

INTEGRATION OF CHATGPT WITH WHATSAPP

**A MINOR PROJECT REPORT SUBMITTED
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF DEGREE OF**

**BACHELOR OF ENGINEERING
In
COMPUTER SCIENCE AND ENGINEERING**

SUBMITTED BY

Tazeem Nazir - 2020A1R017 | Shakir Malik – 2020A1R019
Tawheed Musaib – 2020A1T023 | Iftisam Tariq – 2020A1L017
Poonam Bhalwal – 2020A1R020



SUBMITTED TO

Department of Computer Science & Engineering

(Accredited by NBA)

Model Institute of Engineering and Technology (Autonomous)

Jammu, India

2023

CANDIDATE’S DECLARATION

We, **Tazeem Nazir, Shakir Malik, Tawheed Musaib, Iftisam Tariq, Poonam Bhalwal** hereby declare that the work which is being presented in the minor project report entitled, **“Integration of ChatGPT with WhatsApp”** in the partial fulfillment of requirement for the award of degree of B.E. (CSE) and submitted in the Department of Computer Science & Engineering, Model Institute of Engineering and Technology (Autonomous), Jammu is an authentic record of my own work carried by me under the supervision of **Ms. Shafalika Vijayal** (Asst. Prof. CSE, MIET Jammu). The matter presented in this seminar report has not been submitted to this or any other University / Institute for the award of the B.E. Degree.

Signature of the Students

Dated: 16/12/2023.

**Tazeem Nazir, Shakir Malik,
Tawheed Musaib, Iftisam Tariq,
Poonam Bhalwal**

Computer Science and Engineering
Model Institute of Engineering and Technology (Autonomous)
Kot Bhalwal, Jammu, India
(NAAC “A” Grade Accredited)

Ref. No.: CSE-P42

Date: 12/12/2023

CERTIFICATE

Certified that this minor project report entitled “**Integration of ChatGPT with WhatsApp**” is the bonafide work of “**Tazeem Nazir (17), Shakir Malik (19), Tawheed Musaib (T023), Iftisam Tariq (L017), Poonam Bhalwal (20), of 7th Semester, CSE, Model Institute of Engineering and Technology (Autonomous), Jammu**”, who carried out the seminar work under my supervision in September 2023.

Ms. Shafalika Vijayal
Coordinator
Program Manager
Computer Science & Engineering, MIET

ACKNOWLEDGEMENTS

This Minor Project was a great chance for learning and professional development for us. I got to learn about new and updated technologies. We are therefore very lucky to have been provided with the opportunity to be a part of it. I am also grateful for having had the chance to receive the guidance of intellectual professionals who led us through this seminar tenure.

We would like to express my sincere gratitude to Ms. Shafalika Vijayal, Asst. Prof. CSE and Program Manager, MIET for their invaluable support and insightful suggestions throughout the seminar process. Your mentorship and dedication to nurturing our understanding of the subject matter have greatly enriched the content and quality of this report.

We are thankful to Prof. (Dr.) Ashok Kumar, Dean Academics Affairs, MIET for fostering an environment of academic excellence within our department, which has been a constant source of motivation.

We extend my appreciation to Prof. (Dr.) Ankur Gupta, Director, MIET for providing overall direction and encouragement, making it possible for us to engage in meaningful research and learning.

We would also like to thank our parents for their unwavering support and encouragement, and teammates who were part of this Minor Project.

We express our sincere gratitude to Model Institute of Engineering and Technology (Autonomous), Jammu for giving me the opportunity to work on the Seminar Report during our final year of B.E.

At the end thanks to the Almighty for Everything.

Tazeem Nazir, Shakir Malik,

Tawheed Musaib, Iftisam Tariq,

Poonam Bhalwal

ABSTRACT

Artificial Intelligence (AI) represents the pinnacle of technological advancement, encompassing a broad spectrum of tools and techniques that enable machines to simulate human-like intelligence. Among these groundbreaking advancements, Language Models (LMs) stand as a remarkable subset of AI. These LMs, such as GPT (Generative Pre-trained Transformer) models, have revolutionized how machines process and generate text, exhibiting an unprecedented capacity to understand context, generate coherent responses, and even assist in various tasks requiring linguistic prowess. The fusion of AI and Language Models has opened doors to a new era of communication, empowering machines to engage in complex conversations, profoundly impacting numerous sectors ranging from customer service to content generation and beyond. The integration of ChatGPT with WhatsApp presents a comprehensive exploration into the convergence of advanced language models and one of the most widely used messaging platforms globally. This project aims to bridge the capabilities of OpenAI's ChatGPT with the functionalities of WhatsApp, enabling users to interact seamlessly with an AI-powered conversational agent within the familiar environment of the messaging app. The integration involves leveraging APIs, natural language processing techniques, and backend systems to facilitate real-time communication between ChatGPT and WhatsApp users. The project aims to enhance user experience, offer personalized assistance, and provide valuable insights into the potential applications of AI-powered chatbots within messaging platforms. Through this integration, the project seeks to demonstrate the practicality and versatility of employing sophisticated language models like ChatGPT in everyday communication channels, potentially revolutionizing the way users engage and interact on messaging platforms.

Contents

Candidates' Declaration	1
Certificate	2
Acknowledgement	3
Abstract	4
Contents	5
List of Figures	7
Abbreviations Used	8
Chapter 1 INTRODUCTION TO LANGUAGE MODELS	9-11
1.1 What is Language Model?	9
1.2 What is Natural Language Processing (NLP)	9-10
1.3 What Language Models can do?	10-11
Chapter 2 INTRODUCTION TO CHATGPT	12-15
2.1 What is GPT?	12
2.2 Why is GPT important?	12
2.3 What are the use cases of GPT?	12-13
2.4 What is ChatGPT?	14
2.5 How does ChatGPT work?	14-15
Chapter 3 APPLICATION PROGRAMMING INTERFACE	16-22
3.1 What is Application Programming Interface (API)?	16
3.2 How do APIs work?	16-17
3.3 What is the history of APIs?	17-18
3.4 What are the different types of APIs and how they are used.	19-20
3.5 What are some common API use cases	21-22

Chapter 4 WHATSAPP BOTS & HOW IT WORKS	23-24
4.1 Why use WhatsApp bots?	23-24
Chapter 5 INTEGRATION OF ChatGPT WITH WhatsApp	25-29
CONCLUSION	30-31
REFERENCES	32

LIST OF FIGURES

Figure No.	Caption	Page No.
1)	Mobile Keyboard with word suggestions	10
2)	Types of APIs	19
3)	Chatbot Illustration	24
4)	Architecture Diagram of the project	25
5)	Twilio Interface	26
6)	Twilio API Setup	26
7)	OpenAI API setup	27
8)	Code that integrates APIs and the server.	27
9)	Main code file to execute the integration	28
10)	NGROK server	29
11)	Demonstration	29

ABBREVIATIONS USED

AI	Artificial Intelligence
API	Application Programming Interface
GPT	Generative Pre-trained Transformer
LM	Language Model
NLP	Natural Language Processing
HTTP	Hyper Text Transfer Protocol
FRT	First Response Time
POS	Part-of-Speech
RPC	Remote Procedure Calls
CRM	Customer Relationship Management
IoT	Internet of Things
SSO	Single Sign-on

Chapter 1: INTRODUCTION TO LANGUAGE MODELS

1.1 What is a language model?

A **language model** is a type of machine learning model trained to conduct a probability distribution over words. Put it simply, a model tries to predict the next most appropriate word to fill in a blank space in a sentence or phrase, based on the context of the given text.

For example, in a sentence that sounds like this, “*Jenny dropped by the office for the keys so I gave them to [...],*” a good model will determine that the missed word is likely to be a pronoun. Since the relevant piece of information here is *Jenny*, the most probable pronoun is *she* or *her*.

The important thing is that the model doesn’t focus on grammar, but rather on how words are used in a way that is like how people write. [1]

Language models are a fundamental component of natural language processing (NLP) because they allow machines to understand, generate, and analyze human language. They are mainly trained using a large dataset of text, such as a collection of books or articles. Models then use the patterns they learn from this training data to predict the next word in a sentence or generate new text that is grammatically correct and semantically coherent.

1.2 What is Natural Language Processing (NLP)?

Natural language processing (NLP) refers to the branch of computer science—and more specifically, the branch of artificial intelligence or AI—concerned with giving computers the ability to understand text and spoken words in much the same way human beings can.

NLP combines computational linguistics—rule-based modeling of human language—with statistical, machine learning, and deep learning models. Together, these technologies enable computers to process human language in the form of text or voice data and to ‘understand’ its full meaning, complete with the speaker or writer’s intent and sentiment.

NLP drives computer programs that translate text from one language to another, respond to spoken commands, and summarize large volumes of text rapidly—even in real time. There's a good chance you've interacted with NLP in the form of voice-operated GPS systems, digital assistants, speech-to-text dictation software, customer service chatbots, and other consumer conveniences. But NLP also plays a growing role in enterprise solutions that help streamline business operations, increase employee productivity, and simplify mission-critical business processes. [2]

1.3 What language models can do?

Have you ever noticed the smart features in Google Gboard and Microsoft SwiftKey keyboards that provide auto-suggestions to complete sentences when writing text messages? This is one of the numerous use cases of language models.

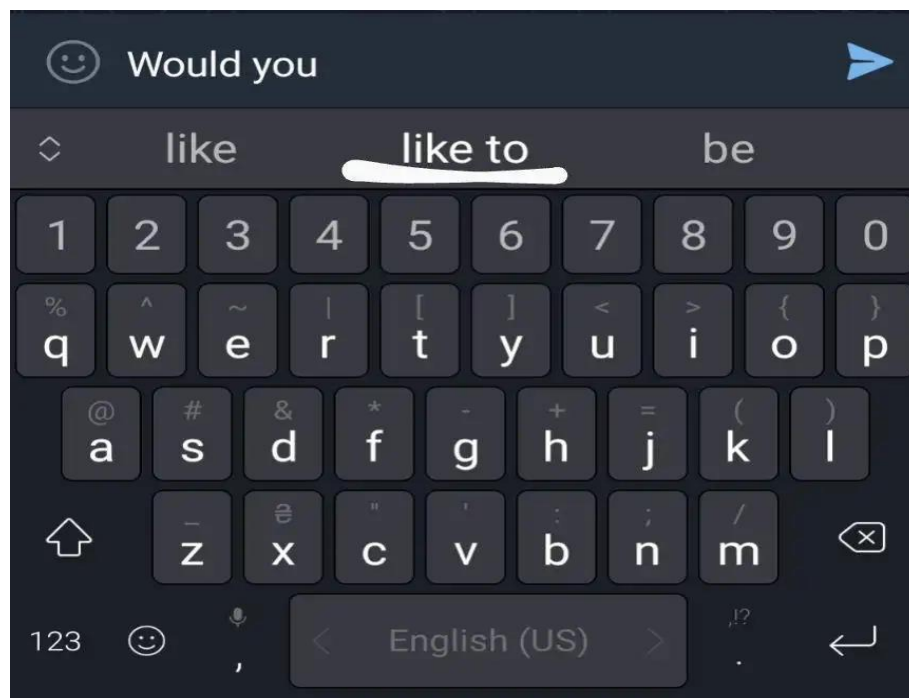


Fig-1: Mobile Keyboard with word suggestions

Language models are used in a variety of NLP tasks, such as speech recognition, machine translation, and text summarization.

Content generation. One of the areas where language models shine the brightest is content generation. This includes generating complete texts or parts of them based on the data and terms provided by humans. Content can range from news articles, press releases, and blog posts to online store product descriptions, poems, and guitar tabs, to name a few.

Part-of-speech (POS) tagging. Language models have been widely used to achieve state-of-the-art results on POS tagging tasks. POS tagging is the process of marking each word in a text with its corresponding part of speech, such as noun, verb, adjective, etc. The models are trained on large amounts of labeled text data and can learn to predict the POS of a word based on its context and the surrounding words in a sentence.

Question answering. Language models can be trained to understand and answer questions with and without the context given. They can provide answers in multiple ways, such as by extracting specific phrases, paraphrasing the answer, or choosing from a list of options.

Text summarization. Language models can be used to automatically shorten documents, papers, podcasts, videos, and more into their most important bites. Models can work in two ways: extract the most important information from the original text or provide summaries that don't repeat the original language.

Conversational AI. Language models are an inevitable part of speech-enabled applications that require converting speech to text and vice versa. As a part of conversational AI systems, language models can provide relevant text responses to inputs.

Code completion. Recent large-scale language models have demonstrated an impressive ability to generate code, edit, and explain code. However, they can complete only simple programming tasks by translating instructions into code or checking it for errors. [1]

Chapter 2: INTRODUCTION TO ChatGPT

2.1. What is GPT?

Generative Pre-trained Transformers, commonly known as GPT, are a family of neural network models that uses the transformer architecture and is a key advancement in artificial intelligence (AI) powering generative AI applications such as ChatGPT. GPT models give applications the ability to create human-like text and content (images, music, and more), and answer questions in a conversational manner. Organizations across industries are using GPT models and generative AI for Q&A bots, text summarization, content generation, and search.

2.2. Why is GPT important?

The GPT models, and in particular, the transformer architecture that they use, represent a significant AI research breakthrough. The rise of GPT models is an inflection point in the widespread adoption of ML because the technology can be used now to automate and improve a wide set of tasks ranging from language translation and document summarization to writing blog posts, building websites, designing visuals, making animations, writing code, researching complex topics, and even composing poems. The value of these models lies in their speed and the scale at which they can operate. For example, where you might need several hours to research, write, and edit an article on nuclear physics, a GPT model can produce one in seconds. GPT models have sparked the research in AI towards achieving artificial general intelligence, which means machines can help organizations reach new levels of productivity and reinvent their applications and customer experiences.

2.3. What are the use cases of GPT?

The GPT models are general-purpose language models that can perform a broad range of tasks from creating original content to writing code, summarizing text, and extracting data from documents.

Here are some ways you can use the GPT models:

Create social media content: Digital marketers, assisted by artificial intelligence (AI), can create content for their social media campaigns. For example, marketers can prompt a GPT model to produce an explainer video script. GPT-powered image processing software can create memes, videos, marketing copy, and other content from text instructions.

Convert text to different styles: GPT models generate text in casual, humorous, professional, and other styles. The models allow business professionals to rewrite a particular text in a different form. For example, lawyers can use a GPT model to turn legal copies into simple explanatory notes.

Write and learn code: As language models, the GPT models can understand and write computer code in different programming languages. The models can help learners by explaining computer programs to them in everyday language. Also, experienced developers can use GPT tools to autosuggest relevant code snippets.

Analyze data: The GPT model can help business analysts efficiently compile large volumes of data. The language models search for the required data and calculate and display the results in a data table or spreadsheet. Some applications can plot the results on a chart or create comprehensive reports.

Produce learning materials: Educators can use GPT-based software to generate learning materials such as quizzes and tutorials. Similarly, they can use GPT models to evaluate the answers.

Build interactive voice assistants: The GPT models allow you to build intelligent interactive voice assistants. While many chatbots only respond to basic verbal prompts, the GPT models can produce chatbots with conversational AI capabilities. In addition, these chatbots can converse verbally like humans when paired with other AI technologies. [3]

2.4. What is ChatGPT?

ChatGPT is a language generation model developed by OpenAI, part of the GPT (Generative Pre-trained Transformer) series. It's designed to understand, generate, and respond to human-like text based on the patterns it learns from massive amounts of data.

2.5. How does ChatGPT work?

1. Transformer Architecture: ChatGPT is built on a Transformer architecture, which enables it to process and generate text in a sequence-to-sequence manner. This architecture uses self-attention mechanisms to understand relationships between words in a sentence, allowing for more coherent and contextually relevant responses.

2. Pre-training on Diverse Data: Before it's used for specific tasks, ChatGPT undergoes extensive pre-training on a diverse range of internet text. This process involves exposing the model to vast amounts of text data from books, articles, websites, and other sources to learn the nuances of language.

3. Fine-tuning for Specific Tasks: After pre-training, ChatGPT can be fine-tuned on more specific datasets or tasks to enhance its performance in particular domains. This fine-tuning allows the model to specialize in areas like customer service, content generation, or other applications.

4. Text Generation and Understanding: When prompted with text input, ChatGPT uses its learned patterns and context from the training data to generate text that follows the input sequence. It understands the context of the conversation to generate relevant and coherent responses.

5. Text Generation and Understanding: When prompted with text input, ChatGPT uses its learned patterns and context from the training data to generate text that follows the input sequence. It understands the context of the conversation to generate relevant and coherent responses.

6. Deployment and Integration: ChatGPT can be integrated into various applications and platforms through APIs (Application Programming Interfaces), enabling it to interact with users in real-time across websites, messaging apps, customer support systems, and more.

Overall, ChatGPT operates by leveraging its vast knowledge of language and context acquired through training on extensive datasets, allowing it to generate human-like responses in natural language conversations. [4]

Chapter 3: APPLICATION PROGRAMMING INTERFACE

3.1 What is Application Programming Interface (API)?

An API, which stands for application programming interface, is a set of protocols that enable different software components to communicate and transfer data. Developers use APIs to bridge the gaps between small, discrete chunks of code to create applications that are powerful, resilient, secure, and able to meet user needs. Even though you can't see them, APIs are everywhere—working continuously in the background to power the digital experiences that are essential to our modern lives.

Here, we'll give a high-level overview of how APIs work before reviewing the different types of APIs and how they are used. We'll also discuss some common use cases for APIs—and offer a few real-world API examples that can help you get started.

3.2 How do APIs Work?

APIs work by sharing data between applications, systems, and devices. This happens through a request and response cycle. A user initiates a request for data by interacting with an application. The request is sent to the API, which retrieves the data and returns it to the user.

In order to better understand this process, it can be useful to think of APIs like restaurants. In this metaphor, the customer is like the user, who tells the waiter what she wants. The waiter is like an API, receiving the customer's order and translating it into easy-to-follow instructions for the kitchen—sometimes using specific codes or abbreviations that the kitchen staff will recognize. The kitchen staff is like the API server because it creates the order according to the customer's specifications and gives it to the waiter, who then delivers it to the customer.

Once that metaphor makes sense, you can go a level deeper and start reviewing the different components of an API, starting with the API client. The API client is responsible for assembling requests in response to user actions and sending them to the appropriate API endpoint. Endpoints are Uniform Resource Identifiers (URIs) that provide access to specific resources in a database. For instance, if a user wants to see all the products at an e-commerce store, the API client will send a GET request to the /products endpoint.

3.3 What is the history of APIs?

In order to fully understand the role that APIs play in our lives, it's important to understand how they have evolved. APIs have been around for decades, with modern web APIs first taking shape in the early 2000s. The history of APIs since that period can be roughly broken down into the following five phases:

Phase 1: Commercial APIs –

In the early 2000s, web APIs emerged as a new method for emerging startups to not only make products and services available online, but to also enable partners and third-party resellers to extend the reach of their platforms. This era of APIs was defined by Salesforce, eBay, and Amazon, and these companies continue to dominate the API playing field today.

Phase 2: Social Media APIs –

A shift in the API landscape occurred in the mid-2000s, as a new group of companies—such as Flickr, Facebook, and Twitter—realized that APIs could change the way we share information with one another. While these APIs weren't as intrinsically linked to revenue as their commercial predecessors, they nevertheless provided significant value to their organizations. For instance, Facebook launched version 1.0 of its API in August of 2006, which allowed developers to access Facebook users' friends, photos, events, and profile information. This API played a crucial role in establishing Facebook as one of the most popular social networks in the world.

Phase 3: Cloud APIs –

In 2006, Amazon introduced Amazon Simple Storage (S3), which marked yet another turning point in the history of APIs. S3 is a basic storage service in which resources are accessible via API and CLI, and its pay-as-you-go model provides a cost-efficient way for organizations to monetize digital assets in the online economy. Just six months later, Amazon released Amazon Elastic Compute (EC2), which enabled developers to use web APIs to deploy infrastructure that would power the next generation of applications. Both S3 and EC2 continue to play an essential role in application development today.

Phase 4: APIs for mobile Applications –

The world was introduced to Apple's iPhone and Google's Android in 2007. The ability to carry the web in our pockets radically changed how we live—and spurred a massive investment in mobile applications that are powered by APIs.

For instance, Twilio launched its API-as-a-product platform in 2007, which allowed developers to make and receive phone calls from any cloud application. Instagram then launched its photo-sharing iPhone application in October 2010, and it had one million users just three months later. Instagram did not initially provide an API, but it began working on one in early 2011 in response to user demand. These API-first companies played an essential role in creating the blueprint for how APIs are delivered today.

Phase 5: APIs for connected devices –

Around 2010, some developers began using APIs to connect everyday objects—such as cameras, thermostats, speakers, microphones, and sensors—to the cloud. This next generation of devices, which includes Fitbit, Nest, Alexa, can send and receive data, content, media, and other digital resources, further changing the way we interact with the world around us.

3.4 What are the different types of APIs, and how are they used?

There are many different types of APIs and ways to categorize them. For instance, you can categorize APIs by who has access to them. This organizational framework includes:

- **Private APIs:** Private APIs, also known as internal APIs, are used to connect different software components within a single organization, and they are not available for third-party use. For instance, a social media application might have a private API that handles the login workflow, another private API that handles the feed, and yet another private API that facilitates communication between users. Some applications may include dozens or even hundreds of private APIs.
- **Public APIs:** Public APIs provide public access to an organization's data, functionality, or services, which third-party developers can integrate into their own applications. Some public APIs are available for free, while others are offered as billable products. For instance, an e-commerce application may incorporate a public payment API, such as Stripe, to handle payment processing without having to build that functionality from scratch.
- **Partner APIs:** Partner APIs enable two or more companies to share data or functionality to collaborate on a project. They are not available to the public and therefore leverage authentication mechanisms to ensure they are only used by authorized partners.

Types of APIs

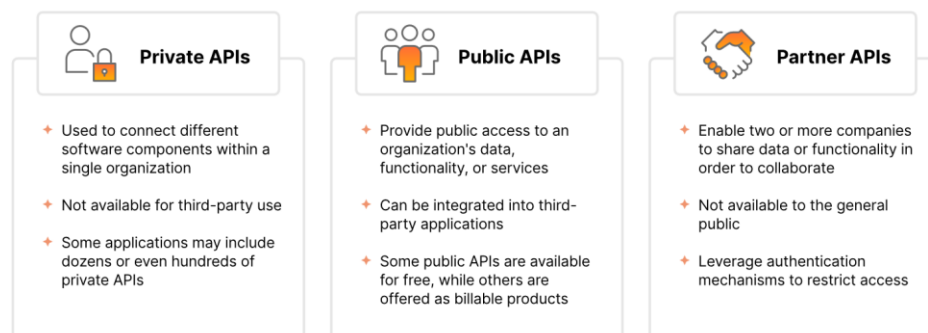


Fig-2: Types of API, Source: - Postman

We can also categorize APIs according to their architectural style, of which there are many. The most frequently used architectural styles are:

- **REST:** As discussed above, REST is the most popular API architecture for transferring data over the internet. In a RESTful context, resources are accessible via endpoints, and operations are performed on those resources with standard HTTP methods such as GET, POST, PUT, and DELETE.
- **SOAP:** SOAP, which stands for Simple Object Access Protocol, uses XML to transfer highly structured messages between a client and server. SOAP is often used in enterprise environments or legacy systems, and while it includes advanced security features, it can be slower than other API architectures.
- **GraphQL:** GraphQL is an open-source query language that enables clients to interact with a single API endpoint to retrieve the exact data they need, without chaining multiple requests together. This approach reduces the number of round trips between the client and server, which can be useful for applications that may run on slow or unreliable network connections.
- **Webhooks:** Webhooks are used to implement event-driven architectures, in which requests are automatically sent in response to event-based triggers. For instance, when a specific event occurs in an application, such as a payment being made, the application can send an HTTP request to a pre-configured webhook URL with the relevant event data in the request payload. The system that receives the webhook can then process the event and take the appropriate action.
- **gRPC:** RPC stands for Remote Procedure Call, and gRPC APIs were originated by Google. In gRPC architectures, a client can call on a server as if it were a local object, which makes it easier for distributed applications and systems to communicate with one another.

3.5 What are some common API use cases?

APIs are extremely versatile, and they support a wide range of use cases that includes:

Integrating with internal and external systems:

One of the most common reasons developers turn to APIs is to integrate one system with another. For instance, you can use an API to integrate your customer relationship management (CRM) system with your marketing automation system, which would allow you to automatically send a marketing email when a sales representative adds a new prospective customer to the CRM.

Adding or enhancing functionality:

APIs let you incorporate additional functionality into your application, which can improve your customers' experience. For instance, if you're working on a food delivery application, you might incorporate a third-party mapping API to let users track their order while it's en route.

Connecting IoT devices:

APIs are essential to the Internet of Things (IoT) ecosystem, which includes devices such as smart watches, fitness trackers, doorbells, and home appliances. Without APIs, these devices would not be able to connect to the cloud—or to one another—which would render them useless.

Creating more scalable systems:

APIs are used to implement microservice-based architectures, in which applications are built as a collection of small services that communicate with one another through private APIs. Microservices are managed, deployed, and provisioned independently of one another, which enables teams to scale their systems in a reliable yet cost-efficient way.

Reducing costs:

APIs help organizations reduce operational costs by automating time-intensive tasks, such as sending emails, pulling reports, and sharing data between systems. They can also reduce development costs by enabling teams to reuse existing functionality, instead of reinventing the wheel.

Improving organizational security and governance:

APIs power many workflows that are essential for organizational security. For instance, single sign-on (SSO), which enables users to use one username and password for multiple systems, is made possible by APIs. APIs are also used to enforce and automate corporate governance rules and policies, such as a requirement that expenses be approved before employees are reimbursed.

This list is far from exhaustive, and it will keep growing as developers continue to create innovative solutions that change the ways we live, work, and interact with one another.

3.6 What are some real-world examples of APIs?

If you're looking for real-world examples of APIs, a good place to start is a public API catalog, such as Postman's Public API Network. The Public API Network, which supports a community of over 30 million developers, is a searchable, highly organized library of APIs that makes it easy to find the API that's right for you. You can use the network's search functionality if you're looking for something specific or browse by category if you want to get inspired. Some companies that have published great examples of APIs on the Public API Network include Salesforce, Notion, Discord, Pinterest, and DoorDash. [5]

Chapter 4: WHATSAPP BOTS AND HOW IT WORKS

WhatsApp chatbot is an automated software powered by rules or artificial intelligence (AI) and runs on the WhatsApp platform. People communicate with WhatsApp chatbot via the chat interface, like talking to a real person. It's a set of automated replies that simulates a human conversation on WhatsApp.

Since May 2022, WhatsApp opened the API for businesses of any size. Previously, it was available only for medium and large businesses, so smaller companies had to contact other providers to access WhatsApp API. [6]

4.1 Why use WhatsApp bots?

WhatsApp as a messaging channel has gained popularity amongst today's digital customers. In fact, 67% of consumers prefer using messaging apps when interacting with a business. Deploying chatbot on platforms like WhatsApp enables businesses to assist their customers wherever they are, with ease.

Here are the reasons, why you should use WhatsApp bots for your business:

- 2. Reduce Response Time:** The average First Response Time (FRT) for a live chat in all industries is 1 minute and 36 seconds. A low response rate creates more satisfied customers who tend to stick with a brand longer than usual and spend 14% more. Response time on WhatsApp can be reduced by using automated messages.

In the WhatsApp business app, these automated messages are available only in the form of "Greeting Messages" where a business can just set up an automated greeting message for the first customer interaction.

But with WhatsApp chatbot and APIs, businesses can set up custom flows for different conversation stages, create custom messages, reduce the response time, and can answer most asked questions without human intervention.

2. Help & Support: Despite all the FAQ databases you build, people will still reach out to the support agents to find out about their account details, any active services, the validity of the services or renew their plan. In that case, sending them again to a FAQ section degrades the customer experience. The best way is to provide all the answers automatically on WhatsApp and address any service request then and there.

Without WhatsApp bots, providing this great customer experience will require a lot of manual work and can lead to a lot of unaddressed queries.

Whereas, with WhatsApp chatbot you can automate all the queries and provide basic services like sending the current plan information, upgrading a plan, or renewing the current plan for a customer. [7]



Fig-3: Chatbot illustration. Source: - Appsierra.com

Chapter 5: INTEGRATION OF ChatGPT WITH WhatsApp

The integration of ChatGPT with WhatsApp presents a comprehensive exploration into the convergence of advanced language models and one of the most widely used messaging platforms globally. This project aims to bridge the capabilities of OpenAI's ChatGPT with the functionalities of WhatsApp, enabling users to interact seamlessly with an AI-powered conversational agent within the familiar environment of the messaging app. The integration involves leveraging APIs, natural language processing techniques, and backend systems to facilitate real-time communication between ChatGPT and WhatsApp users. The project aims to enhance user experience, offer personalized assistance, and provide valuable insights into the potential applications of AI-powered chatbots within messaging platforms. Through this integration, the project seeks to demonstrate the practicality and versatility of employing sophisticated language models like ChatGPT in everyday communication channels, potentially revolutionizing the way users engage and interact on messaging platforms.

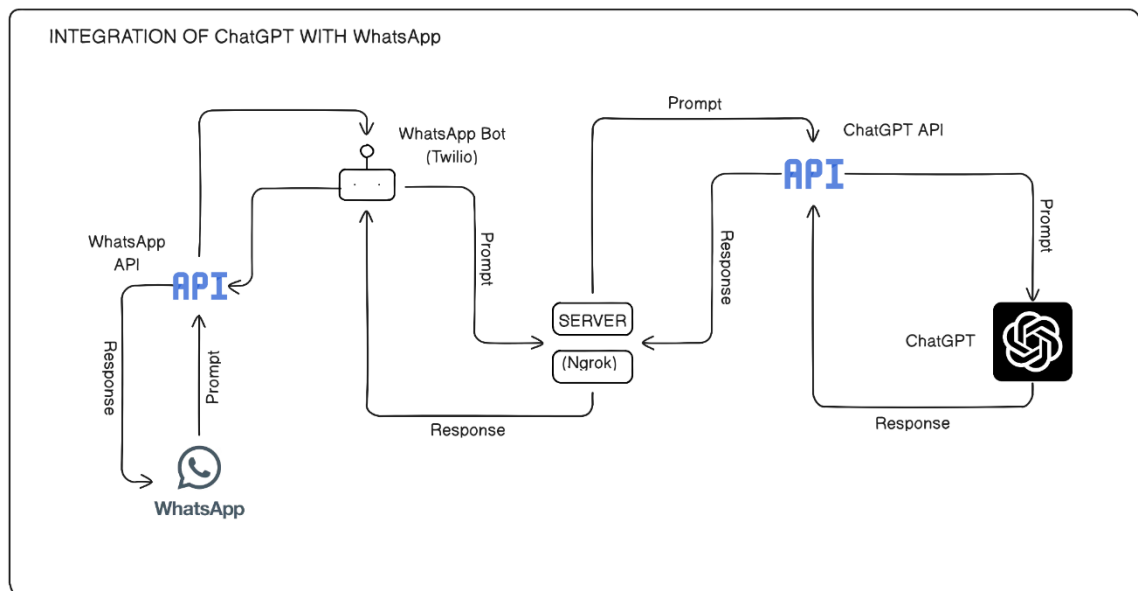


Fig-4: Architecture Diagram of the project.

Twilio Interface

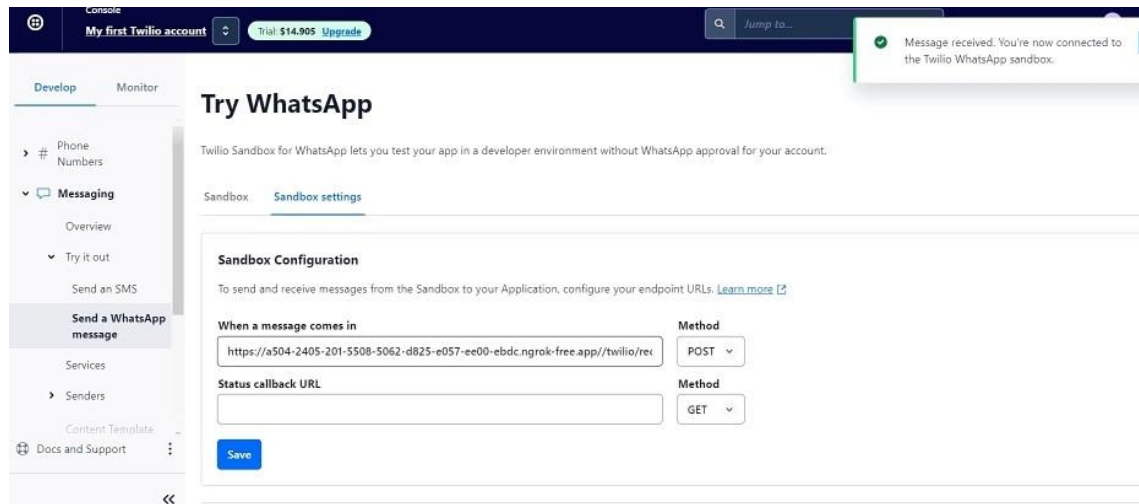


Fig-5: Twilio Interface.

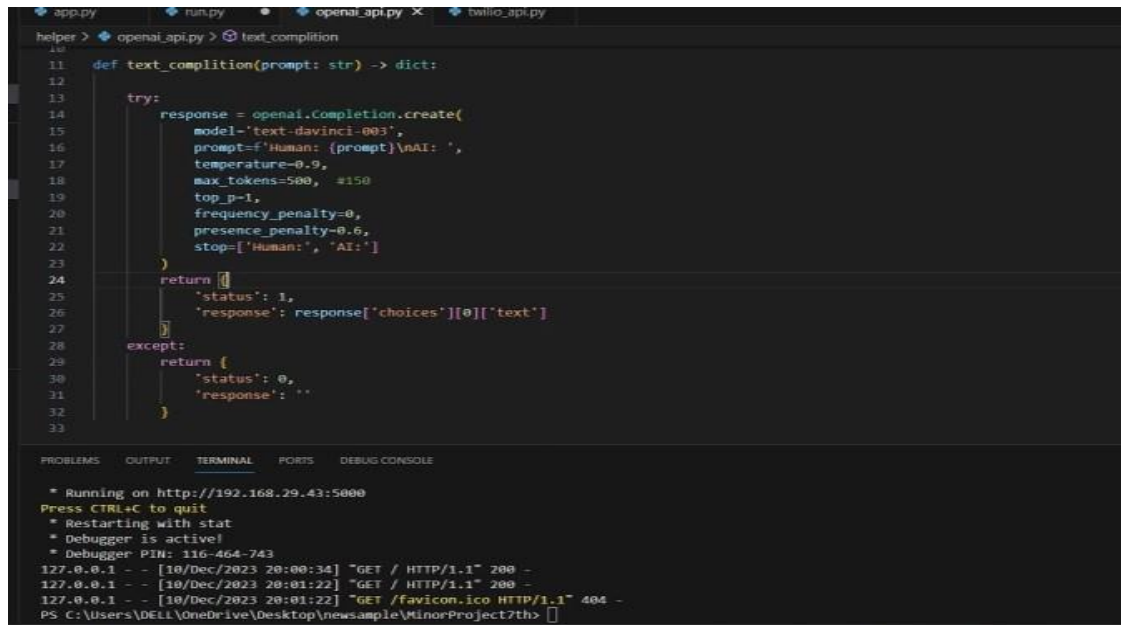
The messages sent from WhatsApp reach here and from here they are sent to the server.

```
app.py  run.py  openai_api.py  twilio_api.py x
helper > twilio_api.py > send_message
5 from dotenv import load_dotenv
6 load_dotenv()
7
8
9 account_sid = 'AC02723908f4f5666c9e9975fc1f835a2b'
10 auth_token = '93d58dca8eae0855ca9a374a56ed6a7'
11 client = Client(account_sid, auth_token)
12
13
14 def send_message(to: str, message: str) -> None:
15     |
16
17     _ = client.messages.create(from_='whatsapp:+14155238886', body=message, to=to)
18

PROBLEMS  OUTPUT  TERMINAL  PORTS  DEBUG CONSOLE
* Running on http://192.168.29.43:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PID: 116-464-743
127.0.0.1 - - [10/Dec/2023 20:00:34] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [10/Dec/2023 20:01:22] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [10/Dec/2023 20:01:22] "GET /favicon.ico HTTP/1.1" 404 -
PS C:\Users\DELL\OneDrive\Desktop\newsample\MinorProject7th>
```

Fig-6: Twilio API setup.

OpenAI API setup:



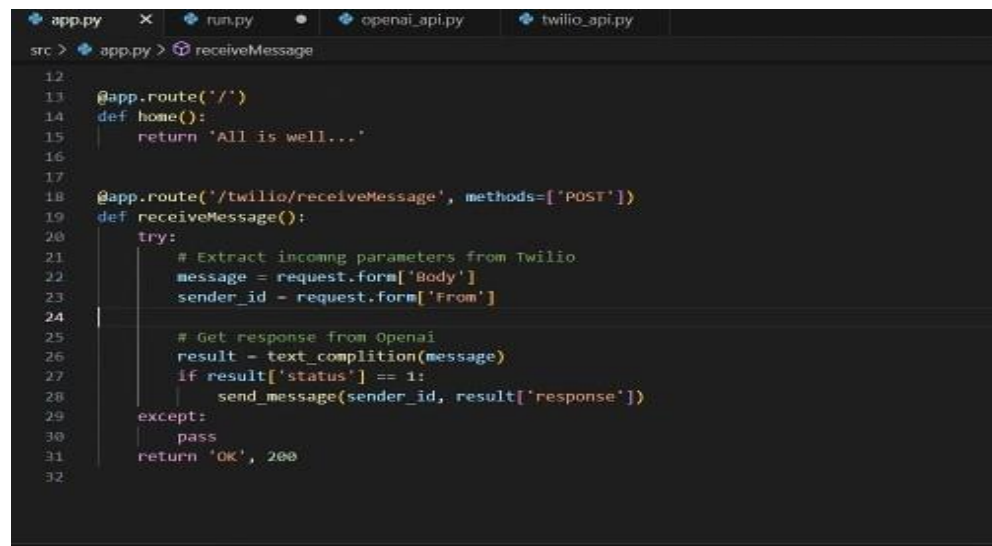
```
11 def text_completion(prompt: str) -> dict:
12
13     try:
14         response = openai.Completion.create(
15             model='text-davinci-003',
16             prompt=f'Human: {prompt}\nAI: ',
17             temperature=0.9,
18             max_tokens=500, #150
19             top_p=1,
20             frequency_penalty=0,
21             presence_penalty=0.6,
22             stop=['Human:', 'AI:']
23         )
24         return {
25             'status': 1,
26             'response': response['choices'][0]['text']
27         }
28     except:
29         return {
30             'status': 0,
31             'response': ''
32         }
33
```

* Running on http://192.168.29.43:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 116-464-743
127.0.0.1 - - [10/Dec/2023 20:00:34] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [10/Dec/2023 20:01:22] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [10/Dec/2023 20:01:22] "GET /favicon.ico HTTP/1.1" 404 -
PS C:\Users\DELL\OneDrive\Desktop\newsample\MinorProject7th>

Fig -7: OpenAI API setup

Here we setup of OpenAI API and add API keys to the code. Using this code the prompts from server will be sent to the ChatGPT and the responses from ChatGPT will be received by this API.

Integration:

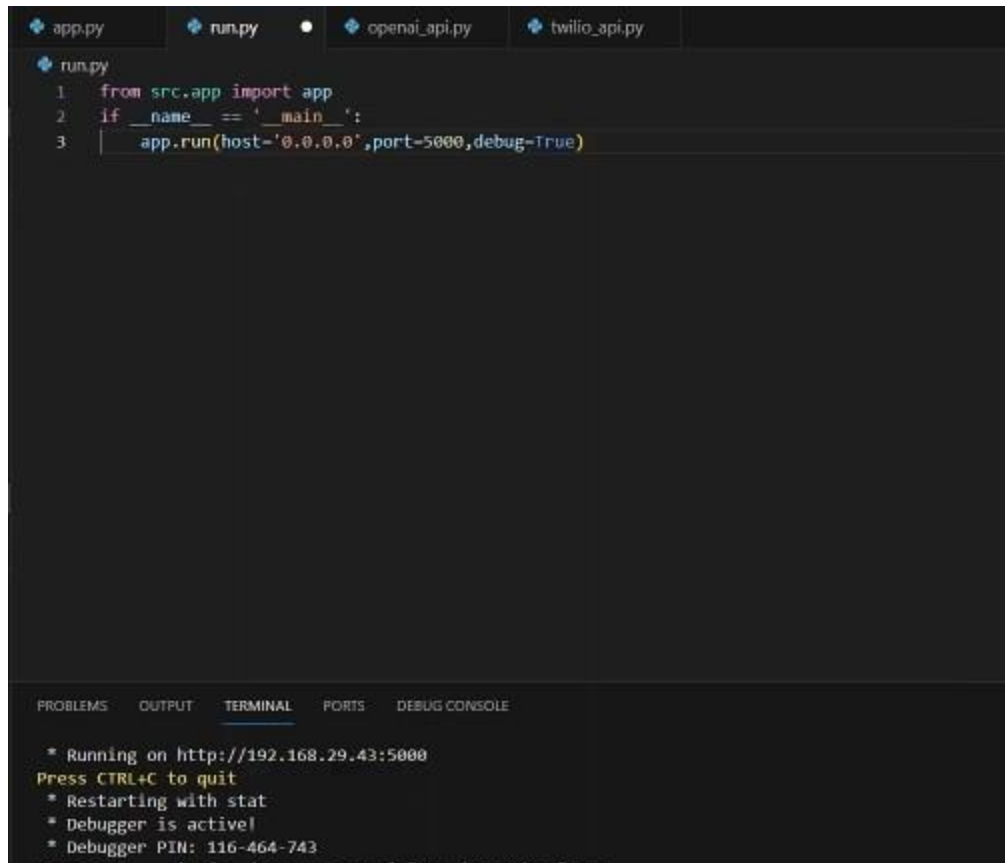


```
12
13 @app.route('/')
14 def home():
15     return 'All is well...'
16
17
18 @app.route('/twilio/receiveMessage', methods=['POST'])
19 def receiveMessage():
20     try:
21         # Extract incoming parameters from Twilio
22         message = request.form['Body']
23         sender_id = request.form['From']
24
25         # Get response from Openai
26         result = text_completion(message)
27         if result['status'] == 1:
28             send_message(sender_id, result['response'])
29     except:
30         pass
31     return 'OK', 200
32
```

Fig-8: Code that integrates APIs with server.

The above code integrates the APIs and creates a server. The code receives the prompt from WhatsApp using Twilio and sends the prompt to the OpenAI API. The code is also responsible for receiving response from ChatGPT API and send it to the user through WhatsApp.

Code to run the project:

A screenshot of a code editor with a dark theme. The editor has four tabs at the top: 'app.py', 'run.py', 'openai_api.py', and 'twilio_api.py'. The 'run.py' tab is active and shows the following Python code:

```
1 from src.app import app
2 if __name__ == '__main__':
3     app.run(host='0.0.0.0', port=5000, debug=True)
```

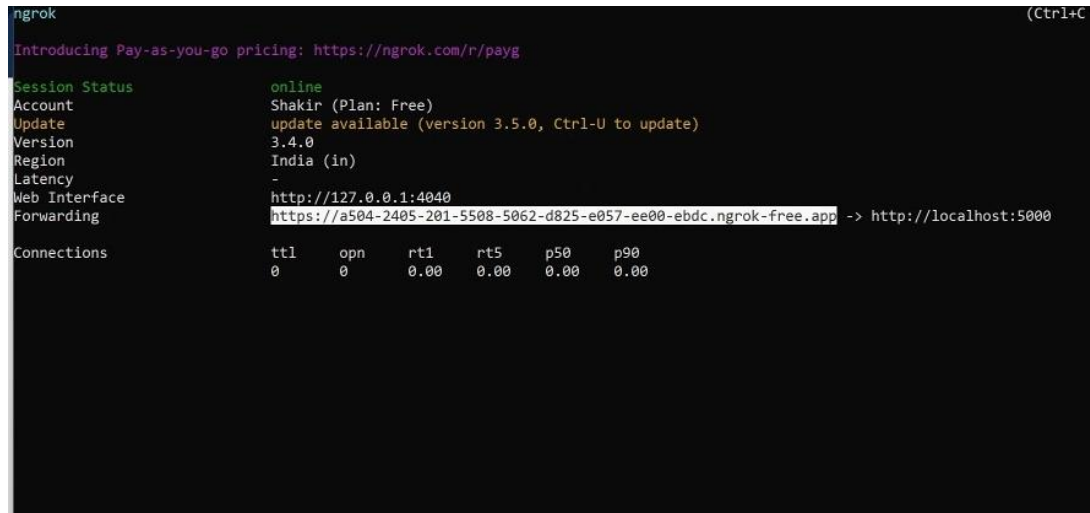
Below the code editor, there is a panel with tabs for 'PROBLEMS', 'OUTPUT', 'TERMINAL', 'PORTS', and 'DEBUG CONSOLE'. The 'TERMINAL' tab is active and displays the following output:

```
* Running on http://192.168.29.43:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 116-464-743
```

Fig-9: Main file to execute the integration.

When the above code is executed all the files associated with the project are executed and also it sets up the local server. After executing the file the user will be able to send the prompt from WhatsApp and will receive the response from the same from ChatGPT.

NGROK SERVER:



```
ngrok (Ctrl+C)
Introducing Pay-as-you-go pricing: https://ngrok.com/r/payg

Session Status      online
Account             Shakir (Plan: Free)
Update              update available (version 3.5.0, Ctrl-U to update)
Version             3.4.0
Region              India (in)
Latency              -
Web Interface       http://127.0.0.1:4040
Forwarding           https://a504-2405-201-5508-5062-d825-e057-ee00-ebdc.ngrok-free.app -> http://localhost:5000

Connections
  ttl   opn   rt1   rt5   p50   p90
    0     0    0.00  0.00  0.00  0.00
```

Fig 10: NGROK server.

Demonstration:

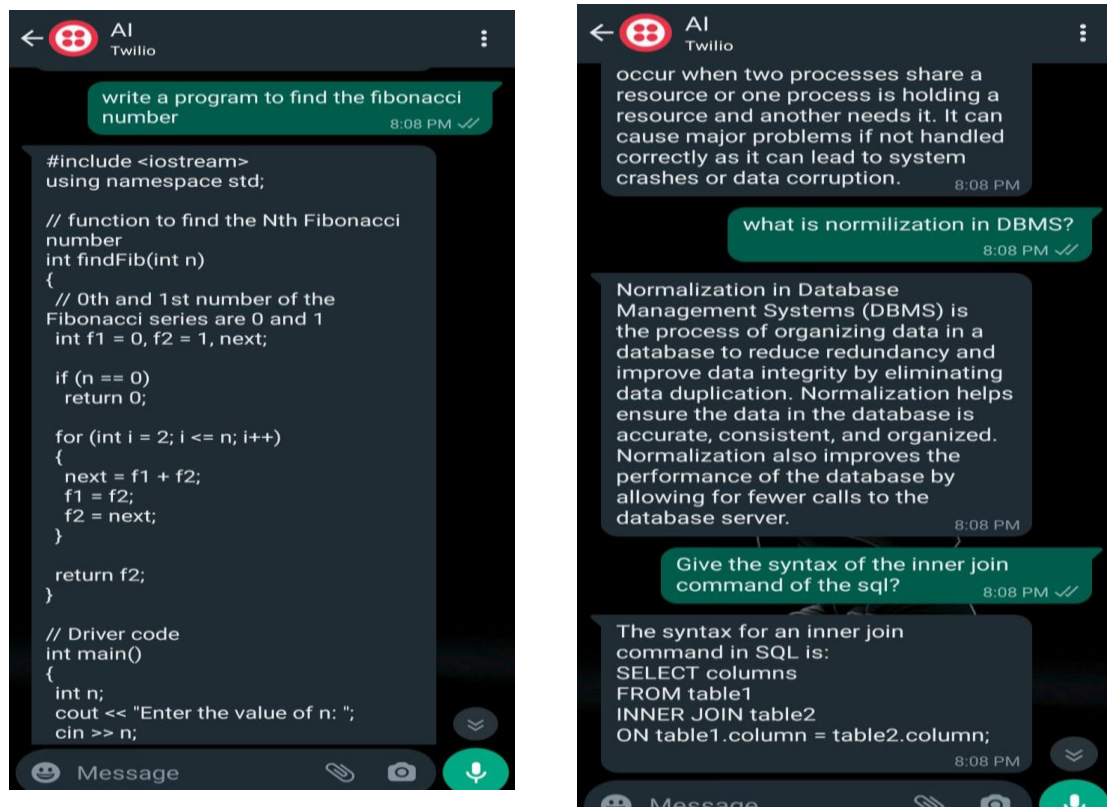


Fig-11: Demonstration.

CONCLUSION

The integration of ChatGPT with WhatsApp marks a significant milestone in harnessing the power of AI-driven conversational technology within a ubiquitous messaging platform. Throughout this project, the aim was to seamlessly merge ChatGPT's advanced language capabilities with the familiar interface of WhatsApp, revolutionizing user interactions and opening doors to a multitude of possibilities.

As this project culminates, several key points emerge:

- 1. Enhanced Conversational Experience:** The successful integration ensures a more natural, contextually aware interaction for WhatsApp users. ChatGPT's ability to understand queries and generate relevant, coherent responses within the messaging environment elevates the overall user experience.
- 2. Personalized Assistance and Customer Support:** Leveraging ChatGPT's capabilities enables personalized assistance, offering tailored information and support. This not only enhances user satisfaction but also provides businesses with efficient tools for customer service and issue resolution.
- 3. Expanded Functionality of WhatsApp:** Beyond its conventional messaging role, this integration expands WhatsApp's capabilities. It transforms the platform into a versatile hub for information retrieval, task automation, educational tools, and potentially disruptive innovations in various domains.
- 4. Potential for Future Innovations:** Successful integration sets the stage for future advancements. It lays the groundwork for more sophisticated applications, including AI-driven learning tools, refined customer engagement strategies, and seamless integration with emerging technologies.
- 5. Ethical and Privacy Considerations:** An integral part of this project was ensuring ethical AI use and safeguarding user privacy. Upholding transparency, fairness, and data protection principles remains essential for responsible AI integration.

In conclusion, the integration of ChatGPT with WhatsApp represents a significant stride in AI-driven communication technology. It not only enhances user experiences and business functionalities within the messaging platform but also sets the precedent for a future where AI-powered interactions redefine how individuals engage, learn, and transact within digital environments. As this project concludes, it emphasizes the importance of continual improvement, ethical implementation, and a user-centric approach in shaping the evolving landscape of AI-integrated platforms like WhatsApp.

REFERENCES

- [1] "AltexSoft.com," [Online]. Available: <https://www.altexsoft.com/blog/language-models-gpt/>.
- [2] "ibm.com," IBM, [Online]. Available: <https://www.ibm.com/topics/natural-language-processing>.
- [3] "AWS," Amazon, [Online]. Available: <https://aws.amazon.com/what-is/gpt/>.
- [4] "ChatGPT," OpenAI, [Online]. Available: chat.openai.com.
- [5] "Postman," Postman, [Online]. Available: <https://www.postman.com/what-is-an-api/>.
- [6] "botscrew.com," botscrew, [Online]. Available: <https://botscrew.com/blog/a-step-by-step-guide-to-create-chatbot-for-whatsapp-for-business/#:~:text=WhatsApp%20chatbot%20is%20an%20automated,a%20human%20conversation%20on%20WhatsApp..>
- [7] "freshworks.com," Freshworks, [Online]. Available: <https://www.freshworks.com/live-chat-software/customer-engagement/build-a-whatsapp-bot-blog/>.