

A
Project Report
on

VOICE BASED DELIVERY OF ACADEMIC RECORDS

Submitted in partial fulfillment for the requirements for the award of the
degree of

BACHELOR OF ENGINEERING

in

INFORMATION TECHNOLOGY

By

S HRUTURAJ	(245119737040)
G OM PRAKASH	(245119737027)
H KUSHAL	(245119737030)

Under the guidance of
CH. SRUJANA,
Assistant Professor, Dept. of I.T



MATURI VENKATA SUBBA RAO ENGINEERING COLLEGE

Department of Information Technology,

(Affiliated to Osmania University, Hyderabad)

Nadergul, Hyderabad, TELANGANA 501510

Academic year: 2022-23

**MATURI VENKATA SUBBA RAO(MVSR)
ENGINEERING COLLEGE**

**(Affiliated to Osmania University, Hyderabad, Recognized by AICTE)
Nadargul, Saroornagar Mandal, Hyderabad-501510**



DEPARTMENT OF INFORMATION TECHNOLOGY

CERTIFICATE

This is to certify that the project work entitled “Voice Based Delivery Academic Records” is a bonafide work carried out by Mr. S Hruturaj(2451-19-737-040), Mr. G.Om Prakash(2451-19-737-027), Mr. H.Kushal(2451-19-737-030) in fulfillment of the requirements for the award of degree of Bachelor of Engineering in Information Technology from Maturi Venkata Subba Rao Engineering College, affiliated to OSMANIA UNIVERSITY, Hyderabad, during the Academic Year 2022-23. under our guidance and supervision.

The results embodied in this report have not been submitted to any other university or institute for the award of any degree or diploma.

Signature of Project Coordinator

Signature of Guide

Signature of Head, ITD

Signature of External Examiner

DECLARATION

This is to certify that the work reported in the present project entitled “Voice Based Delivery Academic Records” is a record of bonafide work done by us in the Department of Information Technology, Maturi Venkata Subba Rao Engineering College, Osmania University. The reports are based on the project work done entirely by us and not copied from any other source.

The results embodied in this project report have not been submitted to any other University or Institute for the award of any degree or diploma to the best of our knowledge and belief.

Roll Number:	Student Name:	Signature of the Student
(2451-19-737-040)	S HRUTURAJ	
(2451-19-737-027)	G OM PRAKASH	
(2451-19-737-030)	H KUSHAL	

ACKNOWLEDGMENT

We with extreme jubilation and deepest gratitude would like to thank our guide, **CH. Srujana, Assitant Professor**, Department of Information Technology, Maturi Venkata Subba Rao (MVSR) Engineering College, for her constant encouragement to us to complete our work in time.

With immense pleasure, we sincerely thank our beloved Head of the department **Dr.K.VenuGopal Rao Dean-Academics & HOD, Department of Information Technology**, Maturi Venkata Subba Rao Engineering College, for permitting and providing facilities to carry out this project.

We would like to extend our gratitude to **Dr. D. Shanthi, Associate Professor, Project Coordinator** , **N. Hanumantha Rao, Assistant Professor**, **T. Anjali, Assistant Professor**, **A. Lalitha, Assistant Professor**, **B. Prasad, Assistant Professor, Department of Information Technology**, Maturi Venkata Subba Rao Engineering College, for their valuable suggestions and timely help during the course of the project.

Finally, we express, from the bottom of our heart and deepest gratitude to the entire faculty, my parents and family for the support, dedication, comprehension and love.

S HRUTURAJ	(2451-19-737-040)
G OM PRAKASH	(2451-19-737-027)
H KUSHAL	(2451-19-737-030)

Maturi Venkata Subba Rao Engineering College
Department of Information Technology

COURSE NAME: Project Work II

COURSE CODE: PW861IT

VISION

To impart technical education to produce competent and socially responsible engineers in the field of Information Technology.

MISSION

- a. To make teaching learning process effective and stimulating.
- b. To provide adequate fundamental knowledge of sciences and Information Technology with positive attitude.
- c. To create an environment that enhances skills and technologies required for industry.
- d. To encourage creativity and innovation for solving real world problems.
- e. To cultivate professional ethics in students and inculcate a sense of responsibility towards society

PROGRAM EDUCATIONAL OBJECTIVES(PEOS)

The Program Educational Objectives of undergraduate program in Information Technology are to prepare graduates who will:

- I. Apply knowledge of mathematics and Information Technology to analyze, design and implement solutions for real world problems in core or in multidisciplinary areas.
- II. Communicate effectively, work in a team, practice professional ethics and apply knowledge of computing technologies for societal development.
- III. Engage in Professional development or postgraduate education to be a life-long learner.

(A) PROGRAM OUTCOMES(POs)

At the end of the program the students (Engineering Graduates) will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyse complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principle and apply 6 these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Lifelong learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

(A) PROGRAM SPECIFIC OUTCOMES (PSOs)

- 1. Hardware design:** An ability to analyse, design, simulate and implement computer hardware / software and use basic analogue/digital circuits, VLSI design for various computing and communication system applications.
- 2. Software design:** An ability to analyse a problem, design algorithm, identify and define the computing requirements appropriate to its solution and implement the same.

COURSE OBJECTIVES AND OUTCOMES

Course Objectives

1. To enhance practical & Professional skills.
2. To familiarize the tools and techniques of symmetric literature survey and documentation.
3. To expose students to industry practices and teamwork.
4. To encourage students to work with innovative and entrepreneurial ideas

Course Outcomes

On successful completion of this course student will be

1. Define a problem of the recent advancements with applications towards society.
2. Outline requirements and perform requirement analysis for solving the problem.
3. Design and develop software and/or hardware, based solutions within the scope of project using contemporary technologies and tools.
4. Test and deploy the applications for use.
5. Develop the Project as a team and demonstrate the application, with effective written and oral communications.

ABSTRACT

An Interactive Voice Response System (IVRS) is an automation tool that generates automated voice to address the queries raised by a human through interactive voice response (speech recognition) and (DTMF) tones input provided via keypad. In this project, we are implementing a robust automation process for ordering delivery of academic certificates through an interactive voice response system and making payment against the charge. This system provides dynamic information related to academics and routes them to the appropriate section based on the text input options. This technology reduces human effort and gives the caller a better experience by routing him to the exact option. DTMF decoding and speech recognition are used to interpret the caller's response to voice prompts. DTMF tones are entered via the telephone keypad. Other technologies include using text-to-speech to speak complex and dynamic information. This IVR system is an efficient and cost-effective solution for establishing a personalized customer experience. The best part about IVR is that it's not dependent on humans to work. Once it's programmed, it will continue to function unless configured otherwise. It is capable of providing 24hrs service to its customers.

LIST OF FIGURES

Figure No.	Name Of the Figure	Page No.
4.1	System Architecture	10
4.2	Use Case Diagram	12
4.3	Class Diagram	13
4.4	Sequence Diagram	14
4.5	Activity Diagram	15
5.1	VS Code	16
5.2	Anaconda	17
5.3	Mongo DB	18
5.4	Twilio	19
5.5	ngork	20
6.1	Phone Call	22
6.2	Welcome message	22
6.3	IVR Menu	23
6.4	Academics Menu	23
6.5	Order Certificate Menu	24
6.6	Query Route	24
6.7	Attendance Details	25
6.8	GPA Percentage	25
6.9	Fee Details	26
6.10	Payment Option	26
6.11	Order Confirmation	27

LIST OF TABLES

Table No.	Name Of the Table	Page No.
2.1	Table Of Existing System	5

TABLE OF CONTENTS

CONTENTS	PAGE NO.
Certificate	i
Declaration	ii
Acknowledgment	iii
Vision & Mission	iv
PEOs, POs & PSOs	v
Course Objectives & Outcomes	viii
Abstract	ix
List of figures	x
List of Tables	xi
Chapter 1: Introduction	
1.2 Problem Statement	2
1.3 Objectives	2
1.4 Motivation	3
1.5 Existing System	4
1.6 Proposed System	4
1.7 Scope Of the Project	4
Chapter 2: Literature Review	5
2.1 Proposed System	6
2.2 Algorithms	8
2.2.1 Dual Tone Multi Frequency	8
2.2.2 Text To Speech and Vice versa	8
2.2.3 Natural Language Processing(NLP)	8
Chapter 3: System Requirements and Specifications	
3.1 Software Requirements	9
3.2 Hardware Requirements	9

Chapter 4: System Design	
4.1 System Architecture Or Block Diagram	10
4.2 Uml Diagrams	11
4.2.1 Use Case Diagram	12
4.2.2 Class Diagram	13
4.2.3 Deployment Diagram	14
4.2.4 Activity Diagram	15
 Chapter 5: Implementation	
5.1 Environmental Setup	16
5.2 Module Description	21
 Chapter 6: Tests and Results	
6.1 Results	22
 Chapter 7: Conclusion and Future Enhancements	
7.1 Conclusion	28
 Chapter 8: References	29
 Appendix	30

CHAPTER 1

INTRODUCTION

Voice-based delivery of academic records is an innovative and modern approach to accessing educational transcripts and other academic credentials. This technology uses speech recognition software and natural language processing to allow users to retrieve their academic records through voice commands. With this technology, students and graduates can access their academic records at any time and from any location, without the need for complicated login procedures or navigating multiple web pages. This has the potential to save time and improve accessibility for individuals who may have difficulty accessing traditional online or physical records.

Voice-based delivery of academic records also has the potential to streamline administrative processes for educational institutions. With automated delivery of academic records, universities and colleges can reduce administrative workload, increase efficiency, and improve customer satisfaction. Overall, the voice-based delivery of academic records is an exciting development that has the potential to revolutionize the way academic credentials are accessed and managed. Even though the internet has received significant attention in the last few decades, voice is still considered an efficient way of communication between humans-to-humans and human-to-computer. IVR systems can lean toward reduced service costs, increase customer satisfaction and offer new and improved services.

IVR phone systems have become so widely used that you've probably found yourself communicating with a virtual assistant over the phone more times than you can count. Academic records are information and documents about a student's academic career. Using IVRS technology we will deliver different academic certificates to students at their doorsteps in collaboration with respective institutions, boards or universities.

Final year/semester marks Cards, degree/diploma certificates etc., are printed and sent to colleges a few months after students leave the institution. They have to visit the institution again to collect such documents. The application helps students avoid such exercise. The documents could be delivered across India and the delivery address could be different from the one given at the time of admission. The document would be delivered within a maximum of five days based on priority mode.

1.2 PROBLEM STATEMENT

To Build an Interactive Voice Response System (IVRS) based application for ordering delivery of academic certificates and making payment against the charge.

1.3 OBJECTIVES

The objectives of voice-based delivery of academic records are as follows:

Improve accessibility: The primary objective of this technology is to improve accessibility to academic records for students and graduates. By providing a voice-based interface, individuals who may have difficulty accessing traditional online or physical records can easily retrieve their academic credentials.

Increase efficiency: With voice-based delivery, academic records can be retrieved quickly and easily, without the need for complicated login procedures or navigating multiple web pages. This can save time for both students and administrative staff, improving the efficiency of the record retrieval process.

Streamline administrative processes: By automating the delivery of academic records, educational institutions can reduce administrative workload and increase efficiency. This can result in cost savings for the institution and improve customer satisfaction for students and graduates.

Foster innovation: Voice-based delivery of academic records is a new and innovative technology that has the potential to transform the way academic credentials are accessed and managed. By adopting this technology, educational institutions can stay at the forefront of technological advancements and provide a modern and convenient service to their students and graduates.

1.4 MOTIVATION FOR IVR TECHNOLOGY

The motivation for IVR technology in voice-based delivery of academic records is similar to that of IVR in customer service. Specifically, IVR technology can:

Improve accessibility: IVR technology allows students and graduates to access their academic records at any time and from any location using just their voice, without the need for complicated login procedures or navigating multiple web pages. This can improve accessibility and convenience for individuals who may have difficulty accessing traditional online or physical records.

Increase efficiency: IVR technology can automate the delivery of academic records, reducing the need for manual intervention by administrative staff. This can improve efficiency by reducing wait times and freeing up staff to handle more complex tasks.

Enhance security: IVR technology can improve security by using voice recognition technology to authenticate users and ensure the accuracy and security of academic records. This can reduce the risk of identity theft and fraud.

Streamline administrative processes: By automating the delivery of academic records, educational institutions can reduce administrative workload and increase efficiency. This can result in cost savings for the institution and improve customer satisfaction for students and graduates.

Foster innovation: IVR technology is an innovative and modern approach to accessing academic records, which can help educational institutions stay at the forefront of technological advancements and provide a modern and convenient service to their students and graduates.

1.5 EXISTING SYSTEM

Final year/semester marks Cards, degree/diploma certificates etc., are printed and sent to colleges a few months after students leave the institution. They have to visit the institution again to collect such documents. In most of the Universities the certificates are applied or information related to academics is collected through their official websites.

The IVR system services are used in diverse domain including telecommunication, insurance, utilities, entertainment, manufacturing, financing, utilities, agriculture, etc. Speech Recognition and other machine learning models are intergrated with new IVR systems

1.6 PROPOSED SYSTEM

Proposed System receives calls automatically from user and it will provide welcome tone. The user will be asked to verify his institutional details. After confirming their identity the user will be directed to an ivr menu. Based on the user input interpreted either thorough speech recognition or dual tone multi frequency (DTMF) tones input provided via keypad, routes them to the appropriate section. This system provides dynamic information related to academics and allows them to order delivery of academic certificates and making payment against the charge.

1.7 SCOPE OF THE PROJECT

The scope of voice-based delivery of academic records is focused on improving accessibility, efficiency, security, and innovation in the management and delivery of academic credentials.

CHAPTER 2

LITERATURE SURVEY

Sl No.	Author's Name & Year	Year of Publication	Objective	Drawbacks
1.	Olawande Daramo..et al	2018	Comparative analysis and review of interactive voice response systems	Not able to capture user requirements and presents a rigid interface to users
2.	Polyna Khudyakov..et al	2018	Designing a Call Center with IVR	<ul style="list-style-type: none"> ● Less number of trunk lines ● Performance drops under heavy traffic
3.	Priscilla Manatsa	2019	Impact of IVR in Canadian Banking Industry	Less awareness about the voice-technology, therefore not well received by clients
4.	Ruth Choy..et al	2022	Personalized Digital Customer Services for Consumer Banking Call-Centre using Neural Networks	<ul style="list-style-type: none"> ● Poor call containment ● Requires an improvement in pattern recognition
5.	Esra Ben Ismail..et al	2022	IVR System use by patients with Heart failure	Self Reporting nature of IVR registered a lot of spam records.

2.1 PROPOSED SYSTEM

IVR BASED SOLUTION

IVR (Interactive Voice Response) is a technology that allows users to interact with a computerized system via voice commands or keypad inputs. IVR systems are commonly used in a variety of settings, including customer service, sales, and education. When it comes to delivering academic records, an IVR system can be used to provide students with a quick and convenient way to access their academic records over the phone. Here's how it works:

The student dials a phone number provided by the school or university. The IVR system answers the call and presents the student with a menu of options to choose from. For example, the student might be asked to press "1" for transcripts, "2" for diploma verification, or "3" for enrollment verification. Once the student selects an option, the IVR system may prompt the student to enter additional information, such as their student ID number or date of birth. Based on the information provided, the IVR system can then retrieve the student's academic records and provide them to the student over the phone. For example, if the student requested transcripts, the IVR system could read out the courses they took, their grades, and any other relevant information.

Alternatively, the IVR system might inform the student that their request cannot be fulfilled over the phone and provide instructions on how to obtain their records in person or through an online portal. Using an IVR system to deliver academic records has several benefits. First, it provides students with a quick and easy way to access their records without having to navigate a complex online portal or wait for a paper copy to be mailed to them. Second, it can help reduce the workload on staff members who would otherwise need to handle these requests manually. Finally, it can improve the overall efficiency of the academic record delivery process, ensuring that students receive their records in a timely manner.

Implementing an IVR system for the voice-based delivery of academic records requires careful consideration of several key points to ensure that the system is effective and meets the needs of the user. Here are some points to consider:

Identify the user needs: Determine the needs and preferences of the users, such as the types of records they need access to, the information they require, and the reasons they may be calling. This information can help to design an IVR system that is tailored to the specific needs of the users.

Define the scope: Define the scope of the IVR system, including the types of records that will be available, the types of requests that can be fulfilled, and any limitations on the system. Ensure that the system meets the relevant regulatory and compliance requirements for academic records.

Develop clear prompts: Develop clear prompts for the IVR system that are easy to understand and navigate. The prompts should be simple and straightforward, with clear instructions on how to access the records and what information is required.

Design an intuitive menu: Design an intuitive menu that is easy to navigate and reflects the needs of the users. Consider using a hierarchical structure that allows users to drill down to the specific record they need.

Optimize the voice recognition: Optimize the voice recognition technology to ensure that it accurately understands and responds to user input. This may involve tuning the voice recognition algorithms or training the system on a specific vocabulary.

Ensure scalability: Ensure that the system is scalable to accommodate future growth and usage. The system should be designed to handle a large volume of requests and users, and should be able to expand as needed.

Provide live assistance: Provide users with the option to speak with a live representative if they encounter any issues with the IVR system. This can help to ensure that users receive the help they need and can reduce frustration and negative experiences.

By considering these points, an IVR system for the voice-based delivery of academic records can be successfully implemented and provide a valuable resource for students and staff.

2.2 ALGORITHMS

2.2.1.DUAL TONE MULTI-FREQUENCY (DTMF)

Dual-Tone Multi-Frequency (DTMF) is a method of encoding and decoding the digit signals transmitted by a touch-tone telephone keypad. When a key on the telephone keypad is pressed, it generates two simultaneous sinusoidal tones of specific frequencies, one from a high-frequency group and one from a low-frequency group. These frequencies are then decoded by the receiving system to determine which key was pressed. DTMF is widely used in interactive voice response (IVR) systems and telephone networks for automated communication with customers or clients .

2.2.2 TEXT TO SPEECH AND SPEECH TO TEXT

Text-to-speech (TTS) technology converts written text into spoken words. This technology is useful in situations where the user needs to hear a piece of text, such as for people who are visually impaired or for use in automated customer service systems. TTS technology works by processing text input and synthesizing it into audible speech output. TTS systems use various techniques such as concatenate synthesis, format synthesis, and neural TTS to generate speech.

2.2.3 NATURAL LANGUAGE PROCESSING (NLP)

Natural Language Processing (NLP) is a sub-field of artificial intelligence and computer science that focuses on the interaction between computers and human language. It involves developing algorithms and computational models that enable computers to understand, interpret, and generate natural language. NLP encompasses a wide range of tasks, including language understanding, language generation, machine translation, sentiment analysis, named entity recognition, speech recognition, and text summarization.

CHAPTER 3

SYSTEM REQUIREMENTS SPECIFICATIONS

3.1 Software Requirements:

- PYHTON : 3.6
- Operating systems: Windows 7 or higher with the latest updates installed.
- Anaconda : Latest Version imported with required libraries.
- VS Code Editor
- Twilio Account

3.2 Hardware Requirements:

- CPU : I5 9th Gen
- RAM : 8 GB
- Hard Disk : 500GB
- Keypad Phone or Any Phone

CHAPTER 4

SYSTEM DESIGN

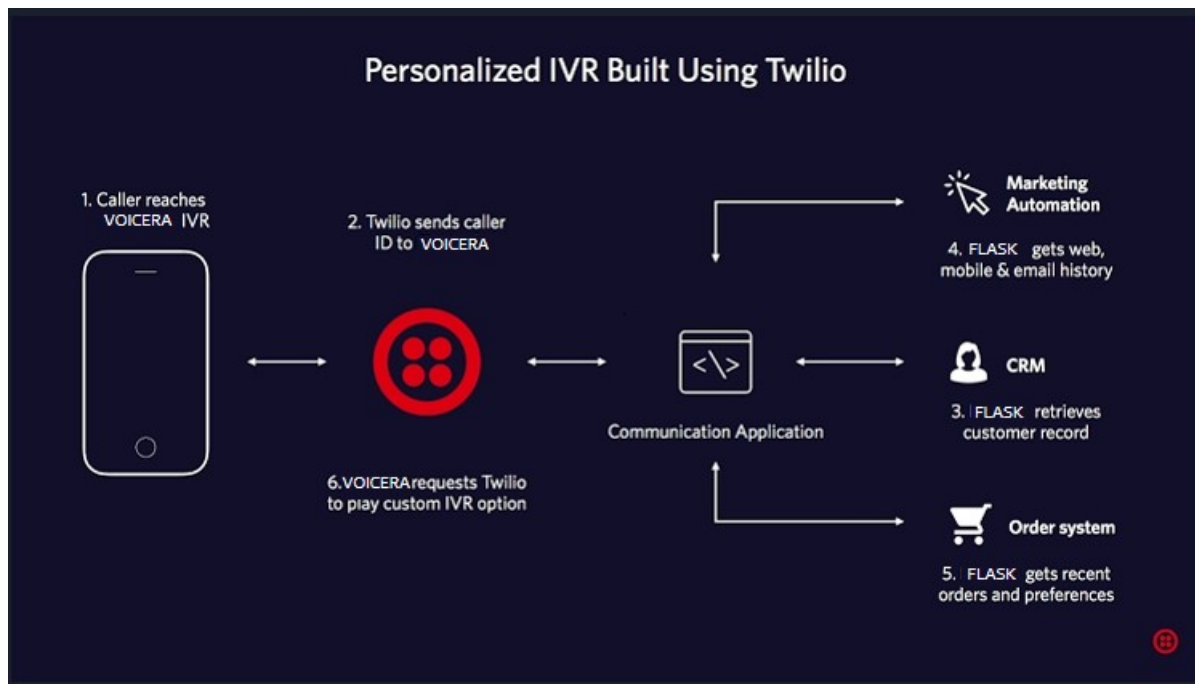


Fig 4.1: Architecture

- The student dials a phone number provided by the school or university.
- The IVR system answers the call and presents the student with a menu of options to choose from. For example, the student might be asked to press "1" for transcripts, "2" for diploma verification, or "3" for enrolment verification. □
- Once the student selects an option, the IVR system may prompt the student to enter additional information, such as their student ID number or date of birth. □
- Based on the information provided, the IVR system can then retrieve the student's academic records and provide them to the student over the phone. For example, if the student requested transcripts, the IVR system could read out the courses they took, their grades, and any other relevant information. □
- Alternatively, the IVR system might inform the student that their request cannot be fulfilled over the phone and provide instructions on how to obtain their records in person.

4.2 UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object-oriented computer software. In its current form UML is comprised of two major components: A Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software systems, as well as for business modeling and other non- software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects-oriented software and the software development process. UML uses mostly graphical notations to express the design of software projects.

4.2.1 USE CASE DIAGRAM

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases.

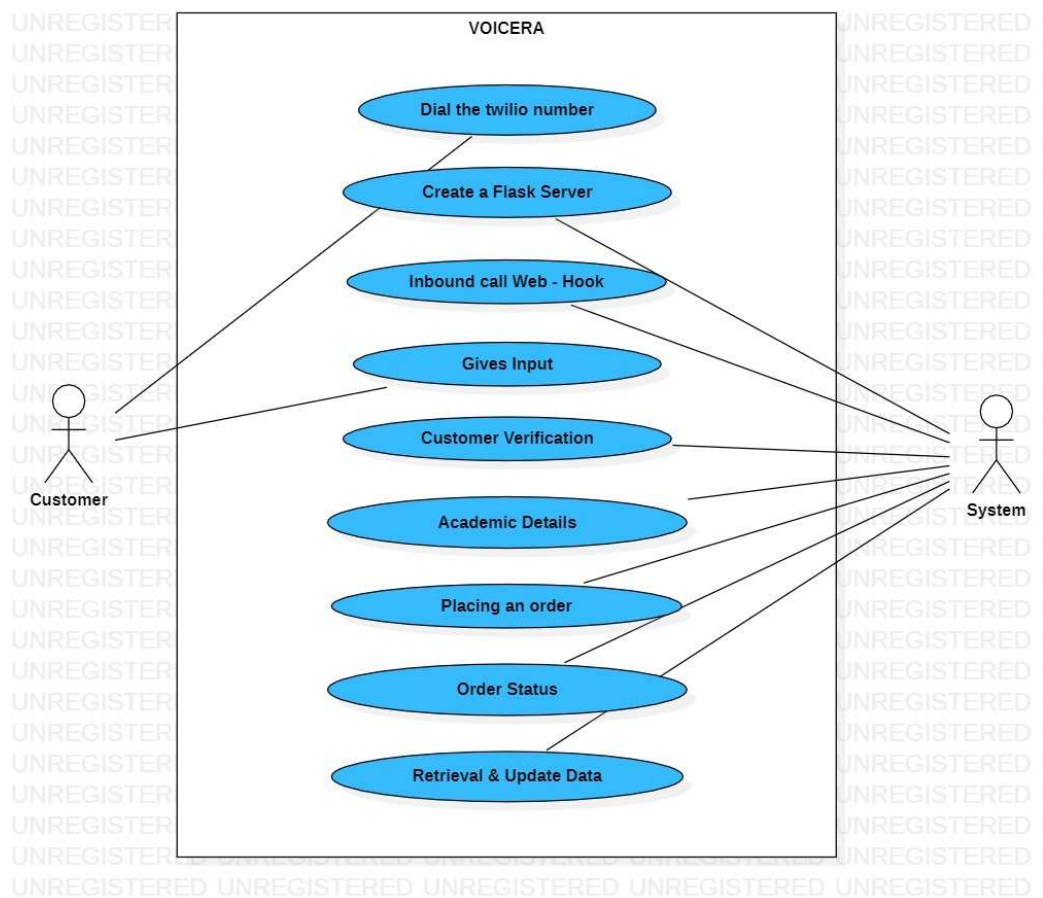


Fig 4.2: Use Case Diagram

4.2.2 CLASS DIAGRAM

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains which information.

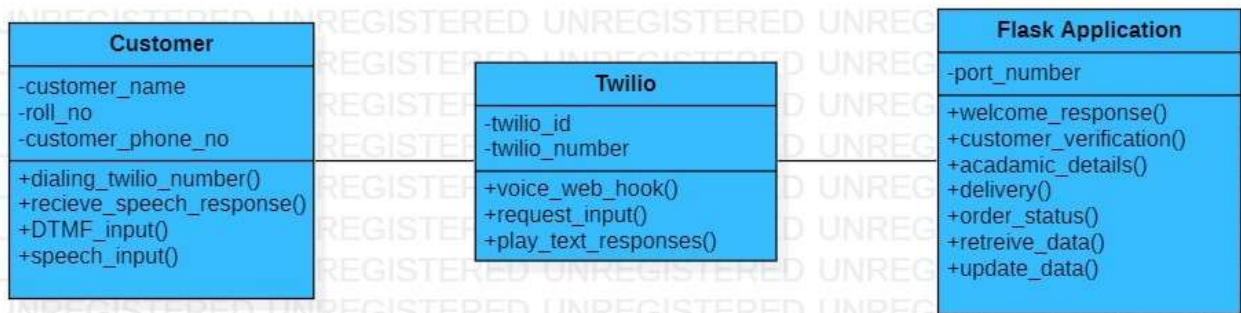


Fig 4.3: Class Diagram

4.2.3 SEQUENCE DIAGRAM:

A sequence diagram in Unified Modelling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

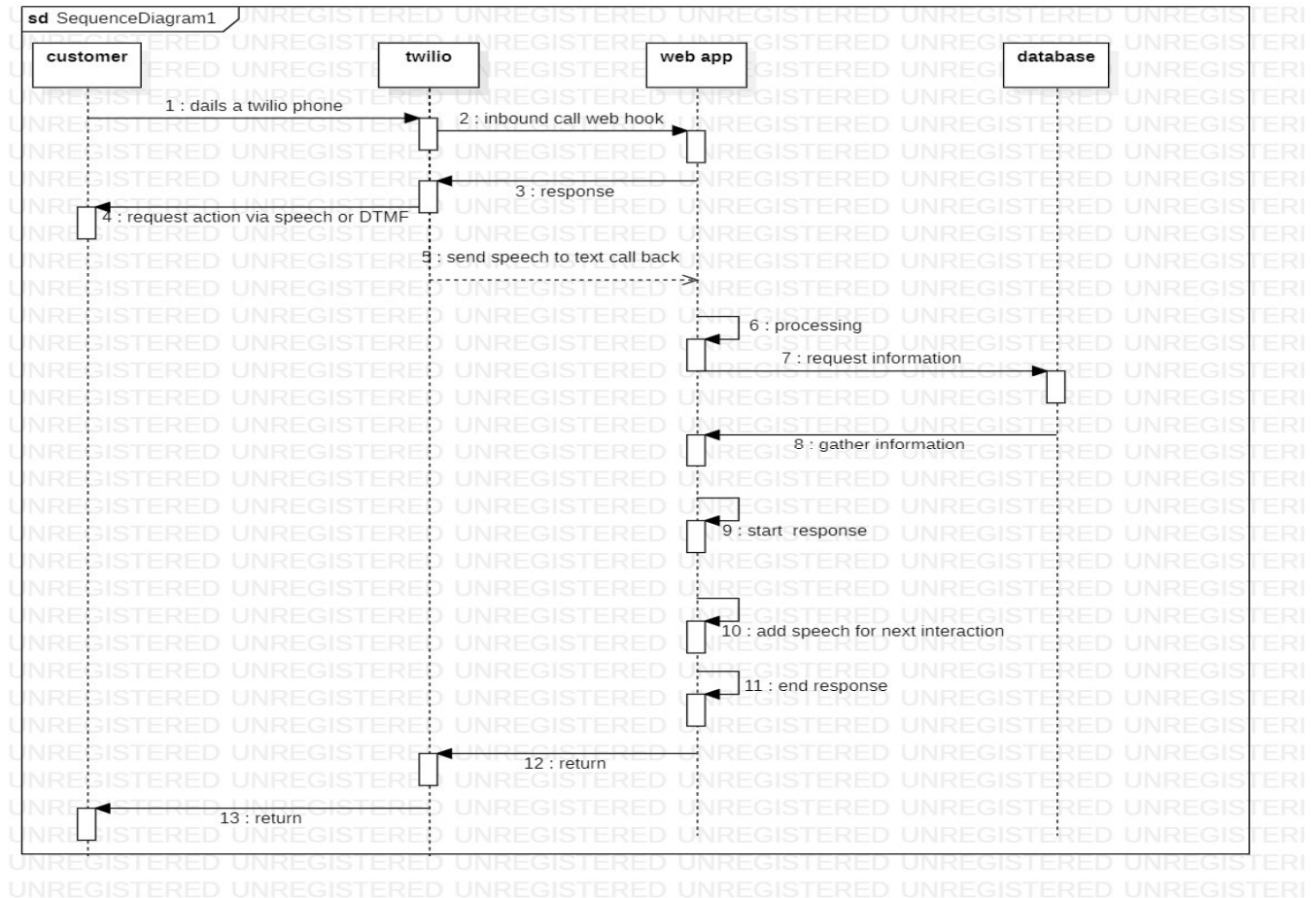


Fig 4.4: Sequence Diagram

4.2.4 Activity diagram:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration, and concurrency. In the Unified Modelling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

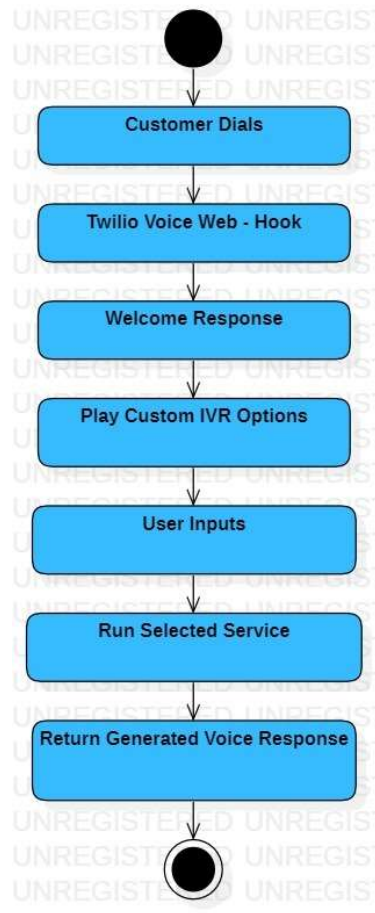


Fig 4.5: Activity Diagram

CHAPTER 5 IMPLEMENTATION

5.1 ENVIRONMENTAL SETUP

5.1.1 Installing VS Code :

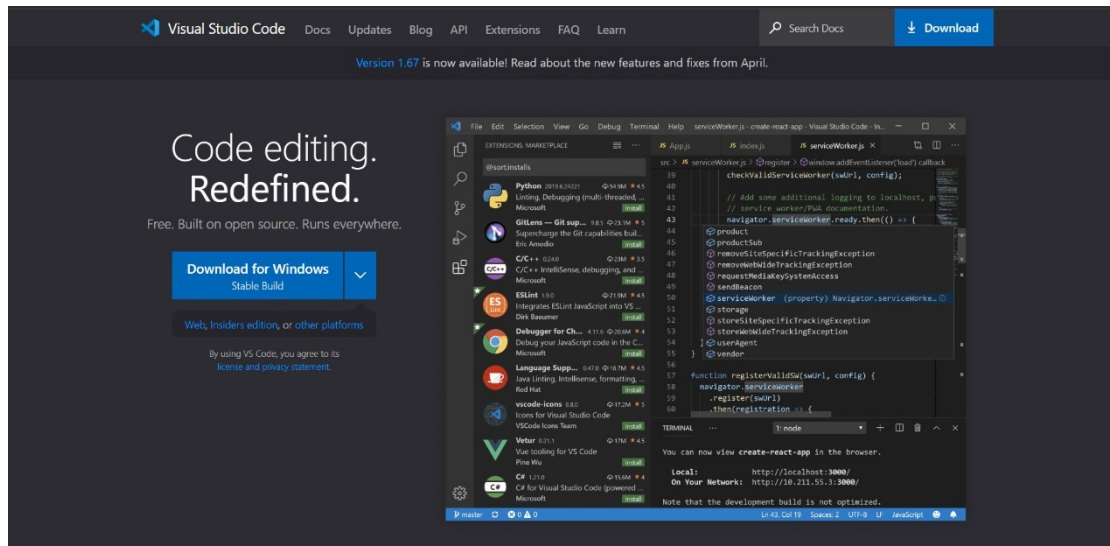


Figure 5.1: VS Code

- 1 Go to the official Visual Studio Code website at <https://code.visualstudio.com/>.
- 2 Click on the "Download" button located in the middle of the page.
- 3 Select your operating system from the available options. Visual Studio Code is available for Windows, macOS, and Linux.
- 4 Once you have selected your operating system, the download will begin automatically. If it doesn't, click the "Download" button.
- 5 Once the download is complete, open the installer. Follow the prompts in the installer to complete the installation process.
- 6 After the installation is complete, launch Visual Studio Code by clicking on the icon in your Applications folder or Start menu.
- 7 Congratulations, you have now installed Visual Studio Code!

5.1.2 Installing Anaconda :

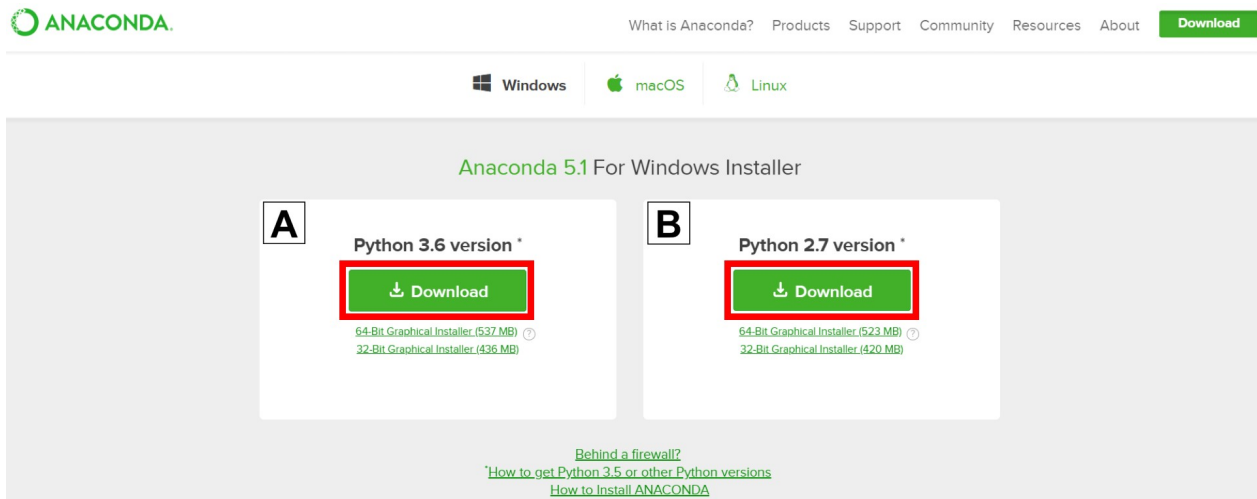


Figure 5.2: Anaconda

- 1 Go to the official Anaconda website at <https://www.anaconda.com/products/individual>.
- 2 Click on the "Download" button for your operating system. Anaconda supports Windows, macOS, and Linux.
- 3 Once the download is complete, launch the installer by double-clicking the downloaded file.
- 4 Follow the prompts in the installer to complete the installation process. You can choose the default settings or customize the installation to suit your needs.
- 5 Once the installation is complete, open Anaconda Navigator from your Applications folder or Start menu.
- 6 The first time you launch Anaconda Navigator, it will take a few minutes to set up your environment. This is a one-time process, so be patient.
- 7 Once the setup process is complete, you can start using Anaconda Navigator to manage your Python environments, install packages, and launch Jupyter notebooks.
- 8 Congratulations, you have now installed Anaconda!

5.1.3 Installing Mongo DB:

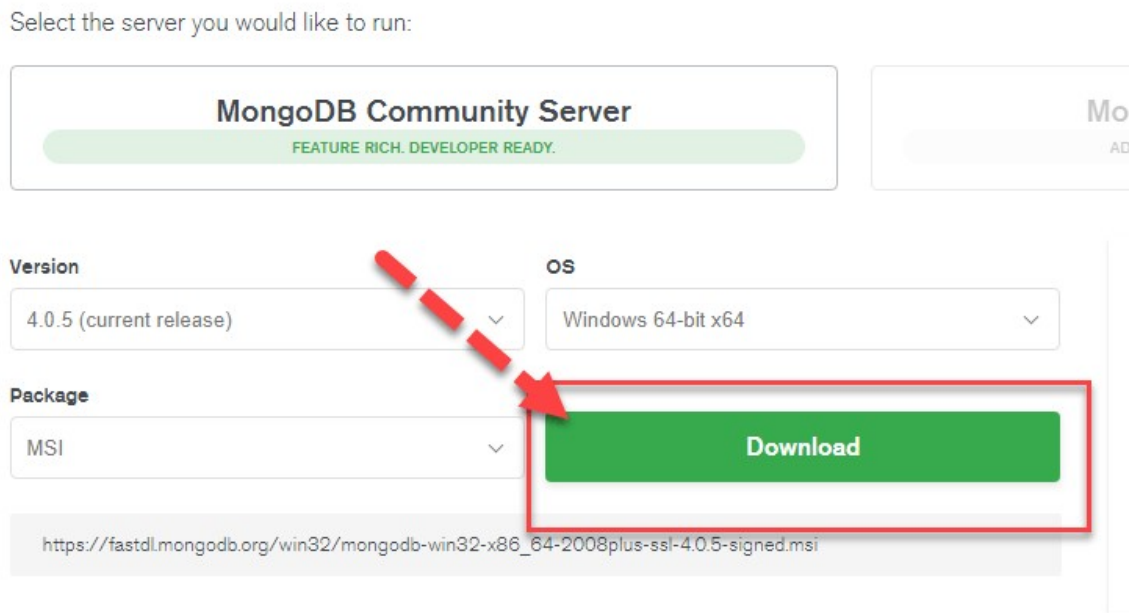


Figure 5.3: Mongo DB

- 1 Go to the official MongoDB website at <https://www.mongodb.com/try/download/community>.
- 2 Scroll down to the "Community Server" section and click on the "Download" button for your operating system.
- 3 Once the download is complete, launch the installer by double-clicking the downloaded file.
- 4 Follow the prompts in the installer to complete the installation process. You can choose the default settings or customize the installation to suit your needs.
- 5 Once the installation is complete, open the Command Prompt or PowerShell on your machine.
- 6 Navigate to the "bin" directory where MongoDB is installed. This is typically "C:\Program Files\MongoDB\Server{version}\bin".
- 7 Type "mongod" to start the MongoDB server.
- 8 MongoDB should now be up and running on your machine!
- 9 Congratulations, you have now installed MongoDB!

5.1.4 Setup Twilio:

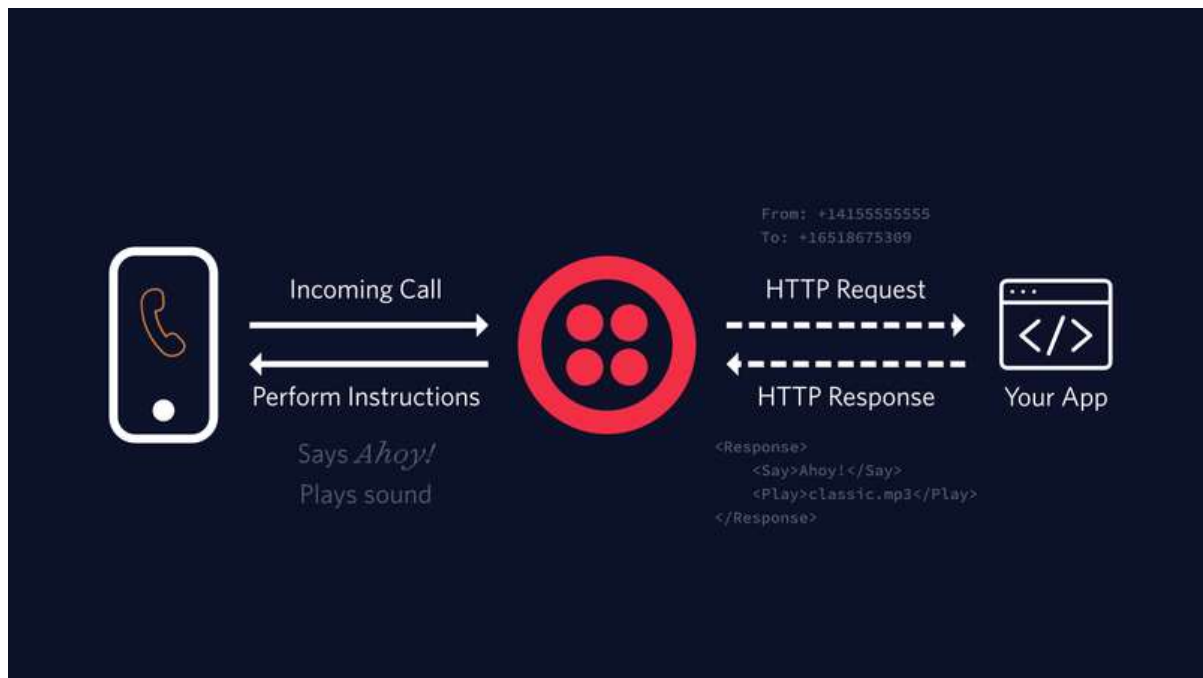
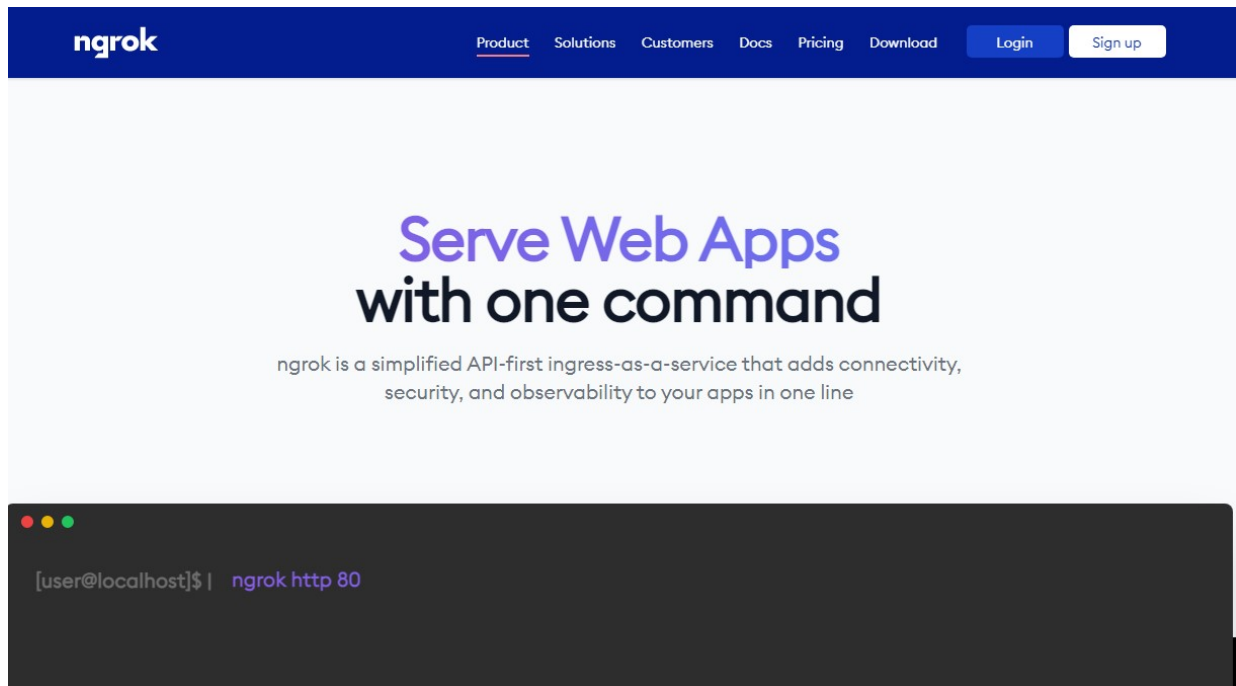


Figure 5.4: Twilio

- 1 Sign up for a Twilio account at <https://www.twilio.com/try-twilio>.
- 2 Once you have signed up, log in to your Twilio account.
- 3 Click on the "Phone Numbers" tab on the left-hand menu.
- 4 Click on the "Buy a Number" button.
- 5 Choose the type of number you want to purchase (local, toll-free, or mobile).
- 6 Select the country and area code for the number you want to purchase.
- 7 Click on the "Search" button to view available numbers.
- 8 Once you have found a number you like, click on the "Buy" button to purchase it.
- 9 After purchasing the number, click on the number in the "Phone Numbers" tab to view its configuration settings.
- 10 Under the "Voice & Fax" section, set the "Configure with" dropdown to "Webhooks/TwiML".
- 11 In the "A Call Comes In" section, enter the URL of your IVR application. This is the URL that Twilio will send incoming calls to when someone dials your Twilio number.
- 12 Save your changes.

5.1.5 Setup ngrok:



- 1 Download and install ngrok: Go to the ngrok website and download the appropriate version of ngrok for your operating system. Once downloaded, extract the files from the zip archive to a folder on your computer.
- 2 Open a terminal or command prompt: Once ngrok is installed, open a terminal or command prompt on your computer.
- 3 Navigate to the ngrok directory: In the terminal or command prompt, navigate to the directory where you extracted ngrok. Authenticate ngrok: If you haven't already done so, you will need to create a free account on the ngrok website.
- 4 Once you have an account, copy the auth token from the ngrok dashboard and run the following command in the terminal or command prompt: `./ngrok authtoken <your-auth-token>` Replace `<your-auth-token>` with your actual auth token.
- 5 Start ngrok: To start ngrok, run the following command in the terminal or command prompt: `./ngrok http <port-number>` Replace `<port-number>` with the port number of the local server you want to expose to the internet.
- 6 Access your local server: Once ngrok is running, it will create a unique URL that you can use to access your local server from anywhere in the world. The ngrok URL will look something like this: `http://random-string.ngrok.io` You can copy and paste this URL into a web browser to access your local server.

5.2 MODULE DESCRIPTION

CUSTOMER: This module would include all the functions and data structures needed to manage customer information, such as their name, address, phone number, and order history. This could include functions for adding new customers, updating existing customer information, and retrieving customer data.

TWILIO: This module would include the functions and data structures needed to interact with the Twilio API, which would be used to manage incoming and outgoing phone calls and text messages. This could include functions for initiating calls and sending text messages, as well as functions for handling incoming calls and messages and responding appropriately.

FLASK: This module would include the functions and data structures needed to build a web application using the Flask web framework. This could include functions for defining routes, rendering templates, and handling user input. Flask would be used to provide a user interface for managing customer data and initiating phone calls and text messages using the Twilio API.

CHAPTER 6

RESULTS(SCREENSHOTS)

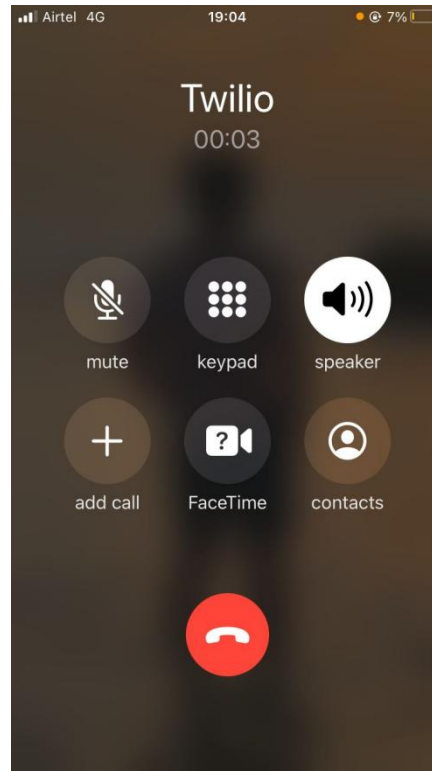


Figure 6.1: Phone Call

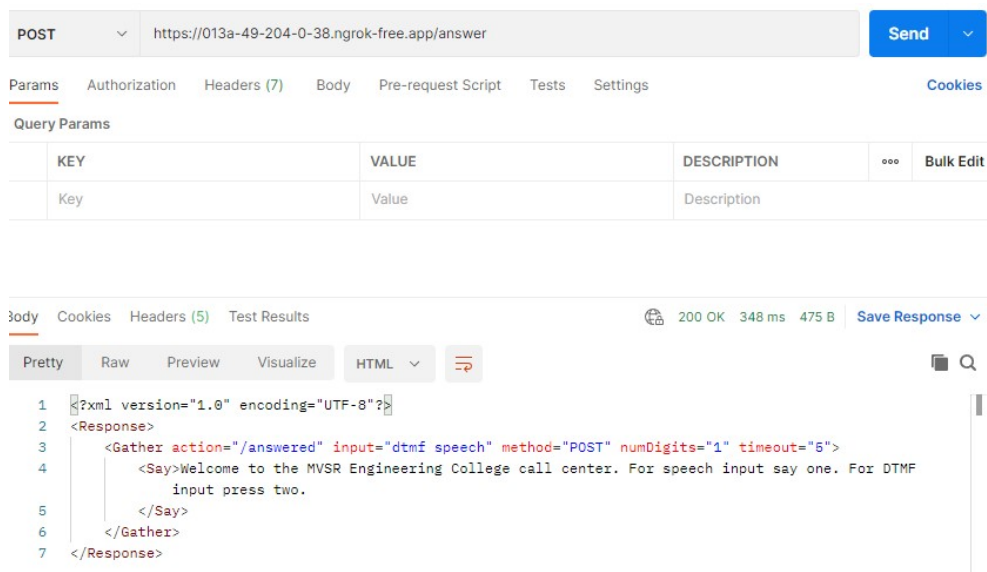


Figure 6.2: Welcome message

POST ▼ https://013a-49-204-0-38.ngrok-free.app/process_dtmf?Digits=245119737040 Send ▼

Params ● Authorization Headers (7) Body Pre-request Script Tests Settings Cookie

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Ed
<input checked="" type="checkbox"/>	Digits	245119737040			
	Key	Value	Description		

Body Cookies Headers (5) Test Results 200 OK 351 ms 454 B Save Response ▼

Pretty Raw Preview Visualize HTML ▼ 🔍

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <Response>
3   <Say>Roll number verified. Press 1 to know your academic details, Press 2 to order your certificates,
4     Press 3 for
5     any other queries</Say>
6   <Gather action="/menu" method="POST" numDigits="1" />
7 </Response>

```

Figure 6.3: IVR Menu

<https://013a-49-204-0-38.ngrok-free.app/menu?Digits=1> Save ▼ 🔍

POST ▼ <https://013a-49-204-0-38.ngrok-free.app/menu?Digits=1> Send ▼

Params ● Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	Digits	1			
	Key	Value	Description		

Body Cookies Headers (5) Test Results 200 OK 352 ms 517 B Save Response ▼

Pretty Raw Preview Visualize HTML ▼ 🔍

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <Response>
3   <Gather action="/academic_menu" input="dtmf" method="POST" numDigits="1" timeout="5">
4     <Say>To know your attendance percentage, Press 1. To know your exam results, Press 2. To know
5       your timetable,
6       Press 3. To know your fee details, Press 4.</Say>
7   </Gather>
8 </Response>

```

Figure 6.4: Academics menu

https://013a-49-204-0-38.ngrok-free.app/menu?Digits=2

Save

POST https://013a-49-204-0-38.ngrok-free.app/menu?Digits=2 Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	Digits	2			
	Key	Value	Description		

Body Cookies Headers (5) Test Results 200 OK 363 ms 444 B Save Response

Pretty Raw Preview Visualize HTML

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <Response>
3   <Gather action="/order_certificate" input="speech" timeout="8">
4     <Say>To order Transcript, Press 1. To order Degree Certificate, Press 2. To Order Mark Sheet,
      Press 3.</Say>
5   </Gather>
6 </Response>

```

Figure 6.5: Order Certificate Menu

https://013a-49-204-0-38.ngrok-free.app/menu?Digits=3

Save

POST https://013a-49-204-0-38.ngrok-free.app/menu?Digits=3 Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	Digits	3			
	Key	Value	Description		

Body Cookies Headers (5) Test Results 200 OK 361 ms 364 B Save Response

Pretty Raw Preview Visualize HTML

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <Response>
3   <Gather action="/query-speech" input="speech" timeout="3">
4     <Say>Please say your query.</Say>
5   </Gather>
6 </Response>

```

Figure 6.6: Query Route

https://013a-49-204-0-38.ngrok-free.app/academic_menu?Digits=1

POST https://013a-49-204-0-38.ngrok-free.app/academic_menu?Digits=1

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	Digits	1			
	Key	Value	Description		

Body Cookies Headers (5) Test Results 200 OK 360 ms 300 B Save Response

Pretty Raw Preview Visualize HTML

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <Response>
3   <Say>Attendace Percentage is 58</Say>
4 </Response>

```

Figure 6.7: Attendace Details

https://013a-49-204-0-38.ngrok-free.app/academic_menu?Digits=2

POST https://013a-49-204-0-38.ngrok-free.app/academic_menu?Digits=2

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	Digits	2			
	Key	Value	Description		

Body Cookies Headers (5) Test Results 200 OK 355 ms 310 B Save Response

Pretty Raw Preview Visualize HTML

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <Response>
3   <Say>GPA Percentage is 5.596156793438814</Say>
4 </Response>

```

Figure 6.8: GPA Percentage

https://013a-49-204-0-38.ngrok-free.app/academic_menu?Digits=4

Save

POST https://013a-49-204-0-38.ngrok-free.app/academic_menu?Digits=4 Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
<input checked="" type="checkbox"/>	Digits	4			
	Key	Value	Description		

Body Cookies Headers (5) Test Results 200 OK 360 ms 294 B Save Response

Pretty Raw Preview Visualize HTML

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <Response>
3   <Say>Remaing Due is 62000</Say>
4 </Response>

```

Figure 6.9: Fee Details

https://013a-49-204-0-38.ngrok-free.app/order_certificate

Save

POST https://013a-49-204-0-38.ngrok-free.app/order_certificate Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

	KEY	VALUE	DESCRIPTION	...	Bulk Edit
	Key	Value	Description		

Body Cookies Headers (5) Test Results 200 OK 352 ms 330 B Save Response

Pretty Raw Preview Visualize HTML

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <Response>
3   <Say>Press 1 for Card Payment. Press 2 for cash on delivery.</Say>
4 </Response>

```

Figure 6.10: Payment Option

POST https://013a-49-204-0-38.ngrok-free.app/make_payment

Save Send

POST https://013a-49-204-0-38.ngrok-free.app/make_payment

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (5) Test Results 200 OK 355 ms 401 B Save Response

Pretty Raw Preview Visualize HTML

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <Response>
3   <Say>Thank you for your payment. Your certificate will be delivered to your registered address within
4     7-10 business
5     days.</Say>
6   <Hangup />
7 </Response>

```

Figure 6.11: Order Confirmation

CHAPTER 7

CONCLUSION

Voice-based delivery of academic records using Interactive Voice Response (IVR) is a promising solution that offers several benefits to both students and educational institutions. It provides a convenient and accessible way for students to access their academic records through a simple phone call, without the need for an internet connection or specialized hardware. voice-based delivery of academic records using IVR has the potential to revolutionize the way students access and manage their academic records, and educational institutions should consider implementing this technology to improve the efficiency and accessibility of their record management systems.

IVR systems can also provide an additional layer of security for academic records, as students would need to provide some form of identification to access their records. This can help to prevent unauthorized access to academic records and protect the privacy and confidentiality of student information.

Overall, an IVR system can be an effective way to provide students with easy and secure access to their academic records. By automating the delivery of academic certificates and academic details, academic institutions can save time and resources while providing a convenient service to their students.

CHAPTER 8

REFERENCES

- [1] S.Uma Maheshwari et al., Interactive Voice Response System Development, International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-8, Issue-2S, December 2018
- [2] Itorobong A. Inam et al., Comparative Analysis and Review of Interactive Voice Response Systems, Conference on Information Communications Technology and Society, 2018
- [3] Samarth Agarwal et al., Personalized Digital Customer Services for Consumer Banking Call Centre using Neural Networks, University of Western Ontario, 2020
- [4] Polyna Khudyakov et al., Designing a call center with an IVR (Interactive Voice Response), 2018
- [5] Esra Ben Ismail et al., IVR System Use by Patients with Heart Failure: Compliance and Services Utilization Patterns, Journal of Medical Systems (2022)
- [6] Priscilla Manatsa, An Analysis of the impact of implementing a new Interactive Voice Response System (IVR) on Client experience in the canadian Banking Industry, International Economics and Finance, Ryerson University, 2019
- [7] Next Generation IVR Using Twilio Speech Recognition and Chatbots
<https://www.linkedin.com/pulse/next-generation-ivr-using-twilio-speech-recognition-chatbots-badri/?trk=v-feed>
- [8] Interactive voice response https://en.wikipedia.org/wiki/Interactive_voice_response

APPENDIX

CODE :

```
from flask import Flask, request
from twilio.twiml.voice_response import VoiceResponse, Gather
import pandas as pd
from pymongo import MongoClient
import json

client = MongoClient("mongodb://admin-hrutu:Fire%401234@cluster0.vwlmw.mongodb.net/")
db = client["mvsr"]
data = db["data"]

app = Flask(__name__)

@app.route("/")
def hello():
    return "Hello World!"

@app.route("/answer", methods=['GET', 'POST'])
def answer_call():
    response = VoiceResponse()
    gather = Gather(input='dtmf speech', timeout=5, num_digits=1, action='/roll', method='POST')
    gather.say('Welcome to the MVSR Engineering College call center. For speech input say one. For DTMF input press two.')
    response.append(gather)
    return str(response)
    # Process the user's choice

@app.route("/roll", methods=['POST'])
def answered():
    response = VoiceResponse()
    if 'Digits' in request.form:
        # If the user chose DTMF input
        gather = Gather(input='dtmf', timeout=3, num_digits=12, action='/process_dtmf', method='POST')
        gather.say('Please enter your Roll number using the keypad.')
        response.append(gather)
    elif 'SpeechResult' in request.form:
        # If the user chose speech input
        gather = Gather(input='speech', timeout=12, num_digits=12, action='/process_speech', method='POST')
        gather.say('Please say your Roll Number.')
        response.append(gather)
    else:
        # Invalid input
        response.say('Sorry, I did not understand your choice. Please try again.')
        response.redirect('/answer')

    return str(response)
```

```

@app.route('/process_speech', methods=['POST'])
def process_speech():
    roll_number = request.values.get('SpeechResult')
    roll = data.find_one(roll_number)
    # TODO: Add code to verify the roll number
    if (roll):
        response = VoiceResponse()
        response.say("Roll number verified. Press 1 to know your academic details, Press 2 to order your certificates, Press 3 for any other queries")
        response.gather(num_digits=1, action='/menu?rollno='+json.dumps(roll), method='POST')
        return str(response)
    else:
        response = VoiceResponse()
        gather = Gather(input='speech', timeout=6, num_digits=12, action='/process_speech', method='POST')
        gather.say('Invalid roll number. Please say your roll number again.')
        response.append(gather)
        return str(response)

@app.route('/process_dtmf', methods=['POST'])
def process_dtmf():
    roll_number = request.values.get('Digits', None)
    roll = data.find_one(roll_number)

    if (roll):
        response = VoiceResponse()
        response.say("Roll number verified. Press 1 to know your academic details, Press 2 to order your certificates, Press 3 for any other queries")
        response.gather(num_digits=1, action='/menu?rollno='+json.dumps(roll), method='POST')
        return str(response)

    else:
        response = VoiceResponse()
        response.say("Invalid roll number. Please try again.")
        response.gather(num_digits=12, action='/process_dtmf', method='POST')
        return str(response)

@app.route("/menu", methods=['GET', 'POST'])
def menu():
    digit_pressed = request.values.get('Digits', None)
    roll = request.values.get('rollno')
    rollno = json.loads(roll)
    if digit_pressed == '1':
        response = VoiceResponse()
        gather = Gather(input='dtmf', timeout=5, num_digits=1, action='/academic_menu?rollno='+roll, method='POST')
        gather.say("To know your attendance percentage, Press 1. To know your exam results, Press 2. To know your timetable, Press 3. To know your fee details, Press 4.")
        response.append(gather)
        return str(response)

    elif digit_pressed == '2':
        response = VoiceResponse()
        gather = Gather(input='dtmf', timeout=8, num_digits=1, action='/certificate_menu?rollno='+roll, method='POST')
        gather.say("To order a transcript, press 1. To order a degree certificate, press 2. To order a mark sheet, press 3.")

```

```

        response.append(gather)
        return str(response)
    elif digit_pressed == '3':
        response = VoiceResponse()
        gather = Gather(input='speech', timeout=3, action='/query-speech')
        gather.say('Please say your query.')
        response.append(gather)
        return str(response)
    else:
        response = VoiceResponse()
        response.say("Invalid selection. Please try again.")
        response.gather(num_digits=1, action='/menu?rollno='+roll, method='POST')
        return str(response)

```

```
@app.route("/academic_menu", methods=['GET', 'POST'])
```

```

def academic_menu():
    digit_pressed = request.values.get('Digits', None)
    roll = request.values.get('rollno')
    rollno = json.loads(roll)
    if digit_pressed == '1':
        response = VoiceResponse()
        att = rollno.attendance
        response.say("Attendace Percentage is "+ str(att))
        return str(response)
    elif digit_pressed == '2':
        response = VoiceResponse()
        res = rollno.gpa
        response.say("GPA Percentage is "+ str(res))
        return str(response)
    elif digit_pressed == '3':
        response = VoiceResponse()
        response.say("Not Avialabel")
        return str(response)
    elif digit_pressed == '4':
        response = VoiceResponse()
        res = rollno.fee
        response.say("Remaing Due is "+ str(res))
    else:
        response = VoiceResponse()
        response.say("Invalid selection. Please try again.")
        response.gather(num_digits=1, action='/academic_menu?rollno='+roll, method='POST')
        return str(response)

```

```
@app.route('/query-speech', methods=['POST'])
```

```

def query():
    # Get the speech input from the request parameters
    speech_result = request.form['SpeechResult']

    # Process the speech input as per your requirements

    return 'Speech input processed: ' + speech_result

```

```
@app.route('/address-speech', methods=['POST'])
```

```
def address():
```

```

# Get the speech input from the request parameters
speech_result = request.form['SpeechResult']
response = VoiceResponse()
gather = Gather(num_digits=1, timeout=7, action='/order_certificate', method='POST')
gather.say('Speech input processed: ' + speech_result + 'is confirmed.')
response.append(gather)
return str(response)

@app.route("/order_certificate", methods=['GET', 'POST'])
def order_certificate():
    student_id = request.values.get('Digits', None)
    # TODO: Add code to retrieve student's information and process the certificate order
    response = VoiceResponse()
    response.say("Thank you for your order. Your certificate will be delivered to your registered address within 7-10 business days.")
    response.hangup()
    return str(response)

def make_payment():
    credit_card_number = request.values.get('Digits', None)
    # TODO: Add code to process the payment
    response = VoiceResponse()
    response.say("Thank you for your payment. Your certificate will be delivered to your registered address within 7-10 business days.")
    response.hangup()
    return str(response)

if __name__ == "__main__":
    app.run(debug=True)

```