यवाद देना चाहता हूँ।

इसके अलावा, अपने शैक्षणिक निदेशक डॉ० हर्षलता पेटकर, के प्रति आभार व्यक्त करता हूँ, जिन्होंने मेरे इस प्रोजेक्ट के दौरान आवश्यक मार्गदर्शन दिया। मैं डॉ. अंजनी कुमार राय, डॉ. गिरीश पांडेय, श्री सागर ईखे, डॉ. स्मिता तराले का भी आभारी हूँ, जिन्होंने एक अतिरिक्त मार्गदर्शक के रूप में मेरी मदद और समर्थन किया।

मैं विशेष रूप से अपने परियोजना उपदेशक श्री राजीव वर्मा को धन्यवाद करना चाहता हूँ, जिन्होंने मुझे लगातार निर्देश और समर्थन प्रदान किया। उनकी बहुमूल्य सलाह और अनुभव ने मेरी परियोजना को सफलतापूर्वक पूर्ण करने में अहम भूमिका निभाई।

मैं हमारे विभागाध्यक्ष प्रो० आनन्द पाटील का आभार व्यक्त करता हूँ, जिन्होंने मुझे उपयोगी सलाह दी और मेरे काम को बेहतर बनाने में मेरी मदद की।

अंत में, मैं अपने माता-पिता, परिवार, मित्रों और सहकर्मियों का आभार व्यक्त करता हूँ, जिन्होंने मेरी इस यात्रा में मुझे प्रोत्साहन दिया और मेरी सहायता की। यह परियोजना उनके समर्थन के बिना संभव नहीं थी।

प्रस्तुतकर्ता  
दीपांशु किशोर आजाद  
नामांकन संख्या: 2023/10/811/014

## अनुक्रमणिका

क्र.सं.	शीर्षक (Title)	पृष्ठ संख्या
I	Offer Letter Kavin India Pvt. Ltd. (Salesforce)	i
II	Offer Letter IT-NetworkZ Infosystem Pvt. Ltd.(AI With Python)	ii
III	Acknowledgement	iii
IV	Certificate of Achievement	iv
V	घोषणा पत्र	v
VI	प्रमाण पत्र (निदेशक)	vi
VII	प्रमाण पत्र (विभागाध्यक्ष)	Vii
VIII	आभार	viii
1	<b>Salesforce</b>	
1.1	D-Mart (Salesforce CRM)	1
1.1.1	परिचय (Introduction)	2
1.1.2	उद्देश्य (Objective)	4
1.1.3	प्लेटफॉर्म और तकनीक (Platform & Technology Stack)	5
1.1.3.1	App Launcher का उपयोग और एप्लिकेशन विवरण	6
1.1.4	मुख्य ऑब्जेक्ट्स और फ़ील्ड्स (Key Objects & Fields)	7
1.1.5	रिपोर्टिंग और विश्लेषण (Reporting and Analytics)	16

<b>1.1.6</b>	<b>निष्कर्ष (Conclusion)</b>	23
<b>2</b>	<b>AI With Python</b>	
<b>2.1</b>	<b>Speech to Speech Translator in Real Time</b>	
<b>2.1.1</b>	परिचय (Introduction)	25
<b>2.1.2</b>	उद्देश्य एवं विवरण (Objective and Description)	25
<b>2.1.3</b>	प्रयुक्त तकनीकें (Technologies Used)	26
<b>2.1.4</b>	कार्यप्रणाली (Working Process)	26
<b>2.1.5</b>	रिपोर्टिंग और विश्लेषण (Reporting and Analytics)	28
<b>2.1.6</b>	निष्कर्ष (Conclusion)	31
<b>2.2</b>	<b>Text Summarization</b>	
<b>2.2.1</b>	परिचय (Introduction)	33
<b>2.2.2</b>	उद्देश्य (Objective)	33
<b>2.2.3</b>	PEGASUS मॉडल का परिचय	36
<b>2.2.4</b>	Workflow	36
<b>2.2.5</b>	Architecture Flow	39
<b>2.2.6</b>	Algorithm	39
<b>2.2.7</b>	डेटा प्रवाह और निष्कर्ष (Data Flow & Conclusion)	42
<b>2.3</b>	<b>Text to Speech Converter</b>	

<b>2.3.1</b>	परिचय (Introduction)	44
<b>2.3.2</b>	उद्देश्य (Objective)	45
<b>2.3.3</b>	तकनीक (Technology)	45
<b>2.3.3</b>	कार्य प्रणाली (Working process)	46
<b>2.3.4</b>	Cloud Architecture	50
<b>2.3.6</b>	Data Flow Diagram	51
<b>2.3.7</b>	निष्कर्ष (Conclusion)	52
<b>2.4</b>	<b>Facial Emotion Detection</b>	
<b>2.4.1</b>	परिचय (Introduction)	55
<b>2.4.2</b>	उद्देश्य (Objective)	56
<b>2.4.3</b>	Working (कार्य)	57
<b>2.4.4</b>	Output (उत्पादन)	58
<b>2.4.5</b>	तकनीक (Technology)	62
<b>2.4.6</b>	रिपोर्टिंग और विश्लेषण (Reporting and Analytics)	65
<b>2.4.7</b>	निष्कर्ष (Conclusion)	68
<b>VIII</b>	<b>सन्दर्भ (References)</b>	

## आकृति की सूची (List of Figures)

क्र.सं.	आकृति (Figures)	पृष्ठ संख्या
	<b>Salesforce</b>	
	<b>D-Mart (Grocery Management System)</b>	
चित्र 1	समयबद्धक स्थापित करना (Scheduler setup)	7
चित्र 2	वस्तु निर्माण (Object Creation)	10
चित्र 3	फ़ील्ड डेटा निर्माण (Fields Data Creation)	13
चित्र 4	विक्रय ऑर्डर ऑब्जेक्ट लुकअप (Sales Order Object Lookup)	15
चित्र 5	नए आपूर्तिकर्ता में मास्टर सूची (Master List in New Supplier)	18
चित्र 6	ऑर्डर लाइन आइटम ऑब्जेक्ट इंटरलिंक लुकअप (Order Line Item Object Interlink Lookup)	19
	<b>AI With Python</b>	
	<b>Speech to Speech Translator in Real Time</b>	
चित्र 7	क्लाउड आर्किटेक्चर (Cloud Architecture)	22
	<b>Text to Speech Converter</b>	
चित्र 8	Output & File Structure	25
चित्र 9	Data Flow Diagram	29
	<b>Text Summarization</b>	
चित्र 10	Output PEGASUS Text Summarizer	36
चित्र 11	Architecture Flow & File Structure	39

चित्र 12	Data Flow Diagram	42
----------	-------------------	----

### **Facial Expression Recognition**

चित्र 13	Output	51
चित्र 14	Data Flow Diagram & Algorithm	51
चित्र 15	Processing of Pyttsx3 Engine	53
चित्र 16	Output	58
चित्र 17	Architecture Diagram & File Structure	65
चित्र 18	Data Flow Diagram	66

## तालिका सूची (List of Tables)

तालिका संख्या	तालिका का नाम	पृष्ठ संख्या
<b>Salesforce</b>		
<b>D-Mart (Grocery Management System)</b>		
तालिका 1	प्रोडक्ट फील्ड डाटा निर्माण (Product Field Data Creation)	13
तालिका 2	कस्टमर ऑब्जेक्ट फील्ड डाटा निर्माण (Customer Object Field Data Creation)	15
तालिका 3	सेल्स ऑब्जेक्ट फील्ड डाटा निर्माण (Sales Object Field Data Creation)	16
तालिका 4	सेल्स ऑर्डर ऑब्जेक्ट फील्ड डाटा निर्माण (Sales Order Object Field Data Creation)	17
तालिका 5	सप्लायर ऑब्जेक्ट फील्ड डाटा निर्माण (Supplier Object Field Data Creation)	19
तालिका 6	फील्ड डाटा (Field Data)	20
तालिका 7	लुकअप रिलेशन्स (Lookup Relations)	20
तालिका 8	मास्टर लुकअप रिलेशन्स (Master Lookup Relations)	21
<b>AI with Python</b>		
<b>Facial Emotion Detection</b>		
तालिका 9	चेहरे की भावनाओं की पहचान परियोजना सारांश (Facial Emotion Detection Project Summary)	62
तालिका 10	चेहरे की भावनाओं की पहचान परियोजना तकनीकें (Facial Emotion Detection Project Technologies)	64

# 1.Salesforce

# **1.1 D-Mart (Grocery Management Syste**

## D-Mart Grocery Management System on Salesforce

### (सेल्सफोर्स प्लेटफॉर्म पर D-Mart ग्रोसरी प्रबंधन सिस्टम)

#### 1.1.1 परिचय (Introduction)

यह रिपोर्ट एक क्लाउड-बेस्ड Salesforce CRM प्रोजेक्ट "D-Mart: Grocery Management System" की परिचयात्मक जानकारी प्रस्तुत करती है। D-Mart एक आधुनिक डिजिटल समाधान है जिसे किराना व्यवसाय के संपूर्ण प्रबंधन को सरल, सुव्यवस्थित और कुशल बनाने के उद्देश्य से विकसित किया गया है। यह प्रोजेक्ट Salesforce प्लेटफॉर्म पर आधारित है, जो विश्व की अग्रणी कस्टमर रिलेशनशिप मैनेजमेंट (CRM) सेवा है। इसका उपयोग छोटे और बड़े व्यवसायों द्वारा ग्राहकों, उत्पादों, ऑर्डरों, सप्लायर्स और अन्य व्यावसायिक घटकों की प्रभावी ट्रैकिंग और प्रबंधन के लिए किया जाता है।

D-Mart का उद्देश्य पारंपरिक किराना व्यापार में डिजिटल रूपांतरण लाकर व्यवसाय संचालन को ऑटोमेट करना और डेटा-ड्रिवन निर्णय लेने की क्षमता प्रदान करना है। Salesforce के माध्यम से इस प्रोजेक्ट में विभिन्न कस्टम ऑब्जेक्ट्स और फील्ड्स का निर्माण किया गया है जैसे कि Products, Customers, Sales Orders, Suppliers, और Order Line Items। प्रत्येक ऑब्जेक्ट एक विशिष्ट भूमिका निभाता है जो व्यवसाय के एक हिस्से का प्रतिनिधित्व करता है और यह आपस में इंटरलिंक होते हैं ताकि सभी डेटा एक केंद्रीय स्थान पर समेकित रहे।

इस सिस्टम के मुख्य लाभों में से एक यह है कि यह पूरी तरह से क्लाउड पर आधारित है, जिससे उपयोगकर्ता किसी भी स्थान और डिवाइस से एक्सेस कर सकते हैं। यह सुविधा विशेष रूप से मैनेजमेंट और स्टाफ को व्यापार की वास्तविक समय की जानकारी देखने और आवश्यक निर्णय लेने में मदद करती है। उदाहरण के लिए, प्रोडक्ट्स ऑब्जेक्ट में हम उत्पाद की यूनिक आईडी के साथ उत्पाद सूची को देख सकते हैं, जिसे Salesforce की इंटरफ़ेस पर आसानी से सर्च, अपडेट और मैनेज किया जा सकता है। इसी तरह, ग्राहक जानकारी, ऑर्डर डिटेल्स, और सप्लायर डेटा को भी केंद्रीकृत तरीके से संग्रहीत और नियंत्रित किया जा सकता है।

प्रोजेक्ट में ऑटोमेशन फीचर्स का भी प्रभावी ढंग से उपयोग किया गया है। Salesforce के फ्लो, प्रोसेस बिल्डर, और वेलिडेशन रूल्स जैसी सुविधाओं की मदद से डाटा एंट्री में संभावित त्रुटियों को रोका गया है और बिजनेस लॉजिक को कस्टमाइज़ किया गया है। उदाहरण के लिए, किसी भी प्रोडक्ट की एंट्री करते समय यह सुनिश्चित किया जा सकता है कि प्रोडक्ट आईडी एक विशेष फॉर्मेट में हो या कस्टमर का फोन नंबर 10 अंकों का ही हो। यह ऑटोमेटेड नियंत्रण सिस्टम को स्मार्ट और यूज़र फ्रेंडली बनाता है।

D-Mart प्रोजेक्ट में रिपोर्टिंग और डैशबोर्ड की विशेषताएं भी शामिल हैं जो व्यवसाय की परफॉर्मेंस को ट्रैक करने के लिए उपयोग होती हैं। Salesforce में बिल्ट-इन रिपोर्टिंग टूल्स की सहायता से बिक्री की रिपोर्ट, स्टॉक रिपोर्ट, सप्लायर प्रदर्शन आदि का विश्लेषण किया जा सकता है। यह विश्लेषण व्यवसाय मालिक को इनसाइट्स प्रदान करता है जिससे वे बिक्री रणनीति, खरीद प्रक्रिया और कस्टमर सर्विस में सुधार कर सकते हैं।

इस प्रोजेक्ट के विकास के दौरान मुख्य फोकस उपयोगकर्ता के अनुभव और इंटरफ़ेस की सरलता पर रहा है। Salesforce के लाइटनिंग इंटरफ़ेस का उपयोग करते हुए एक क्लीन और सहज डैशबोर्ड तैयार किया गया है, जिसमें यूज़र को सभी जरूरी टैब्स जैसे Home, Products, Customers, Sales Orders, Suppliers और Order Line Items आसानी से नेविगेट करने की सुविधा मिलती है। इससे नया उपयोगकर्ता भी बिना किसी विशेष प्रशिक्षण के सिस्टम को समझ और उपयोग कर सकता है।

D-Mart प्रोजेक्ट का एक और महत्वपूर्ण पक्ष इसका स्केलेबिलिटी और कस्टमाइजेशन है। Salesforce प्लेटफ़ॉर्म पर बने होने के कारण, यदि भविष्य में व्यवसाय का आकार बढ़ता है या नई ज़रूरतें सामने आती हैं, तो इस सिस्टम को आसानी से अपग्रेड और कस्टमाइज़ किया जा सकता है। नई ऑब्जेक्ट्स जोड़ी जा सकती हैं, वर्कफ़्लो अपडेट किए जा सकते हैं, और थर्ड-पार्टी ऐप्स को इंटीग्रेट किया जा सकता है। इससे यह समाधान समय के साथ प्रासंगिक और प्रतिस्पर्धात्मक बना रहता है।

इस प्रोजेक्ट का एक विशेष उद्देश्य यह भी है कि यह डेटा सिक्योरिटी को भी प्राथमिकता देता है। Salesforce की सिक्योरिटी मॉडल के अंतर्गत प्रोफाइल, रोल्स, परमिशन सेट्स और ऑब्जेक्ट-लेवल सिक्योरिटी का प्रयोग किया गया है जिससे डेटा केवल अधिकृत यूज़र्स तक ही सीमित रहता है। इससे ग्राहक की संवेदनशील जानकारी, जैसे कि फोन नंबर और ऑर्डर डिटेल्स, पूरी तरह से संरक्षित रहती हैं।

D-Mart न केवल एक तकनीकी प्रोजेक्ट है, बल्कि यह एक रियल-वर्ल्ड बिजनेस सॉल्यूशन है जो किराना व्यवसायों के लिए डिज़ाइन किया गया है। इस सिस्टम के माध्यम से स्टोर ऑनर अपने सभी उत्पादों को ट्रैक कर सकता है, ग्राहक की मांग और बिक्री को समझ सकता है, और सप्लायर की टाइमलाइन और प्रदर्शन पर नज़र रख सकता है। साथ ही, यह प्रोजेक्ट छात्रों और प्रशिक्षुओं के लिए Salesforce के कॉन्सेप्ट्स को व्यावहारिक रूप में समझने का एक प्रभावशाली उदाहरण भी है।

अंततः, D-Mart Salesforce प्रोजेक्ट एक डिजिटल परिवर्तन की दिशा में एक सकारात्मक कदम है जो पारंपरिक किराना व्यवसायों को आधुनिक, कुशल और प्रतिस्पर्धात्मक बनाता है।

यह व्यवसायों को न केवल वर्तमान में तकनीकी रूप से सशक्त करता है, बल्कि उन्हें भविष्य की

संभावनाओं के लिए भी तैयार करता है। इसका उपयोग करते हुए छोटे व्यवसाय भी बड़े ब्रांड्स की तरह अपने ग्राहकों के साथ स्मार्ट तरीके से जुड़ सकते हैं और उन्हें बेहतर सेवाएं प्रदान कर सकते हैं। Salesforce की शक्तिशाली सुविधाओं और D-Mart के सरल डिज़ाइन के मेल से यह प्रोजेक्ट एक आदर्श CRM समाधान के रूप में उभर कर सामने आता है।

### 1.1.2 उद्देश्य (Objective)

D-Mart Salesforce CRM प्रोजेक्ट का मुख्य उद्देश्य किराना व्यवसाय (Grocery Business) को डिजिटल और ऑटोमेटेड प्लेटफॉर्म पर स्थानांतरित करना है, जिससे व्यावसायिक प्रक्रियाओं का प्रबंधन अधिक सरल, तेज़ और कुशल हो सके। यह परियोजना ग्राहकों, उत्पादों, ॲर्डर्स, सप्लायर्स और उनके बीच के इंटरैक्शन को एक केंद्रीकृत सिस्टम में संगठित करती है ताकि व्यापार मालिक और स्टाफ वास्तविक समय में सभी गतिविधियों पर नज़र रख सकें।

इस परियोजना का लक्ष्य है कि पारंपरिक व्यवसाय मॉडल को एक आधुनिक, क्लाउड-बेस्ड समाधान में बदला जाए जहाँ सभी डेटा सुरक्षित तरीके से संग्रहीत हो और जरूरत के अनुसार कहीं से भी एक्सेस किया जा सके। साथ ही, यह परियोजना उपयोगकर्ता को एसा प्लेटफॉर्म प्रदान करती है जिसमें ॲटोमेशन, रिपोर्टिंग, डैशबोर्ड, और इनसाइट्स के माध्यम से स्मार्ट निर्णय लिए जा सकें।

इसके अतिरिक्त, इस परियोजना का उद्देश्य Salesforce जैसे अत्याधुनिक CRM टूल्स के व्यावहारिक उपयोग को समझना और विभिन्न व्यावसायिक घटकों (जैसे कि प्रोडक्ट्स, कस्टमर्स, सप्लायर्स, ॲर्डर लाइन आइटम्स आदि) को कस्टम ॲब्जेक्ट्स और रिलेशनशिप्स के माध्यम से मॉड्यूलर रूप में विकसित करना भी है। परियोजना विद्यार्थियों, प्रशिक्षुओं और स्टार्टअप्स के लिए एक व्यावसायिक मॉडल का तकनीकी संस्करण प्रस्तुत करती है जो कि आसानी से विस्तृत, अनुकूलनशील और सुरक्षित है।

बिलकुल! नीचे आपके D-Mart Grocery Management System (GMS) प्रोजेक्ट के लिए "प्लेटफॉर्म और तकनीक (Platform & Technology)" अनुभाग को आपके कंटेंट के अनुसार संशोधित किया गया है, जिसमें किताबों की जगह किराना व्यापार और संबंधित कस्टम ॲब्जेक्ट्स को दर्शाया गया है:

### **1.1.3. प्लेटफॉर्म और तकनीक (Platform & Technology)**

D-Mart Grocery Management System को Salesforce प्लेटफॉर्म पर विकसित किया गया है, जो एक विश्वसनीय, स्केलेबल और सुरक्षित क्लाउड-आधारित Customer Relationship Management (CRM) प्रणाली है। इस परियोजना में हमने Salesforce के विभिन्न तकनीकी घटकों और टूल्स का उपयोग करके एक संपूर्ण किराना प्रबंधन समाधान तैयार किया है।

#### **i) प्लेटफॉर्म: Salesforce (Lightning Platform)**

इस प्रोजेक्ट का मुख्य आधार Salesforce का Lightning Platform है, जो हमें बिना किसी कोडिंग या न्यूनतम कोडिंग के साथ एप्लिकेशन डेवलप करने की सुविधा देता है। इसमें App Builder, Object Manager और AppExchange जैसे उपकरण शामिल हैं जो कस्टम समाधान बनाना आसान बनाते हैं।

#### **ii) बैकएंड भाषा: Apex (सही फॉर्मेटिंग के साथ)**

Apex एक object-oriented प्रोग्रामिंग भाषा है जो Salesforce में सर्वर-साइड लॉजिक लिखने के लिए प्रयोग होती है। हमने Apex का उपयोग validation rules, triggers, और batch processes के लिए किया है जैसे - ऑर्डर प्रोसेसिंग, स्टॉक अपडेट, और स्वचालित स्टेट्स परिवर्तन।

#### **iii) UI फ्रेमवर्क: Lightning Web Components (LWC)**

LWC एक आधुनिक, component-based JavaScript framework है जो यूजर इंटरफ़ेस को responsive और interactive बनाता है। इसका उपयोग कस्टम फॉर्म्स, प्रोडक्ट सर्च पैनल, और डैशबोर्ड विज़ुअलाइजेशन में किया गया है।

#### **iv) डेटाबेस: Salesforce Custom Objects**

परियोजना में Product, Customer, Sales Order, Supplier, और Order Line Item जैसे custom objects बनाए गए हैं, जिनमें सभी डेटा संरचित रूप में संग्रहीत होते हैं।

#### **v) ऑटोमेशन: Salesforce Flow, Process Builder**

Flow और Process Builder के माध्यम से हमने auto-status update, stock adjustment, email alerts, और order confirmation जैसे कार्यों को automate किया है।

#### **vi) रिपोर्टिंग: Reports & Dashboards**

Salesforce Reports और Dashboards का प्रयोग करते हुए top-selling products, daily orders, revenue trends और supplier performance जैसे व्यावसायिक मैट्रिक्स को visualize किया गया है।

## vii) सुरक्षा: Profiles, Permission Sets, FLS (Field Level Security)

डाटा की सुरक्षा के लिए Admin, Sales Staff, और Supplier के लिए अलग-अलग प्रोफाइल्स बनाई गई हैं। Field-level Security के द्वारा संवेदनशील जानकारी (जैसे बिक्री मूल्य, कस्टमर डिटेल्स) को नियंत्रित किया गया है।

### 1.1.3.1 App Launcher का उपयोग और एप्लिकेशन विवरण

Salesforce के **App Launcher** में D-Mart एप्लिकेशन सहित कई अन्य उपयोगी एप्लिकेशन उपलब्ध होते हैं जिन्हें विभिन्न CRM संबंधित कार्यों के लिए उपयोग किया जाता है। नीचे इनके उपयोग का विवरण दिया गया है:

#### D-Mart (Apni Dukaan)

- मुख्य कस्टम एप्लिकेशन जिसमें Products, Customers, Sales Orders, Suppliers, और Order Line Items जैसे कस्टम ऑब्जेक्ट्स को उपयोग में लिया गया है।
- Grocery store management से संबंधित सभी रिकॉर्ड को प्रोसेस करने और विश्लेषण करने का टूल।

#### Sales

- बिज़नेस के लीड्स, अकाउंट्स और अपॉच्युनिटी को मैनेज करने के लिए।
- बिज़नेस ग्रोथ और सेल्स पर्सन के ट्रैकिंग में उपयोगी।

#### Service / Service Console

- ग्राहकों की समस्याओं को मैनेज करने के लिए, जैसे: केस बनाना, समाधान देना आदि।
- मल्टी-चैनल कस्टमर इंटरेक्शन को एक ही स्क्रीन पर नियंत्रित करना।

#### Salesforce Chatter

- इंटरनल कम्युनिकेशन के लिए एक सोशल नेटवर्क प्लेटफॉर्म। इसमें पोस्टिंग, कमेंट और फ़िड्स के ज़रिए टीम सहयोग कर सकती है।

#### Digital Experiences

- वेबसाइट्स और पोर्टल के कंटेंट और मीडिया को मैनेज करना।

#### Automation / Process Builder / Flow

- कस्टम लॉजिक को बिना कोडिंग के क्रियान्वित करने के लिए। जैसे - ऑर्डर बनने पर ईमेल अलर्ट, स्टॉक खत्म होने पर नोटिफिकेशन आदि।

## Content

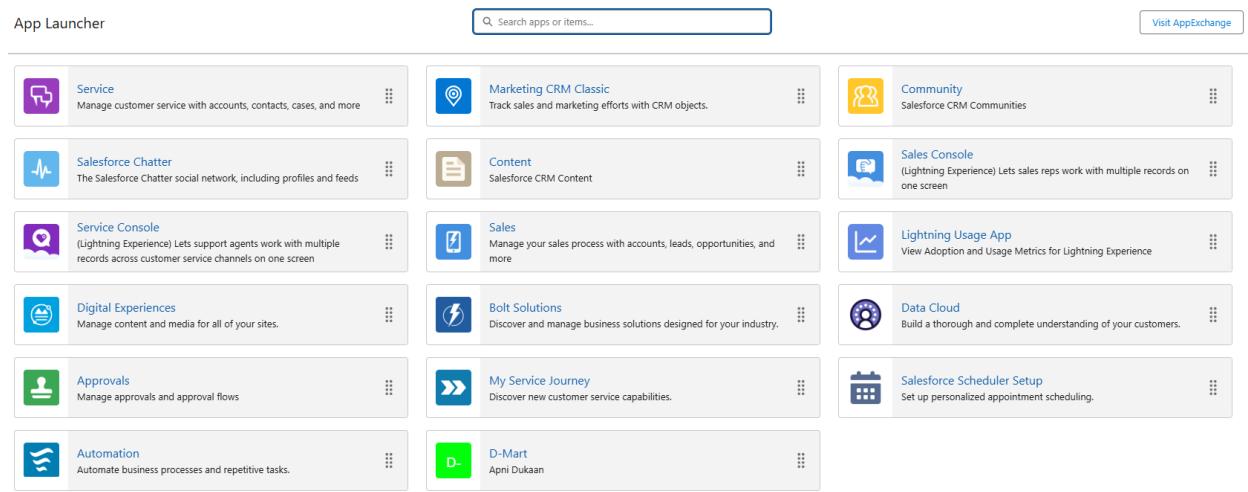
- मार्केटिंग और ग्राहक से संबंधित फ़ाइलों और कंटेंट को स्टोर और साझा करने के लिए।

## Approvals

- किसी प्रक्रिया (जैसे डिस्काउंट, रिफंड) को मैनेजर से अप्रूव कराने की व्यवस्था।

## Salesforce Scheduler Setup

- ग्राहक अपॉइंटमेंट्स को शेड्यूल करने के लिए टूल।



चित्र1 : समयबद्धक स्थापित करना (Scheduler setup)

### 1.1.4. मुख्य ऑब्जेक्ट्स और उनके फ़ील्ड्स:

D-Mart Grocery Management System को Salesforce CRM प्लेटफॉर्म पर विकसित किया गया है, जो एक स्केलेबल, सुरक्षित और क्लाउड-आधारित समाधान है। इस प्रणाली में प्रमुख ऑब्जेक्ट्स को कस्टमाइज़ करके उपयोग किया गया है ताकि ग्राहकों, उत्पादों, आपूर्तिकर्ताओं और बिक्री प्रक्रियाओं का सुचारू रूप से प्रबंधन किया जा सके।

नीचे D-Mart एप्लिकेशन में प्रयुक्त मुख्य कस्टम ऑब्जेक्ट्स और उनके फ़ील्ड्स का विवरण दिया गया है:

### i. Product (उत्पाद)

इस ऑब्जेक्ट के अंतर्गत वे सभी आइटम आते हैं जो D-Mart स्टोर में बेचे जाते हैं।

#### प्रमुख फ़ील्ड्स:

- Product\_Id - यूनिक पहचान (उदाहरण: P-0001, P-0003)
- Product\_Name - उत्पाद का नाम (जैसे: आलू, ब्रेड)
- Category - उत्पाद की श्रेणी (जैसे: Grocery, Beverages)
- Unit Price - प्रत्येक यूनिट की कीमत
- Stock Quantity - स्टॉक में उपलब्ध मात्रा
- Supplier - उत्पाद का सप्लायर (Supplier Object से रिलेशनशिप)

### ii. Customer (ग्राहक)

यह ऑब्जेक्ट उन ग्राहकों का रिकॉर्ड रखता है जो D-Mart से खरीदारी करते हैं।

#### प्रमुख फ़ील्ड्स:

- Customer\_Id
- Customer\_Name
- Phone
- Email
- Membership Status (Regular / Premium)
- Address

### iii. Sales Order (बिक्री आदेश)

यह ऑब्जेक्ट किसी ग्राहक द्वारा किए गए ऑर्डर को ट्रैक करता है।

#### प्रमुख फ़ील्ड्स:

- Order\_Id
- Order\_Date
- Customer\_Name (Customer Object से Lookup)
- Total Amount
- Payment Status (Paid / Pending)
- Delivery Status

#### **iv. Order Line Items (ऑर्डर के आइटम्स)**

यह एक child object होता है जो प्रत्येक ऑर्डर में शामिल प्रोडक्ट्स की जानकारी देता है।

##### **प्रमुख फ़िल्ड्स:**

- OrderLine\_Id
- Sales\_Order\_Id (Parent - Sales Order)
- Product\_Name (Lookup - Product)
- Quantity
- Unit Price
- Total Price

#### **v. Supplier (आपूर्तिकर्ता)**

यह उन कंपनियों या विक्रेताओं का विवरण रखता है जो D-Mart को उत्पाद प्रदान करते हैं।

##### **प्रमुख फ़िल्ड्स:**

- Supplier\_Id
- Supplier Name
- Contact Person
- Phone
- Email

चित्र 2 : वस्तु निर्माण (Object Creation)

## vi. उत्पाद ऑब्जेक्ट (Product Object) - फ़ील्ड्स का विवरण

D-Mart Grocery CRM सिस्टम में "Product" एक प्रमुख कस्टम ऑब्जेक्ट है जिसका उपयोग स्टोर में उपलब्ध सभी उत्पादों की जानकारी को संग्रहीत और प्रबंधित करने के लिए किया जाता है। प्रत्येक उत्पाद की यूनिक पहचान, श्रेणी, मूल्य, आपूर्तिकर्ता और स्टॉक जैसे विवरण इस ऑब्जेक्ट में रिकॉर्ड किए जाते हैं।

नीचे स्क्रीनशॉट में दिखाए गए Product (P-0001 - Arhar Daal) के फ़ील्ड्स का विवरण प्रस्तुत है:

- Product\_Id:** P-0001

यह यूनिक उत्पाद पहचान संख्या है जो हर आइटम को विशिष्ट बनाती है।

- Product Name: Arhar Daal  
उत्पाद का नाम जिसे स्टोर में बेचा जा रहा है।
- Product Code: P-0001  
यह प्रोडक्ट आईडी के समतुल्य एक कोड है, जिसका उपयोग इनवेंटरी ट्रैकिंग और रिपोर्टिंग में होता है।
- Category: PULSES/DALS  
यह उत्पाद की श्रेणी बताता है जिससे उसे रिपोर्टिंग और फ़िल्टरिंग में वर्गीकृत किया जा सकता है।
- Unit Price: \$75  
एक यूनिट का मूल्य (प्राइसिंग के लिए आवश्यक फ़िल्ड)।
- Stock Quantity: 1  
स्टॉक में उपलब्ध मात्रा को दर्शाता है। इस फ़िल्ड का उपयोग स्टॉक मैनेजमेंट और अलर्ट जनरेशन में होता है।
- Supplier Name: Radhey Traders  
उत्पाद किस आपूर्तिकर्ता से प्राप्त हुआ है, उसका नाम।
- Expire Date: 23/06/2026  
उत्पाद की समाप्ति तिथि, जो FMCG और Grocery प्रोजेक्ट में विशेष रूप से महत्वपूर्ण है।
- Category Item: Pulses / Daal  
श्रेणी का अधिक वर्णनात्मक नाम – इसका उपयोग UI में प्रदर्शित नामों को स्पष्ट बनाने के लिए किया गया है।
- Owner: Dipanshu Kishor Azad  
यह दिखाता है कि यह रिकॉर्ड किस यूजर द्वारा बनाया या संचालित किया गया है।
- Created By / Last Modified By:  
इस जानकारी से पता चलता है कि रिकॉर्ड किसने और कब बनाया या अंतिम बार संशोधित किया।

## **Creating of New Product**

### **Step-by-Step Guide:**

#### i. **Product\_Id:**

यह फ़िल्ड Auto-generated होता है या मैन्युअली दर्ज किया जा सकता है, जो प्रत्येक उत्पाद के लिए एक यूनिक पहचान देता है।

ii. **Product Name (Required):**

उत्पाद का नाम दर्ज करें – जैसे "Toor Dal", "Mustard Oil", "Almonds" आदि।

iii. **Product Code:**

एक यूनिक कोड जो प्रोडक्ट को इन्वेंटरी और रिपोर्ट में पहचानने में मदद करता है।

जैसे: P-0012

iv. **Category (Multi-select Picklist):**

उपलब्ध कैटेगरीज में से एक या एक से अधिक श्रेणियाँ चुनी जाती हैं:

- PULSES/DALS
- SPICE
- OILS
- NUTS

इस कैटेगरी के आधार पर उत्पाद को वर्गीकृत और फ़िल्टर किया जा सकता है।

v. **Unit Price:**

एक यूनिट का मूल्य दर्ज करें जैसे \$50, ₹120, आदि।

vi. **Stock Quantity (Required):**

स्टॉक में मौजूद यूनिट्स की संख्या डालें। यह आवश्यक फ़ील्ड है। उदाहरण: 25

vii. **Supplier Name:**

उत्पाद किस सप्लायर से आता है, वह नाम जैसे: "Radhey Traders", "Bharat Foods" आदि।

viii. **Expire Date:**

उत्पाद की समाप्ति तिथि चुनें – विशेष रूप से खाद्य उत्पादों के लिए उपयोगी।

ix. **CategoryItem (Auto-filled or derived):**

कैटेगरी के आधार पर यह फ़ील्ड स्वयं भरता है (या सिस्टम में लॉजिक से भरा जा सकता है), ताकि रिपोर्टिंग में पूरा विवरण दिखे।

Field Label	Field API Name	Data Type	Description
Product Id	Product_Id__c	Text	यूनिक प्रोडक्ट पहचान संख्या
Product Name	Name	Text	प्रोडक्ट का नाम (जैसे - Arhar Daal)
Product Code	Product_Code__c	Text	SKU या सिस्टम कोड

Category	Category__c	Picklist	प्रोडक्ट की श्रेणी (PULSES, OILS etc.)
Unit Price	Unit_Price__c	Currency	एक यूनिट का मूल्य
Stock Quantity	Stock_Quantity__c	Number	उपलब्ध स्टॉक की संख्या
Supplier Name	Supplier_Name__c	Lookup (Supplier__c)	सप्लायर के साथ रिलेशन
Expire Date	Expire_Date__c	Date	प्रोडक्ट की समाप्ति तिथि

तालिका 1: प्रोडक्ट फील्ड डाटा निर्माण (Product Field Data Creation)

## vii. Customer Object - विवरण (Description in Hindi)

The screenshot shows the D-Mart CRM application. On the left, there's a sidebar with navigation links like Home, Products, Customers, Sales Orders, Suppliers, and Order Line Items. A specific customer record for 'Sandeep Pandey' is selected. The main area displays a 'New Contact' form with fields for Name, Salutation, First Name, Last Name, Email, Phone, Account Name, and Title. Below the form, a related list pane shows activities and meetings for the contact. At the bottom right of the form, there are 'Cancel' and 'Save' buttons.

चित्र 3 : फील्ड डेटा निर्माण (Fields Data Creation)

**Customer (ग्राहक) ऑब्जेक्ट** का उपयोग उन सभी ग्राहकों की जानकारी संग्रहीत करने के लिए किया जाता है जो D-Mart स्टोर से खरीदारी करते हैं। इस ऑब्जेक्ट के माध्यम से ग्राहक के नाम, फ़ोन नंबर, ईमेल, पता आदि विवरण सुरक्षित और व्यवस्थित ढंग से स्टोर किए जाते हैं।

इस ऑब्जेक्ट के मुख्य फ़ील्ड्स निम्नलिखित हैं:

- Customer Name: ग्राहक का पूरा नाम
- Phone Number: संपर्क हेतु मोबाइल नंबर
- Email: ग्राहक की ईमेल आईडी
- Address: ग्राहक का स्थायी या डिलीवरी पता
- Created By / Modified By: यह दिखाता है कि यह रिकॉर्ड किस यूजर द्वारा कब बनाया गया था

### Customer Object का उपयोग

- ग्राहकों को यूनिक तरीके से ट्रैक करने के लिए
- प्रत्येक ऑर्डर को संबंधित ग्राहक से लिंक करने के लिए
- कस्टमर हिस्ट्री, कम्युनिकेशन और खरीद का विश्लेषण करने के लिए
- कस्टमर रिपोर्ट और डैशबोर्ड तैयार करने के लिए (जैसे "Top Customers", "Most Frequent Buyers" आदि)

### नया ग्राहक जोड़ना (Creation of Customer Record)

आपके स्क्रीनशॉट में एक नया Contact (Customer के अंतर्गत) जोड़ने का फ़ॉर्म दिखाया गया है। यह फ़ॉर्म ग्राहकों की व्यक्तिगत जानकारी दर्ज करने के लिए प्रयोग होता है।

### नया ग्राहक जोड़ने के लिए स्टेप-बाय-स्टेप प्रक्रिया:

#### i. Salutation:

ग्राहक के लिए उपसर्ग चुनें जैसे श्री (Mr.), श्रीमती (Mrs.), डॉ. आदि।

#### ii. First Name:

ग्राहक का पहला नाम दर्ज करें।

#### iii. Last Name (Required):

ग्राहक का अंतिम नाम दर्ज करना अनिवार्य है।

#### iv. Email:

ग्राहक की ईमेल आईडी, जो नोटिफिकेशन या प्रमोशनल मैसेज भेजने के लिए उपयोग की जा सकती है।

#### v. Phone:

ग्राहक का फ़ोन नंबर दर्ज करें, जो कि संपर्क और सत्यापन के लिए आवश्यक होता है।

vi. **Account Name (Optional):**

यदि ग्राहक किसी संस्था से जुड़ा है (B2B scenario), तो उस संस्था का नाम यहाँ डाला जा सकता है।

vii. **Title:**

ग्राहक की भूमिका या टाइटल – जैसे Owner, Manager, आदि।

viii. **Save:**

सारी जानकारी भरने के बाद **Save** बटन पर क्लिक करके रिकॉर्ड सिस्टम में सेव किया जाता है।

Field Label	Field API Name	Data Type	Description
Customer Name	Name	Text	ग्राहक का पूरा नाम
Phone Number	Phone_c	Phone	ग्राहक का फ़ोन नंबर
Email	Email_c	Email	ग्राहक की ईमेल ID
Address	Address_c	Text Area	ग्राहक का पता

तालिका 2: कस्टमर ऑब्जेक्ट फील्ड डाटा निर्माण (Customer Object Field Data Creation)

## Sales Order Object का विवरण

The screenshot shows a CRM interface for D-Mart. On the left, a 'Sales Order' record is displayed with ID 1, customer Sandeep Pandey, order date 23/06/2025, total amount \$60, payment method UPI, and payment status Paid. It was created by Dipanshu Kishor Azad on 24/06/2025 at 2:18 am. On the right, a 'New Contact' dialog box is open, prompting for contact details: Name, Salutation, First Name, Last Name, Email, Phone, Account Name, and Title. There are 'Cancel' and 'Save' buttons at the bottom right of the dialog.

चित्र 4 : विक्रय ऑर्डर ऑब्जेक्ट लुकअप (Sales Order Object Lookup)

## Sales Order Object क्या है?

Sales Order Object का उपयोग ग्राहक द्वारा किए गए ऑर्डर को रिकॉर्ड करने के लिए किया जाता है। इसमें ग्राहक, उत्पाद, भुगतान की जानकारी और ऑर्डर की स्थिति जैसे डिटेल्स दर्ज किए जाते हैं।

## मुख्य फ़ील्ड्स

(Fields) इस ऑब्जेक्ट में:

Field Name	विवरण (Description)
Sales Order Id	यूनिक आईडी जो हर ऑर्डर को विशिष्ट बनाता है
Customer	Lookup Field (Customer Object से जुड़ा हुआ)
Order Date	ग्राहक द्वारा ऑर्डर किए जाने की तारीख
Total Amount	ऑर्डर का कुल मूल्य (USD, INR आदि में)
Payment Method	भुगतान का तरीका जैसे: UPI, Cash, Card आदि
Payment Status	भुगतान की स्थिति जैसे: Paid, Pending, Failed आदि
Created By	यह दिखाता है कि किस उपयोगकर्ता द्वारा रिकॉर्ड बनाया गया

तालिका 3: सेल्स ऑब्जेक्ट फ़ील्ड डाटा निर्माण (Sales Object Field Data Creation)

### 1.1.5 रिपोर्टिंग और विश्लेषण (Reporting and Analytics)

Lookup Relationship - कैसे काम करता है?

- Customer:** यह एक Lookup Field है जो Customer Object से जुड़ा होता है। इससे हम यह जान पाते हैं कि यह ऑर्डर किस ग्राहक द्वारा किया गया है।
- Product या Order Line Items:** (अगर जोड़ा गया हो) इसमें यह दिखाया जाता है कि ग्राहक ने कौन से और कितने उत्पाद मंगाए हैं।

Sales Order रिकॉर्ड बनाने की प्रक्रिया (Creation Process)

i. **Customer चुनें:**

Lookup द्वारा पहले से मौजूद ग्राहक को चुनें (जैसे "Sandeep Pandey")।

ii. **Order Date दर्ज करें:**

जिस तारीख को ऑर्डर किया गया है।

iii. **Total Amount डालें:**

सभी ऑर्डर किए गए आइटम्स की कुल कीमत।

iv. **Payment Method** चुनें:

जैसे - UPI, Cash, Credit Card आदि।

v. **Payment Status:**

भुगतान हो गया है या नहीं - “Paid”, “Unpaid”, “Pending” आदि।

vi. **Save करें:**

सभी जानकारी सही तरह से भरने के बाद **Save** पर क्लिक करें।

**Sales Order Object** का उपयोग क्यों करें?

- बिक्री का रिकॉर्ड सहेजने के लिए
- ग्राहकों की खरीदारी का विश्लेषण करने के लिए
- भुगतान की स्थिति ट्रैक करने के लिए
- बिक्री रिपोर्ट बनाने के लिए

Field Label	Field API Name	Data Type	Description
Sales Order Id	Sales_Order_Id__c	Auto Number	यूनिक ऑर्डर संख्या
Customer	Customer__c	Lookup (Customer__c )	ग्राहक से कनेक्शन
Order Date	Order_Date__c	Date	ऑर्डर की तिथि
Total Amount	Total_Amount__c	Currency	ऑर्डर का कुल मूल्य
Payment Method	Payment_Method__c	Picklist	भुगतान का तरीका (UPI, Cash, etc.)
Payment Status	Payment_Status__c	Picklist	भुगतान स्थिति (Paid/Unpaid )

तालिका 4: सेल्स ऑर्डर ऑब्जेक्ट फील्ड डाटा निर्माण (Sales Order Object Field Data Creation)

## Supplier रिकॉर्ड बनाने की प्रक्रिया (Steps to Create Supplier Record)

The screenshot shows the D-Mart system interface with a 'Suppliers' module selected. A modal window titled 'New Supplier' is open, prompting for supplier information. The fields include 'Suppliers Id' (set to 001), 'Supplier Name' (Yadav Dairy), 'Phone Number', 'Email', and 'Product Supplied'. The owner of the record is listed as Dipanshu Kishor Azad. Buttons at the bottom allow for 'Cancel', 'Save & New', or 'Save'.

चित्र:5 नए आपूर्तिकर्ता में मास्टर सूची (Master List in New Supplier)

- i. **New Supplier** पर क्लिक करें
- ii. आवश्यक फ़िल्ड्स भरें:
  - a. Supplier Id: (जैसे 001)
  - b. Supplier Name: (जैसे Yadav Dairy)
  - c. Phone Number
  - d. Email
  - e. Product Supplied (जैसे - Milk, Oil आदि)
- iii. Save बटन दबाएं और रिकॉर्ड सेव हो जाएगा।

### Supplier Object के उपयोग:

- सप्लायर्स का मैनेजमेंट और रिकॉर्ड रखना
- किसी प्रोडक्ट की सोर्सिंग ट्रैक करना
- स्टॉक सप्लाई की ऑडिटिंग में मदद
- रिपोर्टिंग और सप्लायर प्रदर्शन विश्लेषण

D-Mart Grocery Management System में Supplier Object का उपयोग उन कंपनियों/फर्म्स का डेटा संरक्षित करने के लिए किया जाता है जो स्टोर को प्रोडक्ट सप्लाई करती

हैं। यह प्रबंधन को स्टॉक की सप्लाई चेन को ट्रैक करने और विश्लेषण करने में सक्षम बनाता है। यह ऑब्जेक्ट मुख्य रूप से Supplier Name, Contact Details और Product Supplied जैसे फील्ड्स को शामिल करता है।

Field Label	Field API Name	Data Type	Description
Supplier Id	Supplier_Id__c	Auto Number	यूनिक सप्लायर आईडी
Supplier Name	Name	Text	सप्लायर का नाम
Phone Number	Phone__c	Phone	सप्लायर का फ़ोन नंबर
Email	Email__c	Email	सप्लायर की ईमेल ID
Product Supplied	Product_Supplied__c	Text	जो प्रोडक्ट सप्लाई किया गया

तालिका 5: सप्लायर ऑब्जेक्ट फील्ड डाटा निर्माण (Supplier Object Field Data Creation)

## 10. Order Line-Item Object

चित्र 6 : ऑर्डर लाइन आइटम ऑब्जेक्ट इंटरलिंक लुकअप (Order Line Item Object Interlink Lookup)

### Order Line-Item Object क्या है?

Order Line Item एक कस्टम ऑब्जेक्ट है जो यह रिकॉर्ड करता है कि किसी एक Sales Order में कौन-कौन से प्रोडक्ट कितनी मात्रा (Quantity) में खरीदे गए हैं और उनकी कीमत

(Unit Price) क्या है। यह एक junction object की तरह कार्य करता है जो Sales Order और Products को आपस में जोड़ता है।

### मुख्य फ़ील्ड्स (Fields):

Field Name	विवरण (Description)
Order Line-Item Id	यूनिक ID जो हर Line Item को अलग पहचान देता है
Sales Order	Lookup Field - यह उस Sales Order से लिंक होता है
Product	Lookup Field - यह उस Product से लिंक होता है जो खरीदा गया
Quantity	खरीदी गई यूनिट्स की संख्या
Unit Price	प्रति यूनिट की कीमत

तालिका 6: फ़ील्ड डाटा (Field Data)

### रिलेशनशिप:

Object	Type	Description
Sales Order	Lookup	एक Sales Order में कई Order Line Items हो सकते हैं
Product	Lookup	एक Product को कई Order Line Items में जोड़ा जा सकता है

तालिका 7: लुकअप रिलेशन्स (Lookup Relations)

### Order Line Item कैसे Create करें?

- Order Line Items > New पर क्लिक करें
- आवश्यक फ़ील्ड्स भरें:
  - Order Line Item Id: (जैसे 112122)
  - Sales Order: संबंधित Sales Order से लिंक करें
  - Product: उत्पाद का चयन करें
  - Quantity: कितनी यूनिट खरीदी गई
  - Unit Price: प्रति यूनिट मूल्य
- Save बटन दबाएं

### Order Line Item Object के उपयोग:

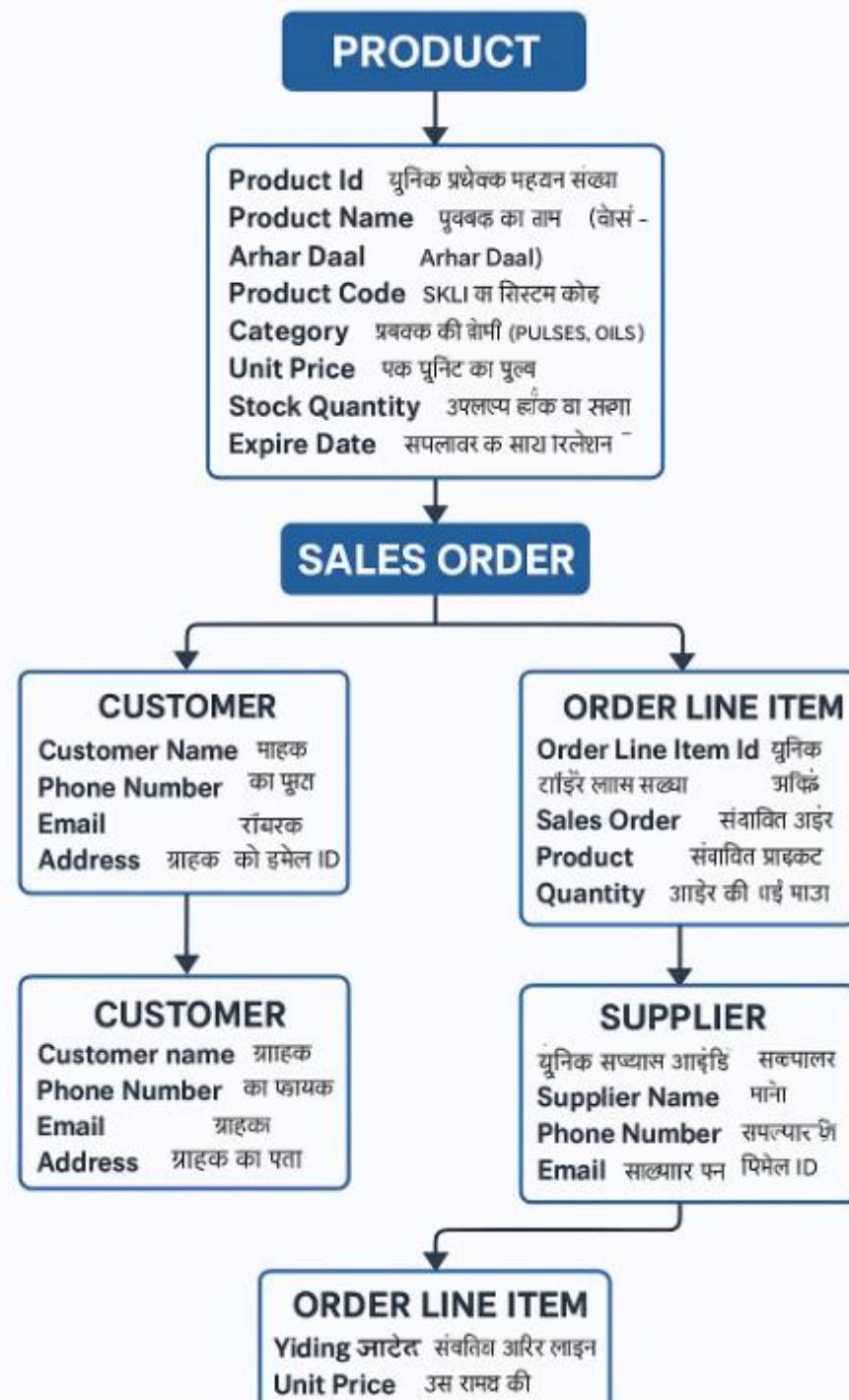
- यह Object किसी भी Sales Order के भीतर खरीदे गए सभी प्रोडक्ट्स की जानकारी सहेजता है।
- प्रत्येक आइटम की Quantity और Price स्टोर करता है जिससे टोटल ऑर्डर वैल्यू का कैलकुलेशन हो सके।
- रिपोर्टिंग, इन्वेंट्री ट्रैकिंग, और बिलिंग में इसका महत्वपूर्ण उपयोग होता है।

Field Label	Field API Name	Data Type	Description
Order Line Item Id	Order_Line_Item_Id_c	Auto Number	यूनिक ऑर्डर लाइन संख्या
Sales Order	Sales_Order__c	Lookup (Sales_Order__c)	संबंधित ऑर्डर
Product	Product__c	Lookup (Product__c)	संबंधित प्रोडक्ट
Quantity	Quantity__c	Number	ऑर्डर की गई मात्रा
Unit Price	Unit_Price__c	Currency	उस समय की प्रोडक्ट की कीमत

तालिका 8: मास्टर लुकअप रिलेशन्स (Master Lookup Relations)

**Order Line Item Object** D-Mart Grocery CRM System का वह भाग है जो किसी एक बिक्री आदेश (Sales Order) में शामिल किए गए प्रत्येक उत्पाद की यूनिट, कीमत और लिंक्ड प्रोडक्ट की जानकारी को स्टोर करता है। यह Object Sales Order और Product के बीच रिलेशनशिप बनाता है और ऑर्डर की सही-सही जानकारी संरक्षित करता है।

## Objects and Fields Cloud Architecture



चित्र 7 : क्लाउड आर्किटेक्चर (Cloud Architecture)

### 1.1.6 निष्कर्ष (Conclusion)

D-Mart Grocery Management System एक क्लाउड-बेस्ड एप्लिकेशन है जिसे Salesforce CRM प्लेटफ़ॉर्म पर विकसित किया गया है। इसका मुख्य उद्देश्य ग्रॉसरी व्यवसाय में ग्राहकों, उत्पादों, सप्लायर्स, ऑर्डर और भुगतान जैसी सूचनाओं का समुचित प्रबंधन करना है। यह सिस्टम पूरी तरह कस्टम ऑब्जेक्ट्स और रिलेशनशिप्स पर आधारित है, जो सभी मॉड्यूल को एक दूसरे से जोड़ता है और डेटा को सुरक्षित व एकीकृत बनाता है।

इस परियोजना में मुख्य ऑब्जेक्ट्स हैं – Product, Customer, Supplier, Sales Order, और Order Line Item। Product ऑब्जेक्ट में प्रोडक्ट का नाम, मूल्य, स्टॉक और सप्लायर की जानकारी होती है। Customer ऑब्जेक्ट में ग्राहक की व्यक्तिगत डिटेल जैसे नाम, फोन, ईमेल और पता शामिल होते हैं। Supplier ऑब्जेक्ट प्रोडक्ट की सप्लाई करने वाले विक्रेताओं को मैनेज करता है। Sales Order ऑब्जेक्ट ग्राहकों द्वारा किए गए ऑर्डर की जानकारी रखता है, जबकि Order Line Item बताता है कि किस ऑर्डर में कौन-कौन से प्रोडक्ट कितनी मात्रा में खरीदे गए हैं।

Salesforce का Lightning Platform, Apex Code, Validation Rules और Process Builder जैसी तकनीकों का उपयोग करके इस प्रणाली को शक्तिशाली और उपयोगकर्ता-अनुकूल बनाया गया है। Lookup Relationships के माध्यम से सभी ऑब्जेक्ट्स एक-दूसरे से जुड़े हुए हैं जिससे डाटा consistency बनी रहती है।

इस प्रोजेक्ट का मुख्य लाभ यह है कि यह एक Grocery Store के सभी operations को एक ही सिस्टम में एकीकृत करता है और ऑटोमेशन, रिपोर्टिंग और डेटा सुरक्षा सुनिश्चित करता है। इसका यूजर फ्रेंडली इंटरफ़ेस और क्लाउड एक्सेसिलिटी इसे कहीं से भी एक्सेस करने योग्य बनाता है, जो आज के डिजिटल युग में अत्यंत आवश्यक है।

यह सिस्टम छोटे से लेकर बड़े स्तर के ग्रॉसरी स्टोर्स के लिए एक परफेक्ट समाधान है।

# 2.1 Speech to Speech Translator in Real Time

## वास्तविक समय में भाषण से भाषण अनुवादक (Speech to Speech Translator in Real Time)

### 2.1.1. परिचय (Introduction)

तकनीकी युग में जब भाषाओं की विविधता संवाद की बाधा बनती है, तब ऐसे उपकरणों की आवश्यकता होती है जो भाषा को तुरंत अनुवाद करके उपयोगकर्ताओं को आपस में जोड़ सकें। हमारा यह प्रोजेक्ट "हिंदी से अंग्रेजी वॉयस ट्रांसलेटर वेब एप्लिकेशन" इसी उद्देश्य को पूरा करता है। यह एक ऐसा वेब आधारित टूल है जो उपयोगकर्ता द्वारा बोले गए हिंदी वाक्य को सुनता है, उसे पहचानता है, उसका अंग्रेजी में अनुवाद करता है, और अंततः उसका उच्चारण भी करता है।

यह प्रोजेक्ट विशेष रूप से उन उपयोगकर्ताओं के लिए उपयोगी है जिन्हें अंग्रेजी भाषा समझने में कठिनाई होती है या जो हिंदी में बोलकर अंग्रेजी में संवाद स्थापित करना चाहते हैं।



चित्र 8 : Output & File Structure

### 2.1.2 उद्देश्य (Objective)

इस प्रोजेक्ट का मुख्य उद्देश्य है:

- हिंदी में बोले गए वाक्य को मशीन के द्वारा पहचानना।
- उस वाक्य का अंग्रेजी में सटीक अनुवाद करना।
- अनुवादित वाक्य को ऑडियो के रूप में प्रस्तुत करना।
- एक यूजर-फ्रेंडली वेब इंटरफ़ेस पर इन सभी कार्यों को प्रदर्शित करना।

यह एप्लिकेशन छात्रों, यात्रियों, भाषायी बाधा का सामना कर रहे लोगों और व्यवसायिक उपयोगकर्ताओं के लिए बेहद उपयोगी हो सकता है।

### 2.1.3 प्रयुक्त तकनीकें (Technologies Used)

इस प्रोजेक्ट को विकसित करने के लिए निम्नलिखित तकनीकों और लाइब्रेरीज़ का उपयोग किया गया है:

- **Python** - प्रोग्रामिंग भाषा के रूप में
- **Flask** - वेब एप्लिकेशन फ्रेमवर्क
- **SpeechRecognition** - वॉइस इनपुट को टेक्स्ट में बदलने के लिए
- **Googletrans** - हिंदी से अंग्रेज़ी में अनुवाद के लिए
- **gTTS (Google Text-to-Speech)** - टेक्स्ट को ऑडियो में बदलने के लिए
- **HTML/CSS** - वेब इंटरफ़ेस के लिए
- **JavaScript (आंशिक)** - ऑटो-प्ले, UI इफेक्ट्स आदि के लिए
- **PyAudio** - माइक्रोफोन से ऑडियो रिकॉर्डिंग के लिए

### 2.1.4 कार्यप्रणाली (Working Process)

इस वेब एप्लिकेशन की कार्यप्रणाली निम्न प्रकार से है:

- i. **बोलना** - उपयोगकर्ता "🎙️ बोलिए" बटन पर क्लिक करता है।
- ii. **रिकॉर्डिंग** - सिस्टम माइक्रोफोन से आवाज़ रिकॉर्ड करता है।
- iii. **वॉइस रिकिनिशन** - speech\_recognition लाइब्रेरी के माध्यम से हिंदी वाक्य को पहचानता है।
- iv. **अनुवाद** - googletrans लाइब्रेरी से हिंदी वाक्य का अंग्रेज़ी में अनुवाद किया जाता है।
- v. **स्पीच जेनरेशन** - gTTS द्वारा अनुवादित वाक्य को ऑडियो (MP3) फॉर्म में बदला जाता है।
- vi. **वेब पर प्रजेंटेशन** - वेब पेज पर हिंदी वाक्य, अंग्रेज़ी अनुवाद और प्ले करने योग्य ऑडियो दिखाया जाता है।

### कोड का संक्षिप्त वर्णन (Code Summary)

#### a. app.py:

- `listen microphone()` - माइक्रोफोन से आवाज़ सुनने व टेक्स्ट में बदलने का कार्य करता है।

- `translate_text()` – हिंदी से अंग्रेज़ी में अनुवाद करता है।
- `speak_text()` – gTTS की सहायता से ऑडियो फाइल तैयार करता है।
- `index()` – Flask का रूट रैंडर करता है, जहाँ POST रिक्वेस्ट से रिकॉर्डिंग होती है और आउटपुट दिखाया जाता है।

#### b. `index.html`:

- HTML पेज जिसमें एक बटन है ("🎙️ बोलिए"), जो POST अनुरोध भेजता है।
- यूजर के बोले गए वाक्य, अनुवादित वाक्य और ऑडियो को वेब पेज पर दिखाया जाता है।
- CSS के माध्यम से एक "listening" एनिमेशन भी दिखाया गया है।

### आउटपुट विवरण (Output Description)

जब उपयोगकर्ता "🎙️ बोलिए" बटन पर क्लिक करता है:

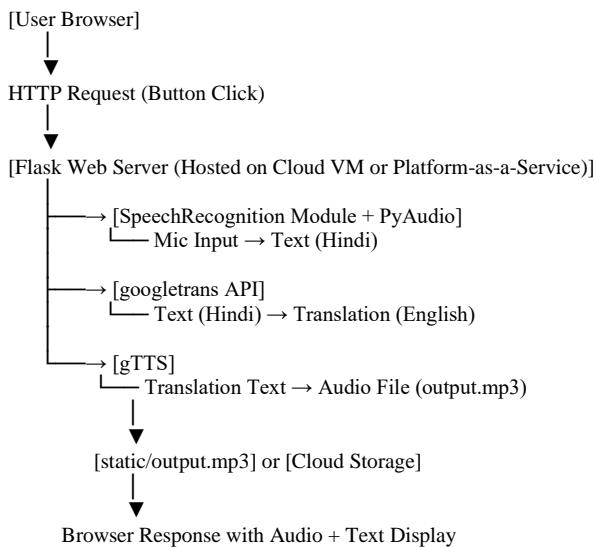
- एक लाल रंग का माइक्रोफोन एनीमेशन चालू होता है (जिससे पता चलता है कि सिस्टम सुन रहा है)।
- उपयोगकर्ता हिंदी में बोलता है, जैसे: "मुझे पानी चाहिए!"
- ऐप उसका अनुवाद करता है: "I need water."
- अनुवादित टेक्स्ट वेबपेज पर दिखाई देता है।
- एक ऑडियो प्लेयर भी दिखता है जिसमें "I need water." का उच्चारण ऑटो-प्ले होता है।

### विशेषताएं (Key Features)

- हिंदी वॉइस इनपुट
- अंग्रेज़ी अनुवाद
- ऑटो-प्ले ऑडियो
- वेब-आधारित इंटरफ़ेस
- एनिमेटेड UI (Listening indicator)
- फ्री और ओपन-सोर्स टूल्स का उपयोग

## 2.1.5 रिपोर्टिंग और विश्लेषण (Reporting and Analytics)

### Deployment (डिप्लॉयमेंट)



### Cloud Architecture (क्लाउड आर्किटेक्चर)

#### i. Client (Browser)

- उपयोगकर्ता का वेब ब्राउज़र जो Flask ऐप से इंटरैक्ट करता है।

#### ii. Web Server (Flask App)

- Python आधारित Flask एप्लिकेशन जो यूज़र रिक्वेस्ट हैंडल करता है।
- Speech को टेक्स्ट में बदलता है, अनुवाद करता है और ऑडियो जेनरेट करता है।

#### iii. Speech Recognition Service

- SpeechRecognition + PyAudio का उपयोग कर माइक्रोफोन इनपुट को प्रोसेस करता है।
- इसे Future में Google Cloud Speech-to-Text API से जोड़ा जा सकता है।

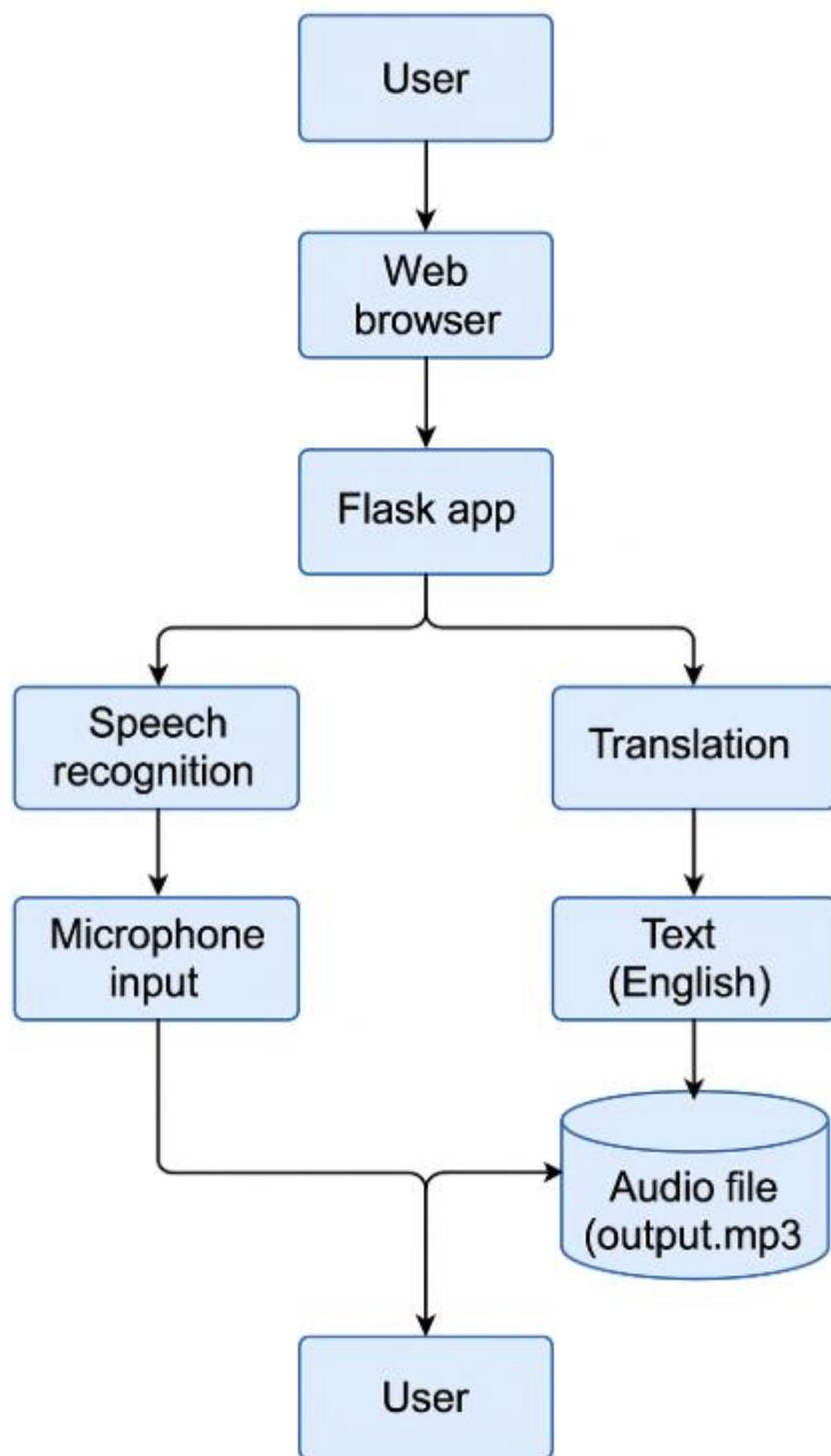
#### iv. Translation Service

- googletrans पैकेज (जो Google Translate API को यूज़ करता है)।

#### vi. Text-to-Speech Service

- gTTS (Google Text to Speech) लाइब्रेरी MP3 ऑडियो जेनरेट करती है।

## डेटा प्रवाह आरेख (Data Flow Diagram)



चित्र 9 : Data Flow Diagram

## एल्गोरिदम (Algorithm)

### चरण 1: प्रारंभ करें

प्रोग्राम शुरू करें और आवश्यक लाइब्रेरी इंपोर्ट करें:

```
speech_recognition, googletrans, gTTS, Flask, playsound, os
```

### चरण 2: माइक्रोफोन से इनपुट प्राप्त करें

- यूजर को बोलने के लिए सूचित करें (" बोलिए...")
- speech\_recognition का उपयोग करके यूजर की आवाज़ को कैप्चर करें।
- अगर आवाज़ पहचानी गईः
  - उसका टेक्स्ट प्राप्त करें
  - अन्यथा "आवाज़ समझ नहीं आई" प्रदर्शित करें

### चरण 3: प्राप्त टेक्स्ट का अनुवाद करें

- googletrans का प्रयोग करके हिन्दी टेक्स्ट को अंग्रेज़ी में अनुवाद करें
- अनुवादित टेक्स्ट को प्रदर्शित करें

### चरण 4: टेक्स्ट को आवाज़ में बदलें

- gTTS का प्रयोग करें
- अनुवादित टेक्स्ट को MP3 फॉर्मेट में सेव करें (उदाहरण: output.mp3)
- MP3 फाइल को static/ फ़ोल्डर में सेव करें

### चरण 5: वेबपेज पर आउटपुट दिखाएं

- HTML टेम्प्लेट में:
  - अनुवादित टेक्स्ट दिखाएं
  - MP3 फाइल को <audio autoplay> टैग से चलाएं

### चरण 6: यदि आवश्यक हो, तो प्रक्रिया दोहराएं

- नई रिक्वेस्ट आने पर फिर से इनपुट लें और पूरे प्रोसेस को दोहराएं

### चरण 7: अंत करें

- सभी फाइलों और संसाधनों को क्लोज करें
- सर्वर चलता रहे जब तक यूजर बंद न करे

## 2.1.6 निष्कर्ष (Conclusion)

यह प्रोजेक्ट हिंदी भाषी उपयोगकर्ताओं को अंग्रेज़ी में संवाद स्थापित करने में मदद करता है। यह विशेष रूप से शिक्षा, ट्रैवल और बिज़नेस के क्षेत्र में उपयोगी साबित हो सकता है। इस वेब एप्लिकेशन की सबसे बड़ी खूबी है - इसकी सादगी, सटीकता और ब्राउज़र-आधारित इंटरफ़ेस। भविष्य में इसमें और भाषाओं का समर्थन, ऑफलाइन कार्यक्षमता और मोबाइल ऐप के रूप में विस्तार संभव है।

# 2.2 Text Summerization

## पाठ सारांशीकरण (Text Summarization)

### 2.2.1 परिचय (Introduction)

आज के डिजिटल युग में सूचनाओं की मात्रा अत्यधिक बढ़ चुकी है। किताबें, वेबसाइटें, समाचार लेख, ब्लॉग और दस्तावेज़ इतनी अधिक मात्रा में उपलब्ध हैं कि उन्हें पूरा पढ़ना और समझना मुश्किल हो गया है। ऐसी स्थिति में स्वचालित पाठ सारांशीकरण (Automatic Text Summarization) एक महत्वपूर्ण तकनीक बन गई है जो किसी बड़े दस्तावेज़ का सारांश निकालने में मदद करती है। इस परियोजना में हमने Google द्वारा विकसित PEGASUS मॉडल का उपयोग करके एक वेब एप्लिकेशन तैयार किया है जो उपयोगकर्ता द्वारा दिए गए किसी भी टेक्स्ट का संक्षिप्त सारांश उत्पन्न करता है।

### 2.2.2 उद्देश्य (Objective)

इस प्रोजेक्ट का मुख्य उद्देश्य एक ऐसा वेब-आधारित टूल बनाना था, जो किसी भी लंबे पाठ (Text) को इनपुट के रूप में लेकर उसका सारगर्भित और सटीक सारांश उत्पन्न कर सके। इस टूल को छात्रों, शोधकर्ताओं, पत्रकारों और किसी भी सामान्य उपयोगकर्ता द्वारा उपयोग किया जा सकता है जो बड़े डॉक्युमेंट्स को जल्दी समझना चाहता हो।

### प्रयुक्त तकनीकें (Technologies Used)

इस प्रोजेक्ट में निम्नलिखित तकनीकों और टूल्स का प्रयोग किया गया है:

- प्रोग्रामिंग भाषा: Python
- वेब फ्रेमवर्क: Flask
- मशीन लर्निंग मॉडल: PEGASUS (Pre-trained model by Google)
- लाइब्रेरी: Transformers (HuggingFace), Bootstrap (UI के लिए)
- फ्रंटエंड: HTML, CSS (Bootstrap द्वारा स्टाइलिंग)

### NLTK (Natural Language Toolkit)

- यह Natural Language Processing (NLP) के लिए प्रमुख टूलकिट है।
- हमने इसका प्रयोग Stopwords हटाने, Lemmatization करने और शब्दों को टोकनाइज़ करने में किया।
- यह टेक्स्ट को साफ और मशीन के लिए उपयोगी रूप में बदलने में सहायक रहा।

## TensorFlow / Keras

- यह परियोजना का मूल Deep Learning Framework है।
- Keras एक high-level API है जो TensorFlow के ऊपर काम करता है और मॉडल बनाना अधिक आसान बनाता है।
- इसमें हमने “Embedding, Bidirectional LSTM, Dense Layers, Batch Normalization, Dropout, और Softmax Output” का प्रयोग किया।
- साथ ही EarlyStopping और ReduceLROnPlateau जैसे callbacks का उपयोग कर के ट्रेनिंग को स्थिर और बेहतर बनाया गया।

## Scikit-learn

- यह मशीन लर्निंग की एक लोकप्रिय लाइब्रेरी है।
- हमने इसका उपयोग मुख्य रूप से:
  - डेटा को ट्रेनिंग और टेस्ट सेट में विभाजित करने के लिए (train\_test\_split)
  - “One-hot Encoding” करने के लिए किया (क्योंकि हमारे आउटपुट में 3 कक्षाएं थीं)
  - अविष्य में Confusion Matrix या Classification Report जैसी सुविधाएं भी इस लाइब्रेरी से प्राप्त की जा सकती हैं।

## टोकनाइज़ेशन (Tokenization)

टोकनाइज़ेशन वह प्रक्रिया है जिसमें किसी टेक्स्ट (जैसे ट्वीट) को शब्दों (Tokens) में विभाजित किया जाता है और फिर हर शब्द को एक यूनिक संख्या प्रदान की जाती है।

हमने इसके लिए Keras की Tokenizer क्लास का उपयोग किया:

- पहले ट्रेनिंग डेटा (ट्वीट्स) पर .fit\_on\_texts() फंक्शन का प्रयोग किया गया जिससे शब्दों का एक शब्दकोश (word index) तैयार हुआ।
- इसके बाद .texts\_to\_sequences() का उपयोग करके हर ट्वीट को एक संख्यात्मक अनुक्रम (sequence of integers) में बदला गया।

उदाहरण:

makefile

CopyEdit

टेक्स्ट: "hate speech is dangerous"

Tokenized: [15, 8, 3, 49]

## पैडिंग (Padding)

टोकनाइज़ेशन के बाद, हर ट्वीट की लंबाई अलग-अलग हो सकती है। लेकिन डीप लर्निंग मॉडल में इनपुट की लंबाई एक समान होनी चाहिए। इसलिए, हमने Keras की pad\_sequences() फंक्शन का उपयोग किया जिससे हर ट्वीट को अधिकतम 100 शब्दों की लंबाई तक pad किया गया। यदि कोई ट्वीट छोटा हो तो उसमें शून्य (0) जोड़े गए, और अगर लंबा हो तो उसे काट दिया गया।

इस प्रक्रिया को "post-padding" और "post-truncating" कहा जाता है।

## पर्यावरण सेटअप (Environment Setup):

### 1. प्रोग्रामिंग लैंग्वेज और फ्रेमवर्क:

- **Python:** बैकएंड लॉजिक के लिए प्रयोग हुआ।
- **Flask:** एक लाइटवेट वेब फ्रेमवर्क जो HTML फॉर्म को प्रोसेस करने और मॉडल को यूज़र इनपुट से जोड़ने का कार्य करता है।

### 2. फ्रंटएंड:

- **HTML:** वेबपेज की संरचना के लिए।
- **Bootstrap:** वेबसाइट को आकर्षक और उत्तरदायी (responsive) बनाने के लिए।

### 3. मशीन लर्निंग मॉडल और लाइब्रेरीज़:

- **Transformers (Hugging Face):** यह एक शक्तिशाली ओपन-सोर्स लाइब्रेरी है जो NLP टास्क के लिए कई प्री-ट्रेन्ड मॉडल्स देती है। इसमें PEGASUS भी शामिल है।
- **PEGASUS-XSUM Model:** Google द्वारा बनाया गया ट्रांसफॉर्मर मॉडल जो विशेष रूप से टेक्स्ट समरी के लिए डिजाइन किया गया है।
- **torch (PyTorch):** यह मॉडल के लिए फ्रेमवर्क सपोर्ट देता है।

### **मुख्य घटक (Main Components):**

#### i. मॉडल लोडिंग:

इस प्रोजेक्ट में सबसे पहले PegasusTokenizer और PegasusForConditionalGeneration क्लास का उपयोग करके मॉडल और टोकनाइज़र को लोड किया गया। यह एक बार लोड होने के बाद पूरी एप्लिकेशन में उपयोग होता है।

#### ii. पाइपलाइन (pipeline):

HuggingFace की pipeline() सुविधा का प्रयोग करके टेक्स्ट समरी के लिए एक summarization पाइपलाइन तैयार की गई। यह पाइपलाइन मॉडल और टोकनाइज़र को इनपुट लेकर स्वचालित रूप से टेक्स्ट को सारांश में बदल देती है।

#### iii. वेब इंटरफ़ेस:

Flask के साथ एक HTML पेज बनाया गया है जहाँ यूज़र टेक्स्ट इनपुट कर सकता है। जब यूज़र "Summarize" बटन पर क्लिक करता है, तो Flask बैकएंड में दिए गए टेक्स्ट को मॉडल के पास भेजता है और प्राप्त सारांश को वेबसाइट पर वापस दिखाया जाता है।

### **2.2.3 PEGASUS मॉडल का परिचय**

PEGASUS (Pre-training with Extracted Gap-sentences for Abstractive Summarization) एक शक्तिशाली ट्रांसफॉर्मर आधारित मॉडल है जिसे विशेष रूप से सारांश निर्माण (Text Summarization) के लिए डिज़ाइन किया गया है। इसे Google द्वारा विकसित किया गया है और यह HuggingFace Transformers लाइब्रेरी में उपलब्ध है।

PEGASUS अन्य भाषा मॉडल्स से इस मामले में भिन्न है कि इसे बड़े पैमाने पर सार्वजनिक दस्तावेज़ों और लेखों पर प्रशिक्षित किया गया है, जिससे यह उच्च गुणवत्ता वाला सारांश उत्पन्न करने में सक्षम है। हमने अपने प्रोजेक्ट में "google/pegasus-xsum" वर्जन का उपयोग किया है जो मुख्यतः संक्षिप्त सारांश प्रदान करता है।

### **2.2.4 कार्यप्रणाली (Workflow)**

#### I) यूज़र इंटरफ़ेस (UI):

उपयोगकर्ता को एक टेक्स्ट-एरिया प्रदान किया जाता है जहाँ वे अपना टेक्स्ट पेस्ट कर सकते हैं। इसके बाद 'Summarize' बटन पर क्लिक करके प्रोसेस प्रारंभ किया जाता है।

## II) बैकएंड प्रोसेसिंग:

- Flask सर्वर टेक्स्ट को प्राप्त करता है।
- PEGASUS Tokenizer टेक्स्ट को टोकन में बदलता है।
- मॉडल generate() फंक्शन द्वारा सारांश उत्पन्न करता है।
- अंतिम सारांश को decode कर के UI पर दिखाया जाता है।

## III) परिणाम प्रदर्शित करना:

सारांश को Bootstrap आधारित सुंदर UI में "Summary" खंड के तहत दिखाया जाता है।

## मुख्य कोड संरचना

### i. app.py

यह फाइल Flask एप्लिकेशन का मुख्य भाग है। इसमें:

- मॉडल लोड होता है
- एक Summarizer pipeline तैयार होता है
- HTML फॉर्म से प्राप्त टेक्स्ट को प्रोसेस किया जाता है
- सारांश को HTML पेज पर रेंडर किया जाता है

### ii. index.html

यह फाइल उपयोगकर्ता इंटरफ़ेस का कार्य करती है। यह एक फॉर्म प्रदान करती है जिसमें यूजर टेक्स्ट दर्ज करता है और सारांश परिणाम देखता है। Bootstrap का उपयोग करके इसे आकर्षक रूप दिया गया है।

### iii. text\_summary.py

यह स्क्रिप्ट स्टैंडअलोन summarization के लिए है। इसमें PEGASUS Tokenizer और Model का उपयोग कर के सीधे टेक्स्ट का सारांश उत्पन्न किया जाता है।

## विशेषताएँ (Features)

- यूजर-फ्रेंडली इंटरफ़ेस

- तेज़ और प्रभावशाली सारांश
- न्यूनतम कोड में प्रभावी आउटपुट
- हिंदी/अंग्रेज़ी किसी भी भाषा के टेक्स्ट को प्रोसेस करने में सक्षम (मुख्य रूप से अंग्रेज़ी के लिए ट्रैड है)
- क्लाउड या लोकलहोस्ट पर चलाया जा सकता है

### लाभ (Benefits)

- लंबे लेख पढ़ने में समय की बचत
- छात्रों के लिए निबंध, शोध-पत्र आदि का सारांश बनाना आसान
- पत्रकारों और ब्लॉगरों के लिए सहायक
- अकादमिक व तकनीकी दस्तावेजों को जल्दी समझने में मदद

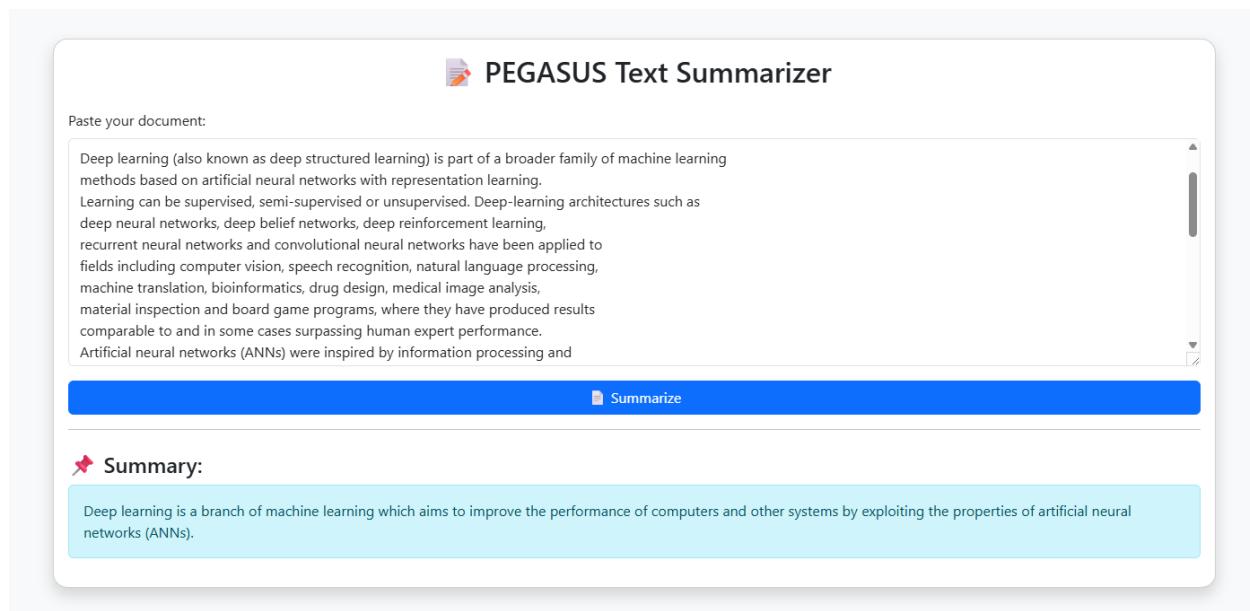
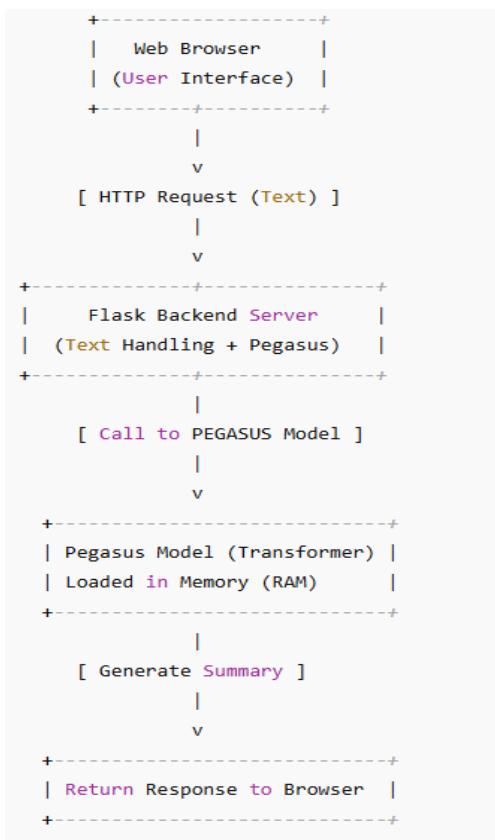
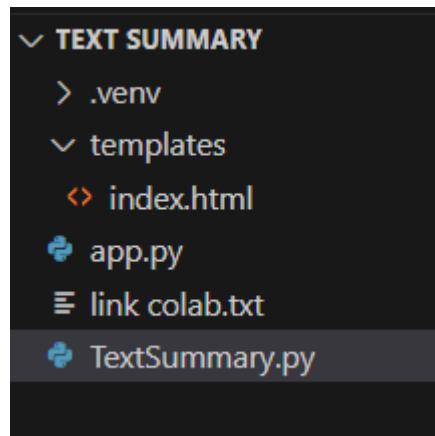


Fig10 : Output PEGASUS Text Summarizer

## 2.2.5 Architecture Flow (आर्किटेक्चर फ्लो)



## File Structure (फाइल संरचना)



चित्र 11 : Architecture Flow & File Structure

## 2.2.6 एल्गोरिदम (Algorithm)

### चरण 1: आवश्यक लाइब्रेरी आयात करें

- Flask, transformers, PegasusTokenizer, PegasusForConditionalGeneration और अन्य जरूरी मॉड्यूल को लोड करें।

### चरण 2: मॉडल और टोकनाइज़र लोड करें

- Google Pegasus XSUM मॉडल का नाम सेट करें।
- Hugging Face से मॉडल और टोकनाइज़र को लोड करें।

### चरण 3: Flask ऐप प्रारंभ करें

- Flask एप्लिकेशन को इनिशियलाइज़ करें और मुख्य ऐप बनाएँ।

### चरण 4: होमपेज रूट बनाएं

- "/" रूट सेट करें, जो GET और POST दोनों रिक्वेस्ट को संभाल सके।
- GET रिक्वेस्ट पर इनपुट टेक्स्ट बॉक्स दिखाएँ।
- POST रिक्वेस्ट पर यूज़र से मिला टेक्स्ट प्राप्त करें।

### चरण 5: इनपुट टेक्स्ट को प्रोसेस करें

- यूजर के इनपुट को टोकन में बदलें, ट्रंकेट और पैड करें ताकि यह मॉडल में उपयोग हो सके।

#### चरण 6: सारांश जनरेट करें

- मॉडल से टेक्स्ट का सारांश निकालें।
- प्राप्त सारांश को डिकोड करें ताकि वह पठनीय हो।

#### चरण 7: सारांश को यूजर को दिखाएँ

- HTML टेम्पलेट के माध्यम से ऑरिजिनल टेक्स्ट और उसका सारांश स्क्रीन पर प्रदर्शित करें।

**एल्गोरिदम: टेक्स्ट समरीकरण (PEGASUS मॉडल आधारित)**

#### चरण 1: प्रारंभ करें

सिस्टम की प्रारंभिक आवश्यकताओं को लोड करें।

#### चरण 2: मॉडल और टोकनाइज़र लोड करें

- Google का प्री-ट्रेन्ड मॉडल pegasus-xsum लोड करें।
- मॉडल और टोकनाइज़र को HuggingFace Transformers से इम्पोर्ट करें।

```
python
CopyEdit
tokenizer = PegasusTokenizer.from_pretrained("google/pegasus-xsum")
model = PegasusForConditionalGeneration.from_pretrained("google/pegasus-xsum")
```

#### चरण 3: उपयोगकर्ता से इनपुट टेक्स्ट प्राप्त करें

- उपयोगकर्ता से टेक्स्ट इनपुट (किसी लेख, पैराग्राफ या दस्तावेज़ का कंटेंट) प्राप्त करें।
- यह इनपुट वेब फॉर्म (HTML) या Python स्क्रिप्ट में दर्ज किया जा सकता है।

```
html
CopyEdit
<textarea name="text">...</textarea>
```

#### चरण 4: टेक्स्ट को टोकनाइज़ करें

- इनपुट टेक्स्ट को मॉडल के लिए उपयुक्त फॉर्मेट में बदलें।
- Padding और truncation लागू करें ताकि टेक्स्ट लिमिट में रहे।

```
python
CopyEdit
```

```
tokens = tokenizer(text, truncation=True, padding="longest",
return_tensors="pt")
```

## चरण 5: सारांश उत्पन्न करें (Generate Summary)

- टोकन किए गए टेक्स्ट को मॉडल में भेजें।
- मॉडल generate() फंक्शन द्वारा आउटपुट टोकन तैयार करता है।

```
python
CopyEdit
summary_ids = model.generate(**tokens)
```

## चरण 6: आउटपुट डिकोड करें

- आउटपुट टोकनों को फिर से सामान्य टेक्स्ट (सारांश) में बदलें।

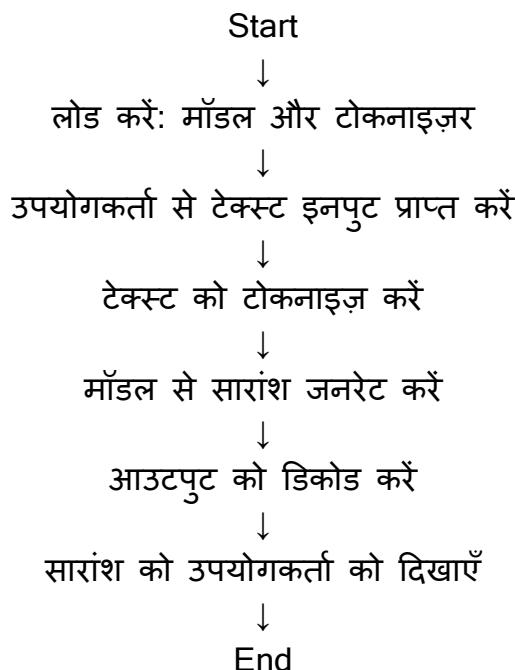
```
python
CopyEdit
summary_text = tokenizer.decode(summary_ids[0], skip_special_tokens=True)
```

## चरण 7: सारांश उपयोगकर्ता को दिखाएँ

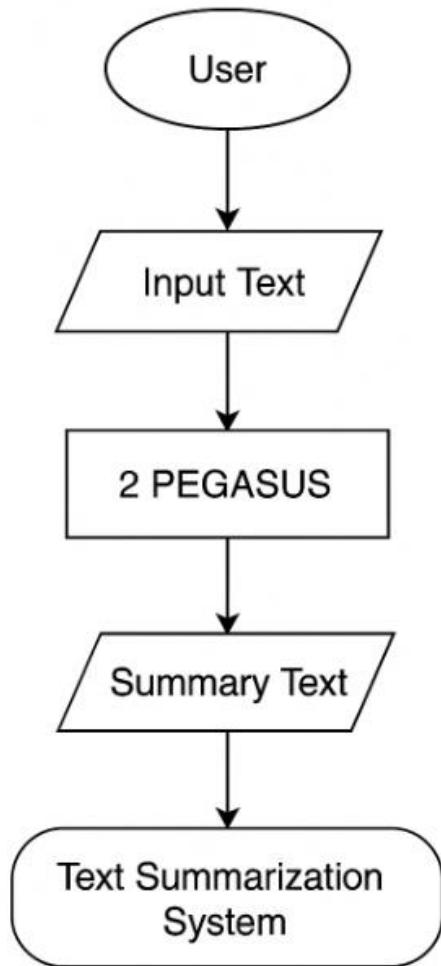
- वेब एप्लिकेशन में HTML के माध्यम से सारांश प्रदर्शित करें।
- Flask render\_template() के साथ उपयोग करें।

```
html
CopyEdit
<div class="alert alert-info">{{ summary_text }}</div>
```

## चरण 8: समाप्त करें



## डेटा प्रवाह आरेख (Data Flow Diagram)

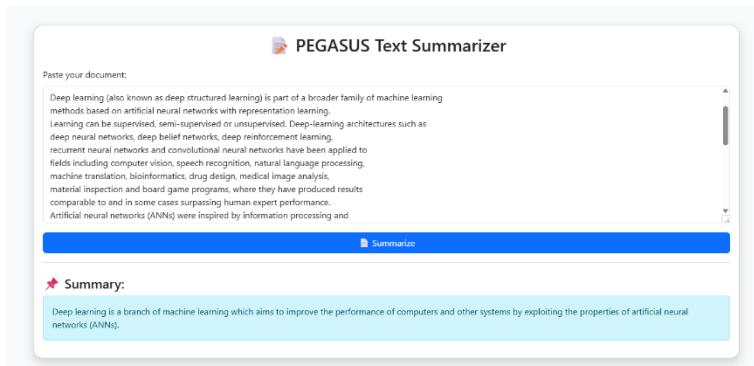


### 2.2.7 निष्कर्ष (Conclusion)

यह टेक्स्ट समरीकरण प्रणाली एक उपयोगी और प्रभावशाली टूल है जो आज के सूचना युग में समय बचाने और जानकारी को जल्दी समझने में मदद करता है। PEGASUS जैसे प्री-ट्रेंड डीप लर्निंग मॉडल्स के माध्यम से हम मानव जैसी टेक्स्ट प्रोसेसिंग क्षमताएं विकसित कर पा रहे हैं। इस प्रोजेक्ट के माध्यम से हमने यह सीखा कि कैसे आधुनिक NLP तकनीकों का उपयोग करके वास्तविक समस्याओं का समाधान किया जा सकता है।

यह प्रणाली छात्रों, शोधकर्ताओं और व्यावसायिक पेशेवरों के लिए समान रूप से उपयोगी हो सकती है और इसे आगे और अधिक कार्यक्षमता के साथ बढ़ाया जा सकता है।

चित्र 12 : Data Flow Diagram



# 2.3 Text to Speech Converter

## टेक्स्ट-टू-स्पीच कन्वर्टर (Text to Speech Converter)

### 2.3.1 परिचय (Introduction)

आज के डिजिटल युग में सूचना को प्राप्त करने और साझा करने के तरीके तेज़ी से बदल रहे हैं। कृत्रिम बुद्धिमत्ता (Artificial Intelligence) और मशीन लर्निंग (Machine Learning) जैसी तकनीकों ने मानव जीवन को अधिक आसान और प्रभावी बना दिया है। टेक्स्ट-टू-स्पीच (Text-to-Speech) एक ऐसी तकनीक है जो लिखे हुए टेक्स्ट को ऑडियो में बदल देती है, जिससे उपयोगकर्ता उसे सुन सकते हैं।

यह परियोजना एक सरल लेकिन उपयोगी टेक्स्ट-टू-स्पीच वेब एप्लिकेशन पर आधारित है, जिसमें उपयोगकर्ता अपनी इच्छित सामग्री को टाइप कर उसे कंप्यूटर की आवाज़ में सुन सकते हैं या उसे ऑडियो फ़ाइल के रूप में सहेज सकते हैं। इस परियोजना में हमने Python, Flask, और pyttsx3 लाइब्रेरी का उपयोग किया है।

### टेक्स्ट-टू-स्पीच तकनीक का सार (What is TTS Technology?)

**Text-to-Speech (TTS)** एक कंप्यूटर आधारित तकनीक है जो लिखित टेक्स्ट को स्वचालित रूप से कृत्रिम आवाज़ में बदल देती है। इसका मतलब है कि कंप्यूटर या मोबाइल स्क्रीन पर लिखा गया कोई भी टेक्स्ट, एक बोलने वाली आवाज़ में बदला जा सकता है, जिससे उपयोगकर्ता उस जानकारी को सुन सकता है। यह विशेष रूप से उन लोगों के लिए वरदान है जो पढ़ने में असमर्थ हैं, जैसे:

- नेत्रहीन व्यक्ति जिन्हें किसी स्क्रीन पर टेक्स्ट पढ़ना संभव नहीं होता।
- बुजुर्ग, जिन्हें छोटी स्क्रीन पर टेक्स्ट पढ़ना मुश्किल लगता है।
- दृष्टिहीन या डिस्लेक्सिया जैसे शिक्षण विकार से पीड़ित व्यक्ति जो टेक्स्ट समझने में परेशानी महसूस करते हैं।

TTS तकनीक का उपयोग आज कई क्षेत्रों में हो रहा है जैसे कि वर्चुअल असिस्टेंट (जैसे Alexa, Siri), ई-लर्निंग, नेविगेशन सिस्टम, स्मार्टफोन ऐप्स, और हेल्थकेयर में।

### **2.3.2 उद्देश्य (Objective)**

इस परियोजना के मुख्य उद्देश्य निम्नलिखित हैं:

- i. उपयोगकर्ता को एक ऐसा प्लेटफार्म प्रदान करना जो टेक्स्ट को स्पीच में बदल सके।
- ii. नेत्रहीन और पढ़ने में असमर्थ लोगों की मदद करना।
- iii. तकनीकी विकास के साथ जुड़ना और व्यावहारिक परियोजनाओं के माध्यम से कौशल में वृद्धि करना।
- iv. वेब एप्लिकेशन का निर्माण जो आसानी से उपयोग किया जा सके।

### **2.3.3 प्रयुक्त तकनीकें (Technologies Used)**

#### **i. Python**

यह प्रोग्रामिंग भाषा सरल, शक्तिशाली और लोकप्रिय है। इसमें बहुत सारी लाइब्रेरीज़ हैं जो मशीन लर्निंग, वेब डेवेलपमेंट, और ऑडियो प्रोसेसिंग जैसे क्षेत्रों में उपयोगी हैं।

#### **ii. pyttsx3**

pyttsx3 एक ऑफलाइन टेक्स्ट-टू-स्पीच कन्वर्जन लाइब्रेरी है। यह बिना इंटरनेट के काम करती है और उपयोगकर्ता को ऑडियो सुनाने या फ़ाइल में सेव करने की सुविधा देती है।

#### **iii. Flask**

Flask एक हल्का व फ्रेमवर्क है जो Python में वेब एप्लिकेशन बनाने के लिए प्रयोग होता है। इसका प्रयोग हमने बैकएंड के रूप में किया है।

#### **iv. HTML और Bootstrap**

HTML को हमने वेब इंटरफ़ेस डिज़ाइन के लिए उपयोग किया है और Bootstrap से उसे स्टाइलिश व रिस्पॉन्सिव बनाया गया है।

#### **v. render\_template और request:**

- HTML फाइल को सर्वर से ब्राउज़र में भेजने के लिए render\_template का उपयोग किया जाता है।
- request का उपयोग यूज़र इनपुट लेने के लिए किया जाता है।

#### vi. `send_file`:

- जब यूजर ऑडियो फाइल डाउनलोड करना चाहता है, तब यह फाइल को HTTP रिस्पॉन्स के रूप में ब्राउज़र में भेजता है।

### 2.3.4 कार्यविधि (Working Process)

#### i. यूजर इनपुट इंटरफ़ेस:

- एक HTML फॉर्म यूजर को टेक्स्ट इनपुट देने और "Save as audio file" का विकल्प चुनने की सुविधा देता है।

#### ii. रूटिंग (Routing):

- '/' रूट पर यूजर को इनपुट फॉर्म दिखाया जाता है।
- '/speak' रूट पर फॉर्म की जानकारी प्रोसेस होती है और स्पीच जनरेट की जाती है।

#### iii. स्पीच जनरेशन:

- `pyttsx3.init()` से TTS इंजन को आरंभ किया जाता है।
- वॉइस सेटिंग्स (जैसे कि स्पीड, वॉल्यूम, वॉइस) को यूजर फ्रेंडली बनाया गया है।
- यदि यूजर ने "Save" का विकल्प चुना है, तो स्पीच को .mp3 फाइल के रूप में सेव किया जाता है।
- अन्यथा, स्पीच को सीधे ब्राउज़र में चलाया जाता है।

इस परियोजना में दो मुख्य भाग हैं:

- main.py** (कमांड लाइन आधारित एप्लिकेशन)
- app.py** और **index.html** (वेब आधारित एप्लिकेशन)

#### I. **main.py**

यह फाइल एक कमांड लाइन एप्लिकेशन है। इसमें उपयोगकर्ता से टेक्स्ट इनपुट लिया जाता है और पूछा जाता है कि क्या वे उसे फाइल में सेव करना चाहते हैं या केवल सुनना चाहते हैं। यदि वे सेव करना चाहते हैं तो एक ऑडियो फाइल बनाई जाती है।

मुख्य कार्य इस प्रकार होते हैं:

- pyttsx3 के माध्यम से स्पीच इंजन को इनिशियलाइज़ किया जाता है।
- आवाज़ की गति और वॉल्यूम को सेट किया जाता है।
- उपयोगकर्ता से टेक्स्ट और सेव विकल्प लिया जाता है।
- टेक्स्ट को बोलकर सुनाया जाता है या MP3 फॉर्मेट में सेव किया जाता है।

## II. app.py और index.html

यह भाग वेब एप्लिकेशन है। इसमें उपयोगकर्ता एक वेब पेज पर जाकर टेक्स्ट दर्ज करते हैं।

**मुख्य प्रक्रियाएँ:**

- Flask सर्वर HTML पेज को लोड करता है।
- उपयोगकर्ता टेक्स्ट इनपुट करता है और 'Save as audio file' का विकल्प चुन सकता है।
- यदि उपयोगकर्ता सेव करता है, तो pyttsx3 उस टेक्स्ट को MP3 फाइल के रूप में static/ फोल्डर में सेव करता है।
- यदि नहीं करता है, तो वही टेक्स्ट आवाज़ में बोला जाता है।
- अंत में यूजर को संदेश दिया जाता है कि स्पीच सफलतापूर्वक चल चुकी है।

**कोड विश्लेषण**

**main.py में:**

```
python
CopyEdit
engine = pyttsx3.init()
engine.setProperty('rate', 100)
engine.setProperty('volume', 1.0)
engine.setProperty('voice', voices[1].id)
```

- rate: आवाज़ की गति को नियंत्रित करता है।
- volume: ध्वनि की तीव्रता निर्धारित करता है।
- voices: अलग-अलग आवाज़ (पुरुष या महिला) के लिए उपयोग किया जाता है।

**app.py में:**

python

CopyEdit

```
@app.route('/speak', methods=['POST'])
```

- यह फ़ंक्शन वेब फ़ॉर्म से POST अनुरोध को स्वीकार करता है।
- टेक्स्ट को पढ़ता है और pyttsx3 के माध्यम से प्रोसेस करता है।

**index.html में:**

- HTML का प्रयोग करके एक सुंदर यूजर इंटरफ़ेस बनाया गया है।
- उपयोगकर्ता से इनपुट लेने के लिए <textarea> का उपयोग किया गया है।
- एक चेकबॉक्स है जो ऑडियो को सेव करने की अनुमति देता है।

**विशेषताएँ**

- i. **ऑफलाइन कार्यशीलता:** इस परियोजना में pyttsx3 का उपयोग किया गया है, जो इंटरनेट कनेक्शन के बिना भी कार्य करता है।
- ii. **ऑडियो फ़ाइल सेव करना:** उपयोगकर्ता MP3 फ़ॉर्मेट में फाइल सेव कर सकता है, जिसे किसी भी मीडिया प्लेयर से चलाया जा सकता है।
- iii. **सरल इंटरफ़ेस:** HTML और Bootstrap के माध्यम से एक उपयोगकर्ता-अनुकूल इंटरफ़ेस बनाया गया है।
- iv. **ओपन-सोर्स:** सभी लाइब्रेरीज़ ओपन-सोर्स हैं, जिससे यह प्रोजेक्ट बिना किसी लाइसेंस शुल्क के तैयार किया जा सकता है।

**उपयोगिता**

- नेत्रहीन व्यक्तियों के लिए सहायक
- बच्चों को पढ़ाई में मदद करने के लिए
- भाषाई प्रयोगों में
- इंबुक्स को सुनने योग्य बनाने के लिए
- भाषण अभ्यास के लिए

## भविष्य की संभावनाएँ

इस परियोजना को और बेहतर बनाया जा सकता है:

- i. भाषा विकल्प: हिंदी, अंग्रेजी और अन्य भाषाओं का चयन करने की सुविधा दी जा सकती है।
- ii. स्पीच स्पीड स्लाइडर: उपयोगकर्ता स्वयं स्पीच की गति नियंत्रित कर सके।
- iii. ऑफलाइन टेक्स्ट डिटेक्शन: उपयोगकर्ता किसी वेबसाइट या PDF से टेक्स्ट ले सकें।
- iv. क्लाउड सेविंग: उपयोगकर्ता अपनी ऑडियो क्लाउड में सेव कर सकें। प्रोजेक्ट में उपयोग की गई प्रमुख लाइब्रेरीज़ का विवरण

### 1. pyttsx3

- **उद्देश्य:** यह Python की एक Text-to-Speech (TTS) लाइब्रेरी है जो ऑफलाइन वॉइस जनरेशन को सपोर्ट करती है।
- **मुख्य विशेषताएँ:**
  - इंटरनेट कनेक्शन की आवश्यकता नहीं।
  - आवाज़ की गति (rate), वॉल्यूम (volume), और आवाज़ के प्रकार (voice – male/female) को नियंत्रित किया जा सकता है।
  - .say() से टेक्स्ट बोलता है, और .save\_to\_file() से MP3 या WAV फ़ाइल में सेव करता है।

### 2. Flask

- **उद्देश्य:** यह एक लाइटवेट वेब फ्रेमवर्क है, जो Python में वेब एप्लिकेशन बनाने के लिए प्रयोग होता है।
- **मुख्य विशेषताएँ:**
  - आसान रूटिंग (URL सेट करना)।
  - HTML टेम्पलेट सपोर्ट करता है (जैसे Jinja2)।
  - POST और GET जैसी HTTP रिक्वेस्ट हैंडल करता है।
  - तेजी से प्रोटोटाइप बनाने के लिए उपयुक्त।

### 3. HTML + Bootstrap

- **उद्देश्य:** यूजर इंटरफ़ेस (फ्रंटेंड) डिजाइन करने के लिए उपयोग किया गया है।
- **मुख्य विशेषताएँ:**
  - HTML फॉर्म से टेक्स्ट इनपुट।
  - Bootstrap से रेस्पॉन्सिव, सुंदर और मोबाइल फ्रैंडली डिज़ाइन।

- बटन, चेकबॉक्स, फॉर्म आदि को सुंदरता प्रदान करता है।

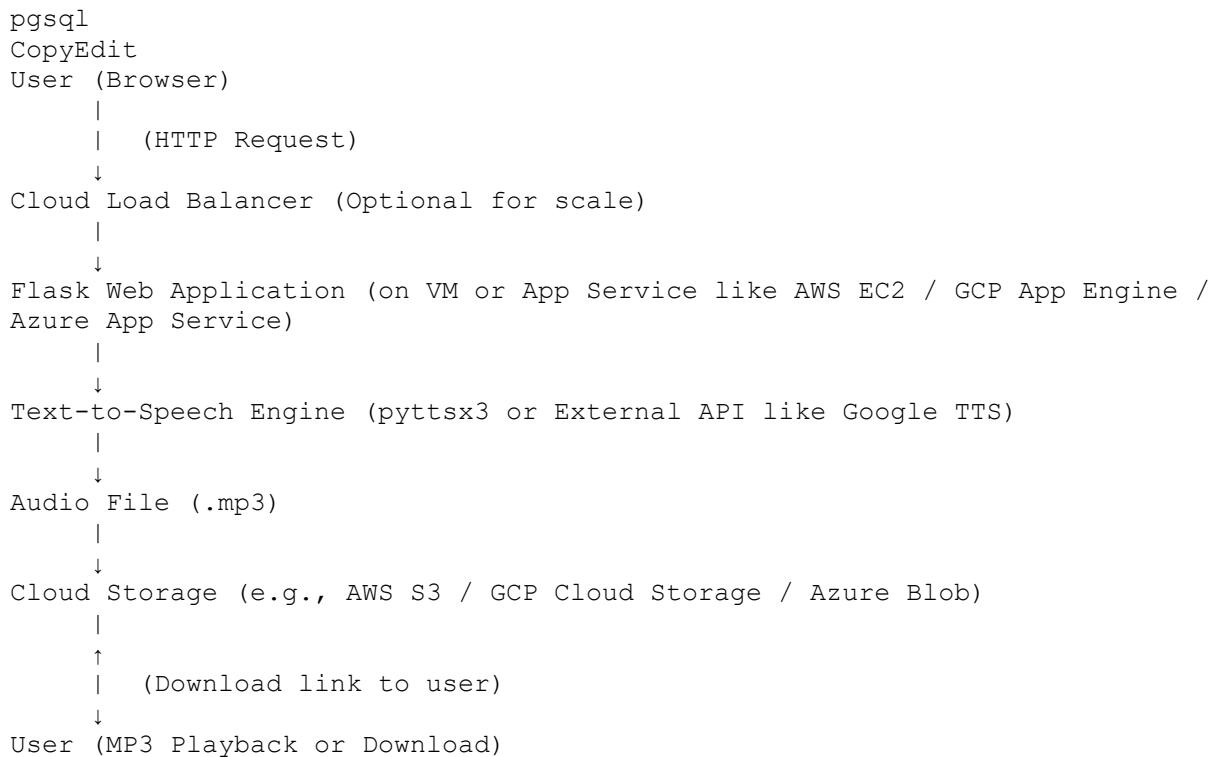
### फोल्डर संरचना (Directory Structure):

```

vbnet
CopyEdit
text-to-speech-project/
|
|   main.py
|   app.py
|   templates/
|       index.html
|   static/
|       output.mp3 (auto generated)

```

### 2.3.5 क्लाउड आर्किटेक्चर डिज़ाइन (Cloud Architecture Design)



## आउटपुट (Output):-

 **Text-to-Speech Converter**

Enter your text:

Main Hu Jiyān Mera

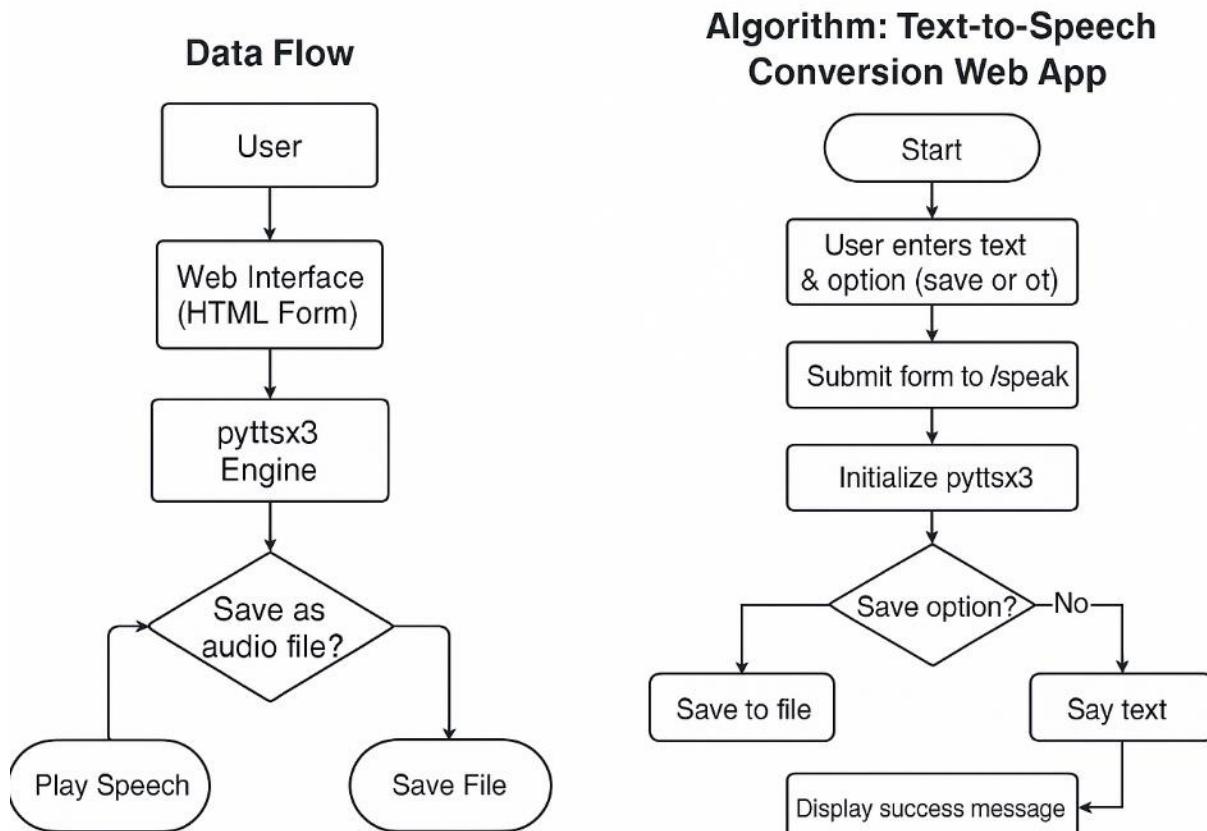
Save as audio file

 Convert to Speech

✓ Speech played successfully!

चित्र 13 : Output

### 2.3.6 डेटा प्रवाह आरेख (Data Flow Diagram)



चित्र 14 : Data Flow Diagram & Algorithm

## एल्गोरिदम (Algorithm) : Text-to-Speech Conversion Web App

1. Start
2. उपयोगकर्ता वेब पेज पर टेक्स्ट इनपुट करता है।
3. उपयोगकर्ता चेकबॉक्स द्वारा तय करता है कि फाइल सेव करनी है या नहीं।
4. HTML फॉर्म /speak रूट पर POST अनुरोध भेजता है।
5. Flask एप्लिकेशन POST डेटा को प्राप्त करता है:
  - o text और save\_option
6. pyttsx3 इंजन इनिशियलाइज़ किया जाता है।
7. इंजन की सेटिंग्स सेट की जाती हैं:
  - o गति (rate) = 100
  - o वॉल्यूम = 1.0
  - o आवाज़ (male/female) चयन
8. If save\_option is True:
  - o save\_to\_file(text, filename) कॉल होता है।
  - o engine.runAndWait() से फाइल सेव होती है।
  - o यूजर को MP3 फाइल डाउनलोड के लिए भेजी जाती है।
9. Else:
  - o engine.say(text) और engine.runAndWait() से टेक्स्ट बोला जाता है।
  - o एक HTML रिस्पॉन्स द्वारा मैसेज दिखाया जाता है: " Speech played successfully!"
10. End

### 2.3.7 निष्कर्ष (Conclusion)

इस प्रोजेक्ट "Text-to-Speech Converter" के माध्यम से हमने यह सिद्ध किया कि टेक्नोलॉजी का उपयोग केवल सुविधाओं तक सीमित नहीं है, बल्कि यह समाज के हर वर्ग तक पहुँच बनाने और उन्हें सक्षम करने का एक सशक्त माध्यम भी बन सकता है। इस प्रणाली के द्वारा उपयोगकर्ता सरलता से अपने टेक्स्ट को ऑडियो में बदल सकते हैं, और चाहे तो उस ऑडियो को सेव भी कर सकते हैं। यह कार्य बिना किसी जटिल प्रक्रिया के एक यूजर-फ्रेंडली इंटरफ़ेस के माध्यम से किया जा सकता है, जो हर आयु वर्ग के लोगों के लिए सुलभ है।

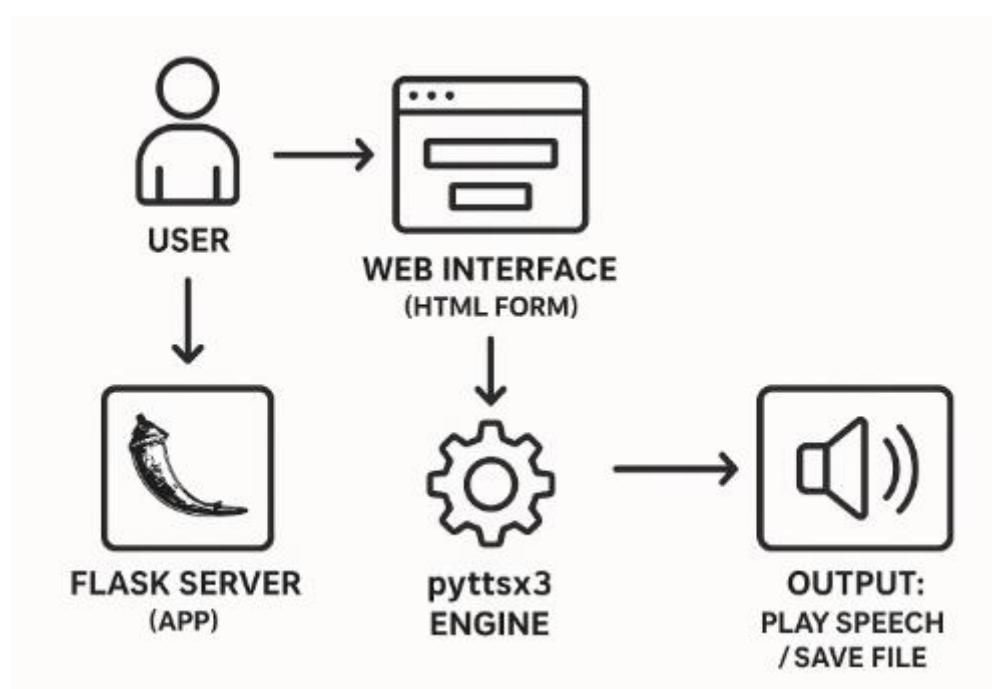
इस परियोजना में हमने "Python, Flask," और "pyttsx3" जैसी मजबूत तकनीकों और लाइब्रेरीज़ का उपयोग किया है। "Python" के सरलीकृत सिटैक्स और "Flask" के हल्के व तेज़ वेब फ्रेमवर्क की मदद से इस प्रोजेक्ट को तेजी से डेवेलप और डिप्लॉय करना संभव हुआ। वहीं "pyttsx3" जैसी लाइब्रेरी ने हमें "Offline Text-to-Speech" कन्वर्ज़न की सुविधा दी, जिससे यह सिस्टम इंटरनेट की निर्भरता के बिना भी कार्य करने में सक्षम है।

इस एप्लिकेशन का प्रयोग विशेष रूप से नेत्रहीन, बुजुर्ग, शिक्षण विकलांगता से पीड़ित, या पढ़ाई में सहायक साधन की तरह उपयोग करने वाले विद्यार्थियों के लिए वरदान साबित हो सकता है। इसके अलावा, इसे शिक्षा, हेल्थकेयर, स्मार्ट डिवाइसेज़ और सरकारी सेवाओं जैसे विभिन्न क्षेत्रों में उपयोग किया जा सकता है।

एक और प्रमुख विशेषता इस परियोजना की यह है कि यह किफायती (low cost) और ऑफलाइन-सक्षम (offline-capable) है, जिससे यह विकासशील और ग्रामीण क्षेत्रों के लिए भी उपयोगी बनती है जहाँ इंटरनेट की सुविधा सीमित होती है।

इस परियोजना से यह स्पष्ट होता है कि कैसे एक साधारण विचार—"टेक्स्ट को आवाज़ में बदलना"—एक प्रभावशाली और समाजोपयोगी समाधान बन सकता है। भविष्य में इसमें और सुधार किए जा सकते हैं जैसे कि मल्टी लैंग्वेज सोर्ट, वॉयस कस्टमाइजेशन, रीयल-टाइम स्पीच ट्रांसलेशन, आदि।

अंततः, यह प्रोजेक्ट न केवल एक तकनीकी प्रयोग है, बल्कि एक सशक्त सामाजिक उपकरण है, जो डिजिटल समावेशिता (digital inclusion) की दिशा में एक ठोस कदम है। यह हमें यह भी सिखाता है कि जब टेक्नोलॉजी को सही दिशा में उपयोग किया जाए, तो वह सिर्फ जीवन को आसान ही नहीं, बल्कि समानता से भरपूर भी बना सकती है।



चित्र 15 : Processing of Pyttsx3 Engine

# 2.4 Facial Emotion Detection

## Facial Emotion Detection

(चेहरे की भावनाओं की पहचान)

### 2.4.1 परिचय (Introduction)

21वीं सदी में मानव सभ्यता ने तकनीकी उन्नति की उस ऊँचाई को छू लिया है, जहाँ मशीनें अब केवल आदेशों का पालन करने वाले यंत्र नहीं रह गई हैं, बल्कि वे सोचने, समझने और विश्लेषण करने में भी सक्षम हो गई हैं। कृत्रिम बुद्धिमत्ता (Artificial Intelligence - AI) इसी दिशा में एक क्रांतिकारी खोज है, जिसने दुनिया भर के उद्योगों, सेवाओं और शिक्षा प्रणालियों में अभूतपूर्व बदलाव लाए हैं।

AI की ही एक उपशाखा है कंप्यूटर विज़न (Computer Vision), जो कंप्यूटर को इंसानी आँखों की तरह "देखने" और "समझने" की शक्ति प्रदान करती है। पहले के समय में कंप्यूटर केवल संख्याएँ और टेक्स्ट ही समझता था, लेकिन अब वह तस्वीरों और वीडियो का विश्लेषण कर सकता है। कंप्यूटर विज़न ने सुरक्षा प्रणाली, चिकित्सा, रोबोटिक्स, वाहन निर्माण, मीडिया और इंटरैक्टिव टेक्नोलॉजी में क्रांतिकारी परिवर्तन लाया है।

इसी क्षेत्र का एक अत्यंत रोचक और संवेदनशील उपयोग है – चेहरे की भावनाओं की पहचान (Facial Emotion Recognition)। यह तकनीक इंसानों के चेहरे के हावभाव (facial expressions) को देखकर उनकी मानसिक स्थिति को पहचानने का प्रयास करती है। जैसे कि कोई व्यक्ति मुस्कुरा रहा हो तो वह प्रसन्न है, अगर चेहरा सिकुड़ा हो तो वह क्रोधित है, आदि। एक इंसान स्वाभाविक रूप से यह सब पहचान लेता है, लेकिन कंप्यूटर को इसके लिए मशीन लर्निंग और डीप लर्निंग एल्गोरिदम की ज़रूरत होती है।

प्रस्तुत प्रोजेक्ट “expression.py” इसी विचार पर आधारित है। इसमें हमने एक ऐसा सिस्टम विकसित किया है जो किसी भी व्यक्ति की तस्वीर को इनपुट के रूप में लेकर उस तस्वीर में मौजूद चेहरे से प्रमुख भावना को पहचानने की क्षमता रखता है। इस उद्देश्य को पूरा करने के लिए हमने Python भाषा का उपयोग किया है और विशेष रूप से DeepFace नामक ओपन-सोर्स Deep Learning लाइब्रेरी को अपनाया है। DeepFace पहले से प्रशिक्षित (pre-trained) मॉडल्स के आधार पर काम करती है और यह अलग-अलग भावनाओं की सटीक पहचान करने में सक्षम है।

इस प्रोजेक्ट के माध्यम से यह सिद्ध किया जा सकता है कि अब कंप्यूटर केवल गणना करने की मशीन नहीं, बल्कि संवेदनशील और बौद्धिक निर्णय लेने वाली इकाई बन सकती है। यह प्रणाली न केवल तकनीकी दृष्टिकोण से अत्यंत उपयोगी है, बल्कि इसका सामाजिक उपयोग भी

अत्यधिक है – जैसे मानसिक स्वास्थ्य का आकलन, शिक्षा में छात्रों की मनःस्थिति जानना, ग्राहकों की प्रतिक्रिया समझना, आदि।

इस प्रकार, “expression.py” प्रोजेक्ट एक ऐसा प्रयास है जो AI को मानवीय भावनाओं के करीब लाता है और एक ऐसा भविष्य दर्शाता है जहाँ मशीनें केवल हमारी मदद ही नहीं करेंगी, बल्कि हमारे साथ “अनुभव” भी साझा कर सकेंगी।

#### 2.4.2 उद्देश्य (Objective)

इस प्रोजेक्ट का मूल उद्देश्य यह है कि कंप्यूटर को इस स्तर तक प्रशिक्षित किया जाए कि वह इंसानों की तरह चेहरे की भावनाओं को पहचान सके। अर्थात् जब कोई व्यक्ति खुश, दुखी, नाराज़ या हैरान हो, तो कंप्यूटर केवल उसके चेहरे को देखकर उस भावना का अनुमान लगा सके और उसे स्पष्ट रूप से बता सके।

भावनाएँ (Emotions) किसी भी इंसान की मानसिक स्थिति को दर्शाने का सबसे सटीक तरीका होती हैं। हम अपने चेहरे के हावभाव से अनजाने में ही बहुत कुछ ज़ाहिर कर देते हैं – जैसे मुस्कुराना, भौंहें चढ़ाना, मुँह सिकोड़ना आदि। इन हावभावों को कंप्यूटर द्वारा पढ़ना और उनकी व्याख्या करना एक कठिन कार्य है, लेकिन Deep Learning की मदद से अब यह संभव हो चुका है।

इस परियोजना (expression.py) के अंतर्गत, हम Python की सहायता से DeepFace लाइब्रेरी को उपयोग में लाते हैं। यह लाइब्रेरी पहले से प्रशिक्षित गहरे न्यूरल नेटवर्क (Deep Neural Networks) का उपयोग करती है, जो मानव चेहरे की विविध विशेषताओं को पहचान सकती है। DeepFace द्वारा दिए गए मॉड्यूल इमेज में मौजूद चेहरे को ढूँढते हैं, फिर उस चेहरे का विश्लेषण करके यह निर्धारित करते हैं कि उसमें प्रकट होने वाली प्रमुख भावना कौन-सी है।

इस तकनीक का वास्तविक जीवन में उपयोग अनेक क्षेत्रों में किया जा सकता है:

- i. **मानसिक स्वास्थ्य मूल्यांकन:** किसी व्यक्ति के चेहरे के भावों से यह अंदाज़ा लगाया जा सकता है कि वह तनाव में है या डिप्रेशन का शिकार है।
- ii. **शैक्षणिक क्षेत्र:** शिक्षक ऑनलाइन कक्षा के दौरान छात्रों के चेहरे से समझ सकते हैं कि वे ध्यान दे रहे हैं या नहीं।
- iii. **सुरक्षा प्रणाली:** संदिग्ध गतिविधियों की पहचान के लिए भावनात्मक विश्लेषण एक महत्वपूर्ण भूमिका निभा सकता है।
- iv. **ग्राहक सेवा:** ग्राहक की संतुष्टि का मूल्यांकन उसके चेहरे के हावभाव से किया जा सकता है।

इसलिए, इस प्रोजेक्ट का उद्देश्य केवल तकनीकी प्रदर्शन तक सीमित नहीं है, बल्कि यह मानवीय व्यवहार को मशीनों द्वारा समझे जाने की दिशा में एक उपयोगी और व्यवहारिक कदम है।

इस प्रोजेक्ट के माध्यम से हम यह भी सीखते हैं कि किसी समस्या को किस तरह टुकड़ों में बाँटकर सुलझाया जा सकता है – जैसे इमेज इनपुट लेना, इमेज को पढ़ना, इमेज को प्रदर्शित करना, DeepFace से एनालिसिस करना और परिणाम को प्रस्तुत करना।

अंततः, यह उद्देश्य केवल कोडिंग तक सीमित न होकर, मानवता और मशीन के बीच एक बेहतर संवाद स्थापित करने का प्रयास है – जिसमें मशीनें न केवल हमारी भाषा, बल्कि हमारे हावभाव और भावनाएँ भी समझ सकें।

#### 2.4.3 कार्यप्रणाली (Working Mechanism)

इस प्रोजेक्ट की संपूर्ण प्रक्रिया को पाँच मुख्य चरणों में विभाजित किया जा सकता है:

##### i. इमेज संग्रहण (Image Collection)

सबसे पहले एक फोल्डर में उन सभी तस्वीरों को रखा जाता है जिनका विश्लेषण करना होता है। ये तस्वीरें विभिन्न भावनाओं को दर्शाने वाले चेहरों की होती हैं।

##### ii. इमेज को पहचानना और पढ़ना

प्रोग्राम उस फोल्डर से एक-एक करके सभी इमेज फाइलों को पहचानता है और उन्हें पढ़ता है। इमेज की पहचान उसके फाइल एक्सटेंशन (जैसे .jpg, .png आदि) के आधार पर की जाती है।

##### iii. इमेज विज़ुअलाइज़ेशन

हर इमेज को पढ़ने के बाद उसे स्क्रीन पर दिखाया जाता है ताकि उपयोगकर्ता यह देख सके कि किस इमेज का विश्लेषण किया जा रहा है। OpenCV द्वारा इमेज को पढ़ा जाता है और Matplotlib द्वारा प्रदर्शित किया जाता है।

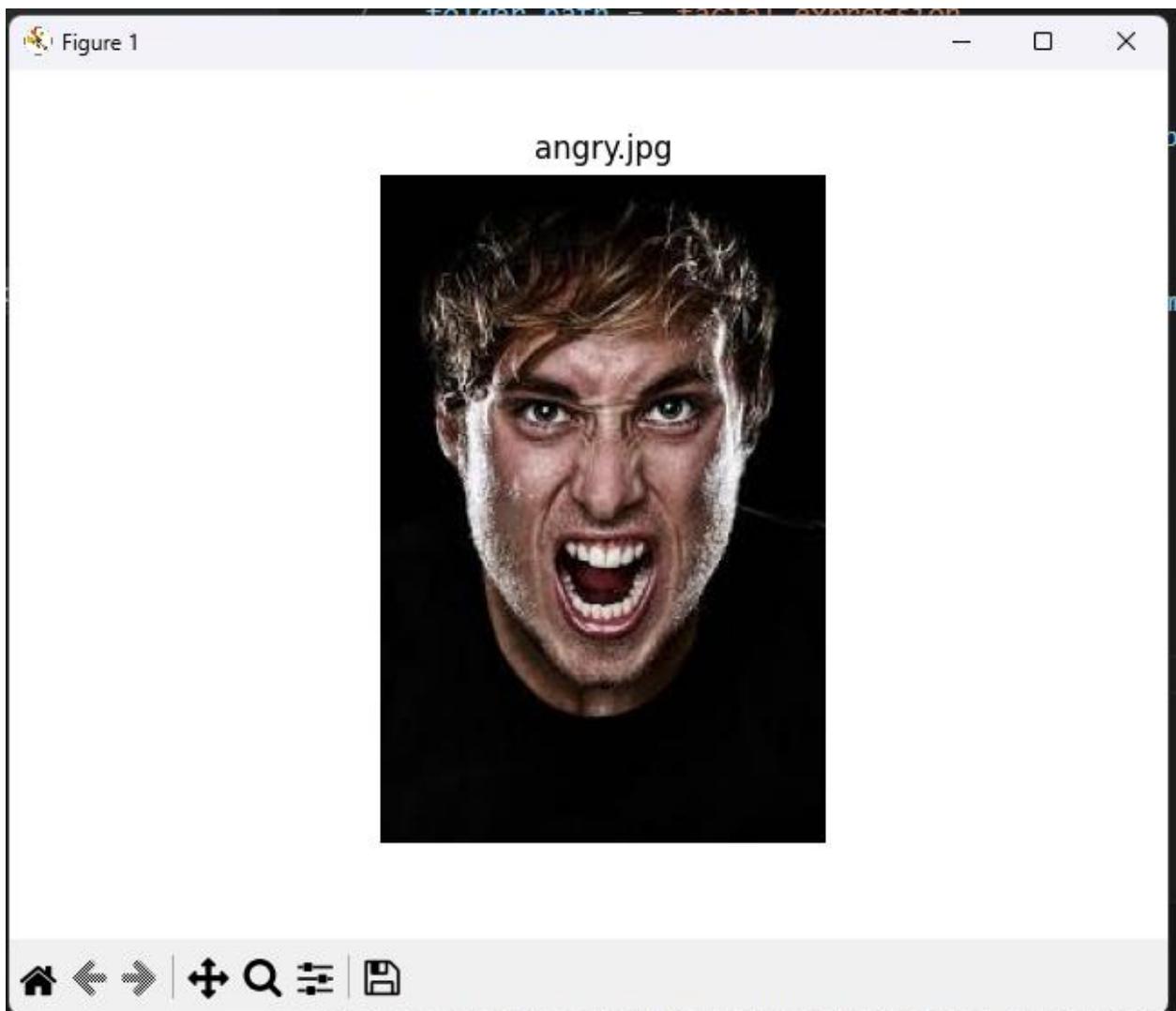
##### iv. इमेज का विश्लेषण (Emotion Analysis)

यह चरण सबसे महत्वपूर्ण है। इसमें DeepFace लाइब्रेरी का प्रयोग किया जाता है, जो कि एक शक्तिशाली फेस एनालिसिस टूल है। DeepFace मॉडल उस इमेज में मौजूद चेहरे को डिटेक्ट करता है और फिर उसकी भावनाओं का विश्लेषण करता है।

इस विश्लेषण के दौरान यह निम्नलिखित प्रमुख भावनाओं में से किसी एक को "Dominant Emotion" के रूप में घोषित करता है:

- खुश (happy)
- गुस्सा (angry)

- दुखी (sad)
- तटस्थ (neutral)
- आश्चर्य (surprise)
- डर (fear)
- घुणा (disgust)



#### 2.4.4 आउटपुट (Output)

```
Dominant Emotion: neutral
Processing: face14.jpg
Dominant Emotion: happy
Processing: face24.jpg
Dominant Emotion: happy

Processing: face7.jpg
Could not process face7.jpg. Error: Face could not be detected in numpy array. Please confirm that the picture is a face photo or consider to set enforce_detection param to False.
PS C:\Users\dipan\OneDrive\Desktop\python Projects\facial_expression> 
```

चित्र 16 : Output

इस प्रोजेक्ट में आउटपुट बहुत ही सहज और स्पष्ट रूप में उपयोगकर्ता को प्रदान किया जाता है। जब कोई इमेज `expression.py` प्रोग्राम में प्रोसेस की जाती है, तो सबसे पहले उस इमेज को स्क्रीन पर प्रदर्शित किया जाता है, जिससे उपयोगकर्ता को यह देखने को मिले कि कौन-सी तस्वीर वर्तमान में विश्लेषण के लिए उपयोग हो रही है।

इसके बाद, DeepFace लाइब्रेरी उस इमेज में मौजूद चेहरे का विश्लेषण करती है और टर्मिनल (या कमांड लाइन) पर एक संक्षिप्त परिणाम दिखाती है। इस परिणाम में दो मुख्य बातें होती हैं:

1. इमेज का नाम - जिससे पता चलता है कि कौन-सी तस्वीर विश्लेषण की जा रही थी।
2. Dominant Emotion - यानी उस चेहरे में जो सबसे प्रमुख भावना पाई गई, जैसे कि "happy", "sad", "angry", "surprise", आदि।

यदि किसी तस्वीर में चेहरा स्पष्ट रूप से नहीं है, या इमेज का फॉर्मेट सही नहीं है, या इमेज में तकनीकी समस्या है, तो प्रोग्राम एक एरर मैसेज (Error Message) देता है। यह एरर उपयोगकर्ता को बताता है कि उस इमेज को प्रोसेस नहीं किया जा सका।

इस प्रकार, आउटपुट उपयोगकर्ता को प्रत्येक इमेज का विश्लेषणात्मक सार देता है और त्रुटियों को भी स्पष्ट रूप से दर्शाता है।

## Execution Start

plaintext

CopyEdit

PS C:\Users\dipan\OneDrive\Desktop\python Projects\facial\_expression> &  
यह कमांड PowerShell में एक नई पंक्ति से स्क्रिप्ट रन करने के लिए है। इसका विशेष कार्य नहीं है, बस इंटरैक्टिव कमांड लाइन को संकेत देता है।

## Script Execution

plaintext

CopyEdit

PS C:\Users\dipan\OneDrive\Desktop\python Projects\facial\_expression> &  
C:/Users/dipan/AppData/Local/Programs/Python/Python310/python.exe  
"c:/Users/dipan/OneDrive/Desktop/python  
Projects/facial\_expression/analyze\_expressions.py"

यह कमांड `analyze_expressions.py` नामक Python स्क्रिप्ट को आपके सिस्टम में इंस्टॉल किए गए Python 3.10 इंटरप्रेटर से रन कर रहा है। स्क्रिप्ट `facial_expression` प्रोजेक्ट फोल्डर में मौजूद है।

## TensorFlow Initialization Messages

plaintext

CopyEdit

2025-06-26 17:18:03.058143: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on...

- यह “TensorFlow” द्वारा दिखाए जाने वाले लॉग मैसेज हैं। “TensorFlow DeepFace” के backend में यूज़ हो रहा है।
- “oneDNN” ऑप्टिमाइज़ेशन सक्षम है जिससे परफॉर्मेंस बेहतर हो सकती है, लेकिन फ्लोटिंग पॉइंट राउंडिंग एरर के कारण थोड़े अलग परिणाम भी आ सकते हैं।

## Deprecation Warning

plaintext

CopyEdit

WARNING:tensorflow:From ... The name tf.losses.sparse\_softmax\_cross\_entropy is deprecated...

- TensorFlow यह चेतावनी देता है कि एक पुराना फंक्शन प्रयोग हो रहा है जो अब deprecated है। यह स्क्रिप्ट के काम पर सीधा असर नहीं डालता, सिर्फ भविष्य के लिए सूचना है।

## Image Processing Begins

plaintext

CopyEdit

Processing: angry.jpg

- अब angry.jpg नामक इमेज को स्क्रिप्ट द्वारा प्रोसेस किया जा रहा है।
- यह इमेज cv2 और matplotlib से पढ़ी और प्रदर्शित की गई होगी।

plaintext

CopyEdit

2025-06-26 17:18:30.158954: I tensorflow/core/platform/cpu\_feature\_guard.cc:210]

...

- TensorFlow बताता है कि CPU-निर्भर ऑप्टिमाइज़ेशन एकिटवेटेड हैं, जिससे यह स्क्रिप्ट तेज़ी से रन हो सकती है।

plaintext

CopyEdit

Dominant Emotion: angry

- DeepFace मॉडल ने यह निर्धारित किया कि angry.jpg में व्यक्ति का प्रमुख भाव "गुस्सा (angry)" है।

## अन्य इमेज प्रोसेसिंग

plaintext

CopyEdit

Processing: cry.jpg

Dominant Emotion: sad

Processing: face1.jpg

Dominant Emotion: neutral

Processing: face14.jpg

Dominant Emotion: happy

Processing: face24.jpg

Dominant Emotion: happy

- यह क्रमशः हर इमेज को प्रोसेस करता है:
  - cry.jpg: भाव "तुखी (sad)"
  - face1.jpg: भाव "तटस्थ (neutral)"
  - face14.jpg और face24.jpg: भाव "खुश (happy)"

यह दर्शाता है कि DeepFace मॉडल ने सफलतापूर्वक चेहरे की भावनाएं पहचानीं।

## Error in Detection

plaintext

CopyEdit

Processing: face7.jpg

Could not process face7.jpg. Error: Face could not be detected in numpy array...

- इस इमेज में DeepFace को कोई चेहरा नहीं मिला।
- संभावित कारण:
  - इमेज में व्यक्ति का चेहरा स्पष्ट नहीं है।
  - इमेज धुंधली या बहुत छोटी है।

- कैमरा एंगल या रोशनी खराब है।

समाधान: `enforce_detection=False` सेट किया जा सकता है ताकि मॉडल बिना चेहरे की पुष्टि किए भी प्रोसेस करने की कोशिश करे।

### Execution Complete

plaintext

CopyEdit

PS C:\Users\dipan\OneDrive\Desktop\python Projects\facial\_expression>

- स्क्रिप्ट समाप्त हो चुकी है और अब आप अगले कमांड दे सकते हैं।

### सारांश (Summary)

इमेज नाम	प्रमुख भावना (Dominant Emotion)	स्थिति
angry.jpg	angry	सफल
cry.jpg	sad	सफल
face1.jpg	neutral	सफल
face14.jpg	happy	सफल
face24.jpg	happy	सफल
face7.jpg	X Not Detected	चेहरा नहीं मिला

तालिका 9: (चेहरे की भावनाओं की पहचान परियोजना सारांश (Facial Emotion Detection Project Summary)

### 2.4.5 प्रयोग की गई तकनीकें (Technologies Used)

इस प्रोजेक्ट में निम्नलिखित तकनीकी टूल्स और लाइब्रेरी का उपयोग किया गया है:

तकनीक/लाइब्रेरी	उद्देश्य
Python	प्रोजेक्ट की मुख्य प्रोग्रामिंग भाषा
OpenCV (cv2)	इमेज को पढ़ने और प्रोसेस करने के लिए
Matplotlib	इमेज को स्क्रीन पर प्रदर्शित करने के लिए

<b>DeepFace</b>	चेहरे की भावनाओं का विश्लेषण करने के लिए
<b>OS Module</b>	सिस्टम फाइलों को एक्सेस और पढ़ने के लिए
<p>Python – प्रोजेक्ट की मुख्य प्रोग्रामिंग भाषा</p> <ul style="list-style-type: none"> <li>यह प्रोजेक्ट Python भाषा में लिखा गया है, जो मशीन लर्निंग, इमेज प्रोसेसिंग और डेटा साइंस जैसे क्षेत्रों में सबसे लोकप्रिय और उपयोगी भाषा मानी जाती है।</li> <li>Python का स्टैटेक्स सरल, पठनीय और उपयोग में आसान होता है, जिससे कोड को विकसित करना तेज़ और सुविधाजनक हो जाता है।</li> </ul> <p>OpenCV (cv2) – इमेज को पढ़ने और प्रोसेस करने के लिए</p> <ul style="list-style-type: none"> <li>OpenCV एक ओपन-सोर्स कंप्यूटर विज़न लाइब्रेरी है जिसका उपयोग इमेज को पढ़ने, प्रोसेस करने और उनमें से चेहरों या अन्य ऑब्जेक्ट्स को पहचानने के लिए किया जाता है।</li> <li>इस प्रोजेक्ट में cv2.imread() का प्रयोग कर इमेज को फोल्डर से पढ़ा गया है।</li> <li>OpenCV RGB इमेज को BGR फॉर्मेट में पढ़ता है, इसलिए बाद में रंगों को सही दिखाने के लिए कन्वर्ज़न किया गया।</li> </ul> <p>Matplotlib – इमेज को स्क्रीन पर प्रदर्शित करने के लिए</p> <ul style="list-style-type: none"> <li>यह एक डेटा विज़ुअलाइज़ेशन लाइब्रेरी है जो ग्राफ, चार्ट और इमेज दिखाने के लिए इस्तेमाल होती है।</li> <li>matplotlib.pyplot.imshow() की मदद से इमेज को स्क्रीन पर प्रदर्शित किया जाता है।</li> <li>इसमें इमेज के नीचे टाइटल, एक्सिस और लेआउट को नियंत्रित किया जा सकता है।</li> </ul> <p>DeepFace – चेहरे की भावनाओं का विश्लेषण करने के लिए</p> <ul style="list-style-type: none"> <li>DeepFace एक प्री-ट्रेन्ड (pre-trained) डीप लर्निंग लाइब्रेरी है, जो चेहरे की पहचान, उम्र, लिंग और भावनाओं (emotions) का विश्लेषण कर सकती है।</li> </ul>	

- DeepFace.analyze() फंक्शन को इमेज और एक्शन (जैसे 'emotion') देकर रन किया जाता है।
- यह मॉडल यह बताता है कि उस व्यक्ति के चेहरे से कौन-सी भावना (जैसे खुशी, दुःख, गुस्सा आदि) प्रमुख रूप से झलक रही है।

**OS Module –** सिस्टम फाइलों को एक्सेस और पढ़ने के लिए

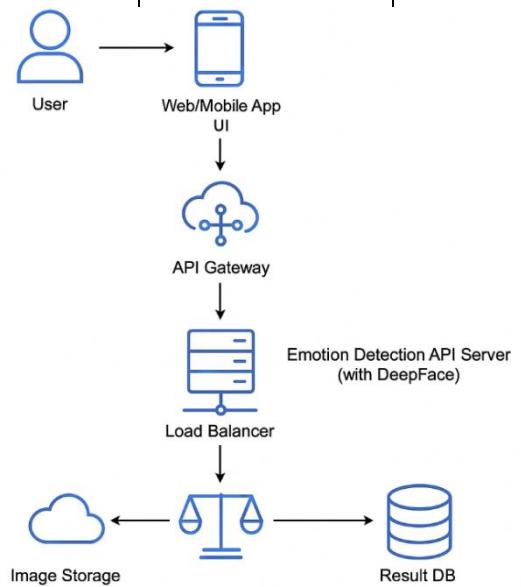
- os मॉड्यूल Python का एक बिल्ट-इन पैकेज है, जिसका उपयोग फाइल्स और फोल्डर्स से जुड़ी गतिविधियों के लिए होता है।
- इस प्रोजेक्ट में os.listdir() के ज़रिए दिए गए फोल्डर के अंदर मौजूद सभी इमेज फाइलों की सूची प्राप्त की जाती है।
- os.path.join() की मदद से हर इमेज की पूर्ण पाथ बनाई जाती है जिससे उसे पढ़ना आसान हो सके।

तालिका 10 : चेहरे की भावनाओं की पहचान परियोजना तकनीकें (Facial Emotion Detection Project Technologies)

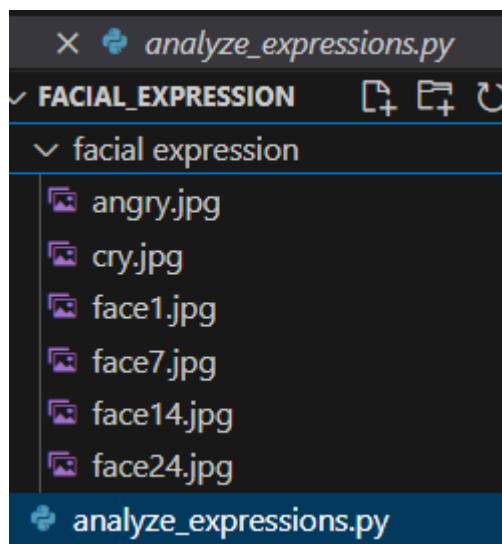
## 2.4.6 रिपोर्टिंग और विश्लेषण (Reporting and Analytics)

### i) Architecture Diagram (Textual Format)

```
css
CopyEdit
[User]
  ↓
[Web/App UI]
  ↓
[API Gateway]
  ↓
[Load Balancer]
  ↓
[Emotion Detection API Server (with DeepFace)]
  ↓
[Image Storage]   [Result DB]
  ↓
[Response to User]
```

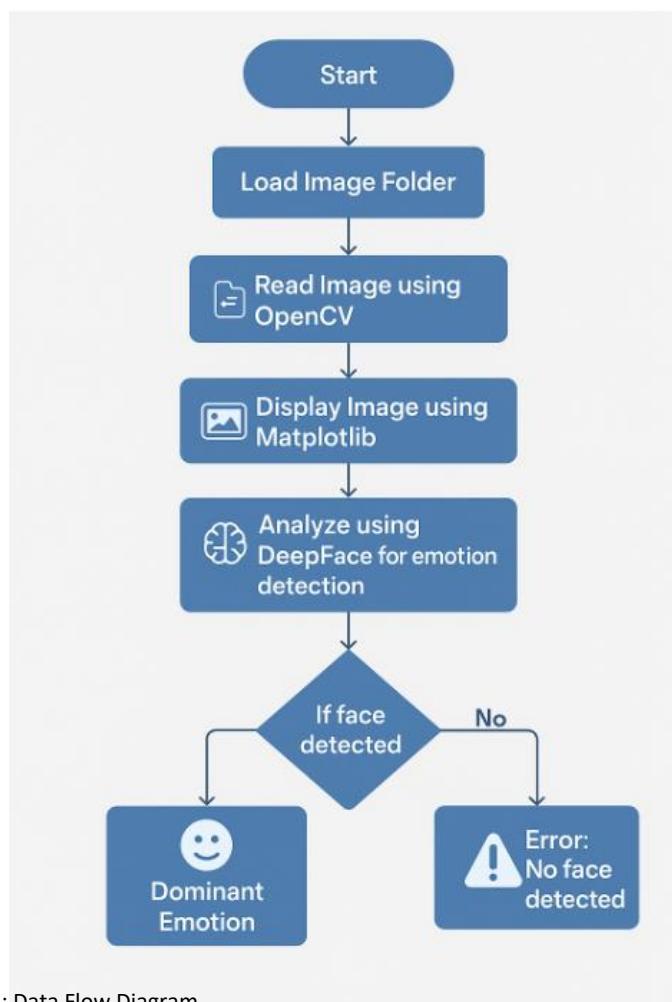


### ii) फाइल संरचना (File Structure)



चित्र 17 : Architecture Diagram & File Structure

### iii) डेटा प्रवाह आरेख (Data Flow Diagram)



चित्र 18 : Data Flow Diagram

### एल्गोरिदम (Algorithm)

: चेहरे से भावना विश्लेषण (Emotion Detection using Image)

इनपुट:

एक फोल्डर जिसमें चेहरे वाली इमेज फाइलें (.jpg, .png आदि) मौजूद हैं।

आउटपुट:

हर इमेज के लिए पहचानी गई प्रमुख भावना (Dominant Emotion) या त्रुटि संदेश (Error Message)

कदम दर कदम प्रक्रिया (Steps):

- i. प्रारंभ करें (Start)
- ii. फोल्डर का पथ सेट करें

- वह फोल्डर चुनें जिसमें इमेज फाइलें संग्रहीत हैं।
- iii. सभी इमेज फाइलों की सूची प्राप्त करें
  - सिर्फ़ .jpg, .jpeg, और .png एक्सटेंशन वाली फाइलों को शामिल करें।
- iv. हर इमेज के लिए निम्नलिखित कार्य करें:
  - a. इमेज पढ़ें (Read Image)
    - OpenCV (cv2.imread()) का उपयोग कर इमेज को पढ़ें।
  - b. इमेज प्रदर्शित करें (Display Image)
    - Matplotlib (plt.imshow()) के माध्यम से इमेज स्क्रीन पर दिखाएं।
  - c. DeepFace से विश्लेषण करें (Analyze Emotion)
    - DeepFace.analyze() को actions=['emotion'] के साथ उपयोग करें।
  - d. चेहरे की उपस्थिति की जाँच करें (Check Face Detection)
    - यदि चेहरा डिटेक्ट हो गया:
      - प्रमुख भावना (dominant\_emotion) को आउटपुट करें।
    - अन्यथा:
      - एक त्रुटि संदेश दिखाएं: "Face not detected".
- v. अगली इमेज पर जाएं
  - लूप के माध्यम से सभी इमेज पर प्रक्रिया दोहराएं।
- vi. समाप्त करें (End)

## 2.4.7 निष्कर्ष (Conclusion)

वर्तमान युग में जहाँ कृत्रिम बुद्धिमत्ता (Artificial Intelligence) निरंतर मानव जीवन को प्रभावित कर रही है, वहाँ चेहरे की भावनाओं की पहचान (Facial Emotion Recognition) एक अत्यंत महत्वपूर्ण और प्रासंगिक तकनीक बन चुकी है। प्रस्तुत प्रोजेक्ट "expression.py" इसी दिशा में एक सार्थक प्रयास है, जिसमें कंप्यूटर विज़न और डीप लर्निंग का उपयोग करते हुए मशीन को इतना सक्षम बनाया गया है कि वह किसी भी व्यक्ति की तस्वीर में उसकी प्रमुख भावना को पहचान सके।

इस प्रोजेक्ट में Python भाषा का उपयोग करके DeepFace जैसी शक्तिशाली ओपन-सोर्स लाइब्रेरी का सहारा लिया गया है, जो पहले से प्रशिक्षित मॉडल्स के आधार पर कार्य करती है। DeepFace न केवल चेहरे को पहचानने में सक्षम है, बल्कि उसकी अभिव्यक्ति (Expression) के आधार पर यह भी बता सकती है कि व्यक्ति खुश है, दुखी है, नाराज़ है या आश्चर्यचकित है। यह प्रक्रिया पूरी तरह से स्वचालित (Automated) है, जिसमें किसी मानवीय हस्तक्षेप की आवश्यकता नहीं होती।

इस प्रणाली की कार्यप्रणाली पूरी तरह चरणबद्ध है - इमेज फोल्डर से एक-एक कर सभी तस्वीरों को पढ़ना, उन्हें "OpenCV" और "Matplotlib" की मदद से प्रदर्शित करना, और फिर DeepFace की सहायता से विश्लेषण करके चेहरे की भावनाओं का पता लगाना। उपयोगकर्ता को हर इमेज का परिणाम टर्मिनल पर सरल भाषा में मिलता है, जिसमें संबंधित इमेज का नाम और उसकी प्रमुख भावना (Dominant Emotion) होती है। यदि किसी इमेज में तकनीकी गङ्गाधारी है, तो वह भी स्पष्ट रूप से दिखा दिया जाता है।

इस तकनीक का उपयोग अनेक क्षेत्रों में किया जा सकता है, जैसे – मानसिक स्वास्थ्य मूल्यांकन, शिक्षा में छात्रों की प्रतिक्रिया का विश्लेषण, ग्राहक सेवा, और सुरक्षा प्रणालियाँ। यह मशीनों को केवल "बुद्धिमान" ही नहीं, बल्कि "संवेदनशील" भी बनाता है।

अंततः, "expression.py" प्रोजेक्ट एक महत्वपूर्ण उदाहरण है कि कैसे कंप्यूटर विज़न और डीप लर्निंग को मिलाकर हम मशीनों को मानवीय व्यवहार और भावनाओं को समझने के योग्य बना सकते हैं। यह प्रोजेक्ट न केवल तकनीकी दृष्टिकोण से उपयोगी है, बल्कि यह आने वाले समय में समाज के लिए अत्यंत लाभकारी सिद्ध हो सकता है।

यह प्रोजेक्ट उन छात्रों, शोधकर्ताओं और डेवलपर्स के लिए एक प्रेरणा है जो आर्टिफिशियल इंटेलिजेंस के मानवीय पक्ष पर काम करना चाहते हैं।

## VIII सन्दर्भ (References)

- <https://www.salesforce.com/in/products/what-is-salesforce/>
- <https://www.salesforce.com/in/>
- <https://www.simplilearn.com/what-is-salesforce-article>
- [https://www.reddit.com/r/explainlikeimfive/comments/15bnx41/eli5\\_what\\_is\\_salesforce/](https://www.reddit.com/r/explainlikeimfive/comments/15bnx41/eli5_what_is_salesforce/)
- <https://www.techtarget.com/searchcustomerexperience/definition/Salesforcecom>
- <https://ascendix.com/blog/what-is-salesforce-what-salesforce-is-used-for/>
- <https://ascendix.com/blog/what-is-salesforce-what-salesforce-is-used-for/>
- <https://www.geeksforgeeks.org/machine-learning/>
- <https://github.com/topics/artificial-intelligence-projects>
- <https://www.upgrad.com/blog/artificial-intelligence-projects-in-python/>
- <https://codewave.com/insights/building-ai-ml-projects-python/>
- <https://codewave.com/insights/building-ai-ml-projects-python/>
- <https://www.saasguru.co/>
- <https://www.salesforceben.com/>
- <https://thinqloud.com/>
- <https://gearset.com/>
- <https://www.kaggle.com/c/dtw2/data>