EVA : EDUCATIONAL VOICE ASSISSTANT

A PROJECT REPORT

submitted by

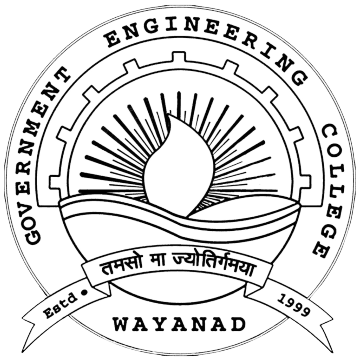
DIYA BAIJU N(WYD20CS027)
ADITH KOLIYOT (WYD20CS004)
AMAAN ZAIN N (WYD20CS012)
AKASH P.T (WYD20CS010)

to

The APJ Abdul Kalam Technological University
in partial fulfilment of the requirements for the award of the Degree

of

Bachelor of Technology
in
Computer Science and Engineering



Department of Computer Science and Engineering

Government Engineering College, Wayanad
Thalappuzha – 670644

AUGUST, 2023

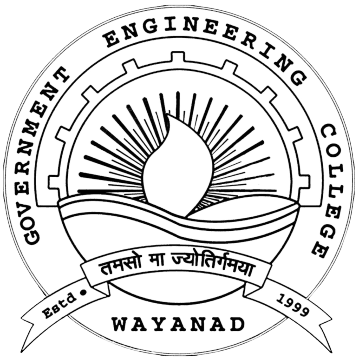
DECLARATION

I, on behalf of authors of the report: Diya Baiju N, Akash P T, Amaan Zain N, Adith Koliyot, hereby declare that the project report " EVA : Educational Voice Assisstant " submitted for partial fulfilment of the requirements for the award of degree of Bachelor of Technology of the APJ Abdul Kalam Technological University , Kerala is a bonafide work done by us under the supervision of Prof. Narasimhan T(Asst.Professor, CSE Department). This submission represents our ideas in our own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that we have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also invoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Place: Thalappuzha
Date: 09-08-2023

Diya Baiju N

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
GOVERNMENT ENGINEERING COLLEGE, WAYANAD
THALAPPUZHA – 670644



CERTIFICATE

This is to certify that the report entitled **" EVA : Educational Voice Assistant "** submitted by **Diya Baiju N(WYD20CS027), Adith Koliyot (WYD20CS004) , Amaan Zain N (WYD20CS012), Akash P.T (WYD20CS010)** to the APJ Abdul Kalam Technological University in partial fulfilment of the requirements for the award of the Degree of Bachelor Technology in Computer Science and Engineering is a bonafide record of the project work carried out by them under our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

Project Supervisor(s)

Department Seal

Project Coordinator(s) Head of the Department

ACKNOWLEDGEMENTS

We are greatly indebted to God Almighty for being the guiding light throughout with His abundant grace and blessings that strengthened us to complete this endeavour with confidence.

We would like to acknowledge Prof. Gilesh M.P, Assistant Professor, Dept. of Computer Science and Engineering, Govt. Engineering College, Wayanad, for inspiring and providing sincere guidance throughout this project.

We also express our sincere gratitude to the project coordinators Ms. Anju K B, Assistant Professor, Dept. of Computer Science and Engineering, Ms. Athulya C Shaji, Assistant Professor, Dept. of Computer Science and Engineering, Govt. Engineering College, Wayanad, for their valuable suggestions and constant encouragements. We are extremely grateful to Dr. Gilesh M P, Professor and Head of the Department, Computer Science and Engineering, Govt. Engineering College, Wayanad for his wise guidance and support.

We would like to express our sincere gratitude to each member of Dept. of Computer Science and Engineering, Govt. Engineering College, Wayanad, for their kind co-operation and encouragement that helped us in completing this project.

We are also extremely grateful to Dr. Jasmin E.A, Principal, for providing all facilities. We would like to extend our gratitude to all well wishers and friends who supported us throughout the course of the project.

ABSTRACT

In today's fast-paced educational environment, the efficient management of academic activities and access to relevant information are crucial for enhancing the overall learning experience. To address this need, we present "EVA - Educational Virtual Assistant," an innovative AI-driven system developed using a Flutter-based Android App integrated with an SQLite foundation. EVA leverages cutting-edge text-to-intent API technology to facilitate spoken audio message inputs and generate contextually relevant output, thereby providing assistance to college students, faculty, and visitors within or connected to the campus ecosystem.

EVA aims to revolutionize the way educational institutions manage and disseminate information, offering a user-friendly and convenient solution to enhance communication and streamline daily tasks. By utilizing spoken audio messages, users can effortlessly interact with the system, enabling a seamless and natural interaction experience. The integration of the SQLite database further empowers EVA to effectively handle data manipulation and storage, ensuring a robust and efficient backend foundation for the AI-driven functionalities.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

In the contemporary educational landscape, the seamless integration of technology has become pivotal in enhancing learning experiences and optimizing administrative processes. As academic institutions strive to provide more efficient and user-friendly solutions, the role of artificial intelligence (AI) and mobile applications has gained prominence. This project introduces "EVA - Educational Virtual Assistant," an innovative AI-powered system designed to cater to the diverse needs of college students, faculty members, and visitors. By harnessing the capabilities of a Flutter-based Android App and leveraging the versatility of SQLite data manipulation, EVA aims to redefine the way individuals interact with their campus environment.

The modern educational ecosystem is characterized by its dynamic nature, with an ever-increasing demand for accessible information and personalized support. Traditional methods of information dissemination and administrative management often fall short of meeting these evolving requirements. EVA addresses these challenges by employing advanced AI technologies, specifically a text-to-intent API, to facilitate natural language interactions through spoken audio messages. This pioneering approach fosters a more intuitive and human-like engagement, enabling users to effortlessly communicate their needs and receive contextually relevant responses.

This project's significance lies in its potential to bridge the gap between technological innovation and the pragmatic requirements of education. By amalgamating AI-driven natural language processing, mobile app development, and database management, EVA

aspires to create a unified and user-centric ecosystem that simplifies daily tasks and enhances overall campus experiences. The subsequent sections delve into the intricacies of EVA's development, functionalities, and the underlying technologies that empower it to serve as an invaluable Educational Virtual Assistant.

1.1 PROBLEM DESCRIPTION

In the realm of higher education, a critical challenge persists: the absence of an efficient and user-friendly system to manage academic activities for college students and faculty, coupled with a streamlined access to comprehensive information about the institution and its facilities. Concurrently, visitors to the campus, including prospective students and guests, struggle to navigate, locate departments, and grasp the array of services available. This results in a disjointed experience and prevents the college from effectively showcasing its offerings.

The existing communication and engagement landscape within these institutions also presents significant gaps. Important announcements, event updates, and policy changes encounter barriers, leading to limited awareness and reduced community cohesion. To address these challenges, the conceptualization and development of EVA - Educational Virtual Assistant - have emerged. The innovative approach aims to enhance academic activity management, offer a unified information hub for visitors, and bridge communication gaps, ultimately fostering an efficient, engaging, and well-informed campus environment.

CHAPTER 2
LITERATURE REVIEW

2.1 VOICE-ENABLED VIRTUAL ASSISTANT

In this study[5] the authors explores the development and implementation of a voice-enabled virtual assistant, shedding light on its significance, design, and potential applications. In an era marked by rapid advancements in technology, the authors delve into the realm of voice interaction, seeking to enhance user experiences through seamless communication with AI-driven virtual assistants. The paper opens by emphasizing the ubiquity of voice-enabled devices and the growing trend towards incorporating voice interactions into various applications. The authors highlight the advantages of voice-based interactions, including hands-free operation and accessibility for individuals with disabilities. The authors describe the integration of automatic speech recognition (ASR) systems, natural language processing (NLP) algorithms, and text-to-speech (TTS) synthesis to create a comprehensive voice interaction platform. This architecture enables users to engage in natural conversations with the virtual assistant, allowing for the extraction of intent from spoken queries and the generation of relevant responses. Furthermore, the paper addresses challenges and future directions in the field of voice-enabled virtual assistants. The authors acknowledge the need for continuous improvements in ASR accuracy, language understanding, and context-awareness to ensure a seamless and efficient user experience. In conclusion, the paper "Voice-Enabled Virtual Assistant" underscores the transformative potential of voice interactions in the realm of virtual assistants. Through a comprehensive exploration of design, implementation, applications, and challenges, the authors provide a holistic view of this burgeoning technology. As voice-enabled devices

become increasingly integrated into our daily lives, the insights presented in this paper serve as a valuable guide for researchers, developers, and practitioners seeking to harness the power of voice interactions to create intuitive, efficient, and user-centric virtual assistant systems.

2.2 INTERNET OF THINGS IN EDUCATION: ARTIFICIAL INTELLIGENCE VOICE ASSISTANT IN THE CLASSROOM

This paper[4] seeks to understand pre-service teachers' perspective on one specific Internet of Things (IoT) device. This paper focuses on Google Assistant. While many classrooms, particularly special education classrooms, have teaching assistants, this paper examine the use of Intelligent Personal Assistants (IPAs) in the classroom to serve as the teaching assistants everyone is familiar with in traditional classroom settings. This paper reports the preliminary data collected on pre-service teachers' perception in integrating voice assistant technology in their current specialized content or their future classrooms.

2.3 INVESTIGATING SIRI AS A VIRTUAL ASSISTANT IN A LEARNING CONTEXT

This paper[1] addresses challenges arising from the use of the conversational interface Siri in pedagogical contexts.

On one hand, it investigate potentialities of Siri as a virtual assistant and knowledge navigator for task accomplishment during classroom activities. Thus, the paper attempt to suggest possible lines of mobilising Siri to afford and improve both task elaboration and language learning. In human-Siri interactions, a wide range of operations are carried

out through closely intertwined oral and written instances of natural language (vocal and visual accounts of 'understanding').

On the other hand, it raises some crucial, critical questions with regard to concrete situations and to prospective pedagogical scenarios involving Siri as a learning assistant. How do students deal with the unpredictability of the interactions with the virtual conversational agent? How do students handle interaction modalities such as identifying and well pronouncing the right commands and the choice of words that activate the diverse features ? This paper seeks to shed light on a valuable use of Siri in school context all the more since teachers and students already rely on mobile device features in quite diverse extracurricular situations.

2.4 SUMMARY OF EXISTING WORKS

Description	Merits	Demerits	Method used
Saint Louis University's QnABot[2] based educational chatbot that supports all modes.	Expand existing and grow new channels. High quality speech recognition and natural language understanding.	Compatibility is a big issue. It only works with paid premium Alexa integrations and only if there is an echo speaker nearby.	Amazon Web Services (aws) Amazon Alexa or Amazon Lex
Bolton College used IBM Watson[3] to build a virtual assistant that enhances teaching, learning, and information access.	Deliver personalized, differentiated, and contextualized responses to each student.	Website-based digital assistant used on the college website.	IBM's Cognitive Platform and Watson Conversation Service.

Table 2.1: Summary of existing works

2.5 MOTIVATION

There is a need for a more efficient and convenient way for college students, faculties, and visitors to manage their academic activities, and a lack of a proper system for accessing information regarding the college and its facilities. The motivation behind developing the EVA project for our college stems from the desire to provide quicker and more effortless assistance to students and faculty for fetching information regarding the college.

2.6 PROBLEM STATEMENT

By creating a user-friendly application and integrating intelligent capabilities, EVA aims to streamline interactions and simplify access to campus resources. With EVA, students can effortlessly navigate through various services, retrieve information, and seek assistance, minimizing effort and maximizing efficiency, to create a more enjoyable college experience.

CHAPTER 3
REQUIREMENT ANALYSIS

3.1 INTRODUCTION

The primary goal of this requirement analysis is to identify and articulate the key functionalities, features, and constraints that the proposed system must address. By gathering and analyzing these requirements thoroughly, we can lay the groundwork for a successful development process that meets the specific needs of all stakeholders while fostering a more convenient and efficient academic environment.

3.1.1 Purpose

A college voice assistant's purpose is to support college students, faculty, and visitors in various aspects of their academic and personal lives. It can serve as a virtual assistant that uses artificial intelligence and natural language processing technologies to answer students' questions and provide guidance on various topics related to the college.

3.2 OVERALL DESCRIPTION

The product perspective of the EVA project is to develop a mobile application that utilizes cutting-edge technology to provide an interactive and conversational interface for users to access information about their college. The product aims to be intuitive, user-friendly, and convenient for users, making it an essential tool for managing their daily activities and staying informed about their college.

3.2.1 User Needs

“EVA" has basically 4 types of users.

- Students - College students

  Characteristics:

  Age group: 18-25 years old

  Familiarity with technology: high

  Goals: To access information about their course schedules, faculty profiles, campus maps, and weather information. They may also use the application to set reminders and receive personalized recommendations based on their preferences.

- Faculty - College faculty members

  Characteristics:

  Age group: 25-60 years old

  Familiarity with technology: moderate to high

  Goals: To access relevant information about their course schedules and departmental events and manage their tasks efficiently.

- Staff - Administrative staff

Characteristics:

Age group: 25-60 years old

Familiarity with technology: moderate to high

Goals: To access information about student schedules, departmental events, and campus maps.

- Visitors

Characteristics:

Age group: varied

Familiarity with technology: varied

Goals: To access information about the college, such as campus maps, directions to specific buildings, information about upcoming events, and other relevant information.

3.2.2 Assumptions and Dependencies

Main Assumptions:

1. Stable internet Connection: The requirement analysis assumes that users of the digital platform will have access to a stable and reliable internet connection. A stable internet connection is essential for seamless access to the platform, smooth data transmission, and real-time updates. Users should be aware that intermittent or slow internet connectivity may hinder their ability to access certain features of the platform effectively.

2. Server reliability: The requirement analysis assumes that the server hosting the digital platform will maintain a high level of uptime and reliability. However, it acknowledges that occasional server downtime or maintenance might occur, and the users should be aware of this possibility. Contingency plans and regular backups should be in place to minimize disruptions and data loss.

3. Responsible usage: It is assumed that the college or platform users will use the collected data only for legitimate academic purposes and will not exploit or share it for any forbidden or unethical reasons.

Dependencies refer to external factors or components that must be in place for the digital platform to function effectively.

The dependencies:

1. Hardware and Devices: The platform's usability depends on the availability of compatible hardware and devices.

2. Database Management: An efficient and secure database management system is essential to store and manage user data, academic records, and other information.

3. Administrative Support: The digital platform requires ongoing administrative support, including regular updates, maintenance, and troubleshooting

4. NLP (Natural Language Processing) Tool: The digital platform's functionality relies on the integration of a reliable NLP tool.

3.3 METHOD OF REQUIREMENT ELICITATION

3.3.1 Use Case Scenarios

In order to design the overall system different possible use case scenarios where considered. Each scenerio differed in the type of information user was requesting and the area from which the user was requesting information. This process allowed to better understand what information had to be added in the dataset. It also allowed to generate an idea of how the user would be interacting with the system thus making the user experience design more effective.

3.3.2 Interviews

Interviews were conducted with peers and other possible users to understand what the user might be expecting from the application. This also allowed in creating a database that satifiably catered to the needs of the user. Understanding the user expectations allowed us to incorporate various features such as the TTS system that provided a better experience to the user.

3.4 SYSTEM FEATURES AND REQUIREMENTS

Functions of the EVA project include Speech Recognition, Natural Language Processing, Information Retrieval, and Event Management.

3.4.1 Functional Requirements

Back-end

(i) Natural Language Processing:

The system should be able to understand natural language queries from users and provide relevant responses.

(ii) Database integration:

The system should retrieve information from a database and present it to the user in a user-friendly manner.

Front-end

(iii) User Interface design:

EVA should have an intuitive and user-friendly interface design that allows users to navigate through the system.

(iv) Voice input and output:

The system should allow users to interact with the system using voice commands and receive audible responses.

3.4.2 External Interface Requirements

Android device with :

- minimum 7" display
- microphone input

- internet connectivity

3.4.3 Nonfunctional Requirements

1. Performance requirements :

The system should be able to respond to user queries and requests quickly and efficiently, without significant delays or lags.

2. Software quality attributes :

In the development phase, rigorous testing and constant user review are to be performed, so that the quality of the software is being maintained and all the requirements are being fulfilled.

3. Usability :

The system should be easy to use and navigate, with a clear and intuitive user interface that allows users to perform tasks quickly and efficiently. This will save working time and lessen the pressure on the user.

3.4.4 Hardware Requirements

1. Touch screen android device with internet connectivity

2. 100 MB free memory

3.4.5 Software Requirements

1. Windows 10 or higher

2. Visual Studio Code

3. Flutter sdk

4. SQLite

5. Dart language

3.5 PROCESS MODEL

Agile software development process model would be the best fit for this project.

Agile is a flexible and iterative approach that promotes collaboration, adaptability, and customer-centric development. It allows for continuous improvement and the incorporation of feedback throughout the development lifecycle. Here's why Agile is well-suited for this project:

1. Incremental Development: The Agile model encourages breaking the project into smaller, manageable increments or iterations. Each iteration delivers a functional and potentially shippable product increment. This approach allows for early delivery of essential features, making it easier for stakeholders to see progress and provide valuable feedback.

2. Customer Involvement: The involvement of college students, faculties, and visitors is crucial for the success of the digital platform. Agile emphasizes regular interactions with stakeholders to gather requirements, prioritize features, and validate functionalities. This collaborative approach ensures that the platform aligns closely with the users' needs and

expectations.

3. Flexibility and Adaptability: As the project progresses, new requirements or changes may arise. Agile handles change effectively by embracing the idea of responding to change rather than following a rigid plan. This adaptability is essential in a dynamic academic environment where needs and priorities can evolve rapidly.

4. Continuous Testing and Integration: Agile encourages continuous testing and integration of new functionalities throughout the development process. This ensures that issues are identified early on and that the platform remains stable and reliable.

5. Regular Review and Retrospective: Agile incorporates regular review meetings and retrospectives to evaluate progress, identify areas for improvement, and adjust development strategies. These practices enable the project team to continuously refine their approach and deliver a high-quality product.

6. Quick Time-to-Market: Agile's iterative nature allows for faster time-to-market for essential features. This can be beneficial in meeting critical deadlines or adapting to academic schedules.

7. Mitigating Risks: By delivering functional increments early, Agile helps identify and address potential risks at an early stage. This proactive approach reduces the likelihood of major setbacks or project failure.

3.6 FEASIBILITY STUDY

3.6.1 Economic Feasibility

Economic feasibility includes the cost of development, i.e, the expenses for developers, tools, software etc, cost of hardware, cost of licences and other miscellaneous costs.

The application requires no cost for development. A paid version of the Text to Intend API is available but free version is also available with a little sacrifice on speed and availability.

In case of hardware costs an Android device, preferably a tablet, is required for hosting EVA to be used in the college.

3.6.2 Technical Feasibility

The challenge of incorporating a machine learning model to find intend from text occurs in development of application. Flutter only supports a very few machine learning models that too only by the use of plugins. Creating a native Android plugin for machine learning model is technically not feasible and may cause performance bottlenecks. It is possible to host the model on a server but it is economically unaffordable hence the feasible solution is to use a freely available API for Text to Intend conversion.

3.6.3 Operational Feasibility

For proper functionality EVA must be able to produce accurate answers for the user queries. To make that possible an extensive and accurate dataset of various types of questions possible and the best possible answers to those questions have to collected.

This must be performed by identifying user needs and user reviews.

CHAPTER 4
ARCHITECTURAL DESIGN

4.1 INTRODUCTION

Architectural design focuses on the design of system architecture. It describes the structure and behavior of the system. It defines the structure and relationship between various modules of system development process. The diagrammatic representation of the system architecture is called the system architecture diagram. This diagram gives us the abstract view of the components and their relationship with the system that makes the system work.

4.2 ARCHITECTURAL DIAGRAM

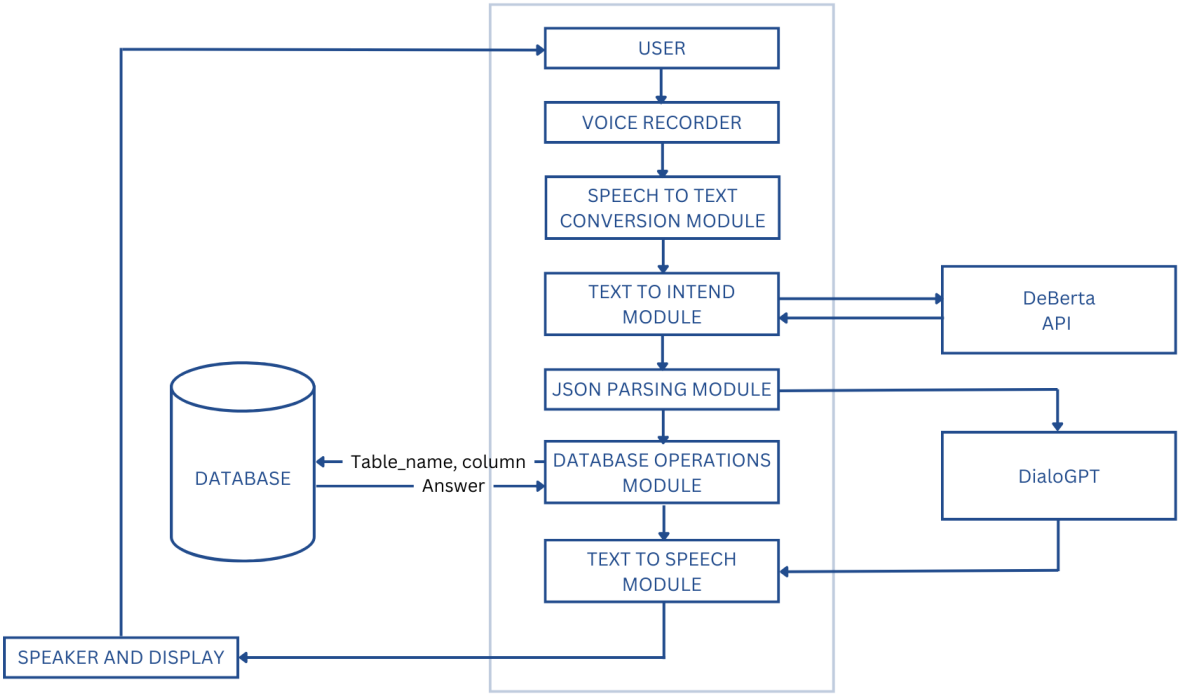


Figure 4.1: Architectural diagram

4.3 MODULE DESCRIPTION

Table 4.1: Description of Modules

MODULE	INPUT	PROCESS	OUTPUT
Voice recorder	Voice command from the user.	Generate the audio file.	Audio file in .aac format.
Speech to text module	Audio file from voice recorder.	Converting speech to text using the device's inbuilt STT capabilities.	Transcribed output.
Text to intent module	Transcription from Speech to text module.	Analyzing input text and recognizing intent based on classifiers. Parsing response from API.	List of intents based on priority.
Database operations module	Intent list from Text to intent module.	Finding respective table and column from the intent list. Creating query. Fetching data from the database.	Answer for the user command as text output.
Text to speech module	Generated output from the database.	Converting text to speech using the device's inbuilt TTS capabilities.	Speech output.

CHAPTER 5
DESIGN AND IMPLEMENTATION

5.1 INTRODUCTION

A data flow diagram depicts the flow of information in a business process graphically. It shows how data is moved from input to file storage and report generation. Flow charts, by visualizing the system flow, provide users with useful insights into the process and open up new avenues for defining and improving their business. A DFD can be broken into levels and layers, allowing users to focus on defining a specific step.

5.2 DATA FLOW DIAGRAMS

5.2.1 Level 0

The Context Level DFD depicts the educational virtual assistant application and its interactions with external entities at a high level. It depicts the data flow in the system by representing the system as a single process. The Context Level DFD provides a high-level overview of the system's operations without getting into specifics.

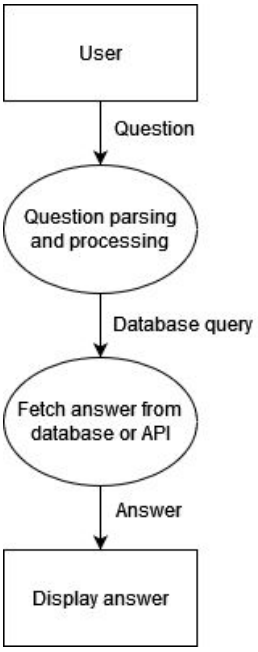


Figure 5.1: Level 0 DFD

5.2.2 Level 1

The Level 1 Data Flow Diagram (DFD) for the educational virtual assistant application provides a more detailed view of the system's operations compared to the Context Level DFD. It breaks down the main process into subprocesses, revealing more interactions and data flows within the system. While the Context Level DFD represents the system as a single process, the Level 1 DFD presents a clearer picture of how data moves between different components within the application.

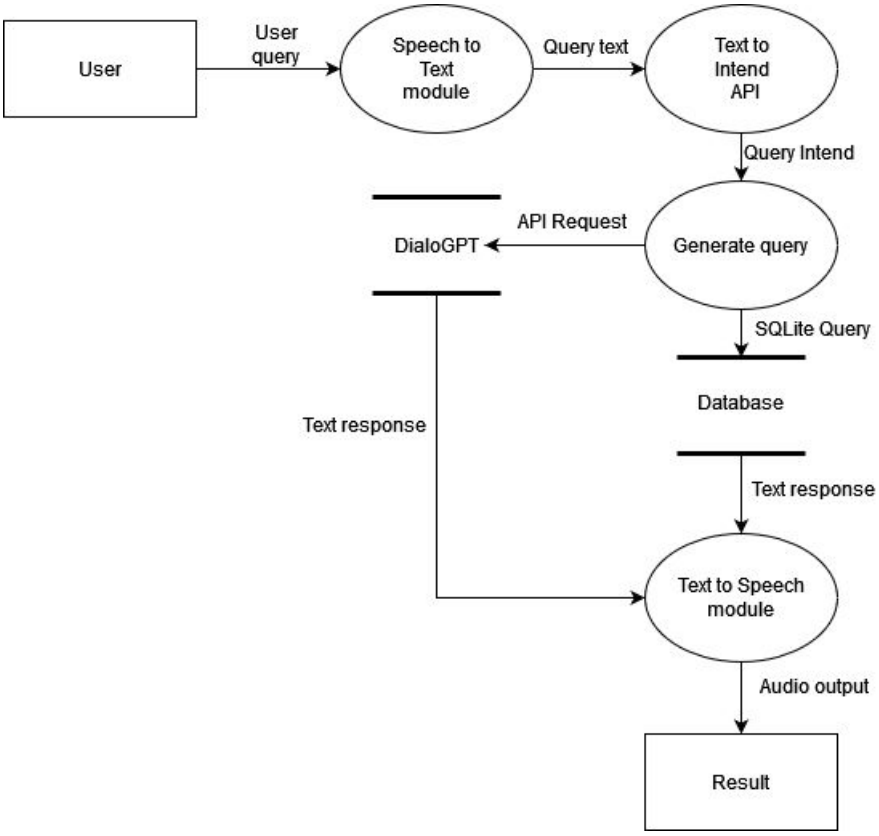


Figure 5.2: Level 1 DFD

5.2.3 Level 2

The Level 2 DFD provides a more granular view of the system's internal processes, showing how user queries are processed, how the AI generates responses, and how specific details are fetched from the database. It allows stakeholders to understand the data flow and interactions within the application at a more detailed level than the Level 1 DFD.

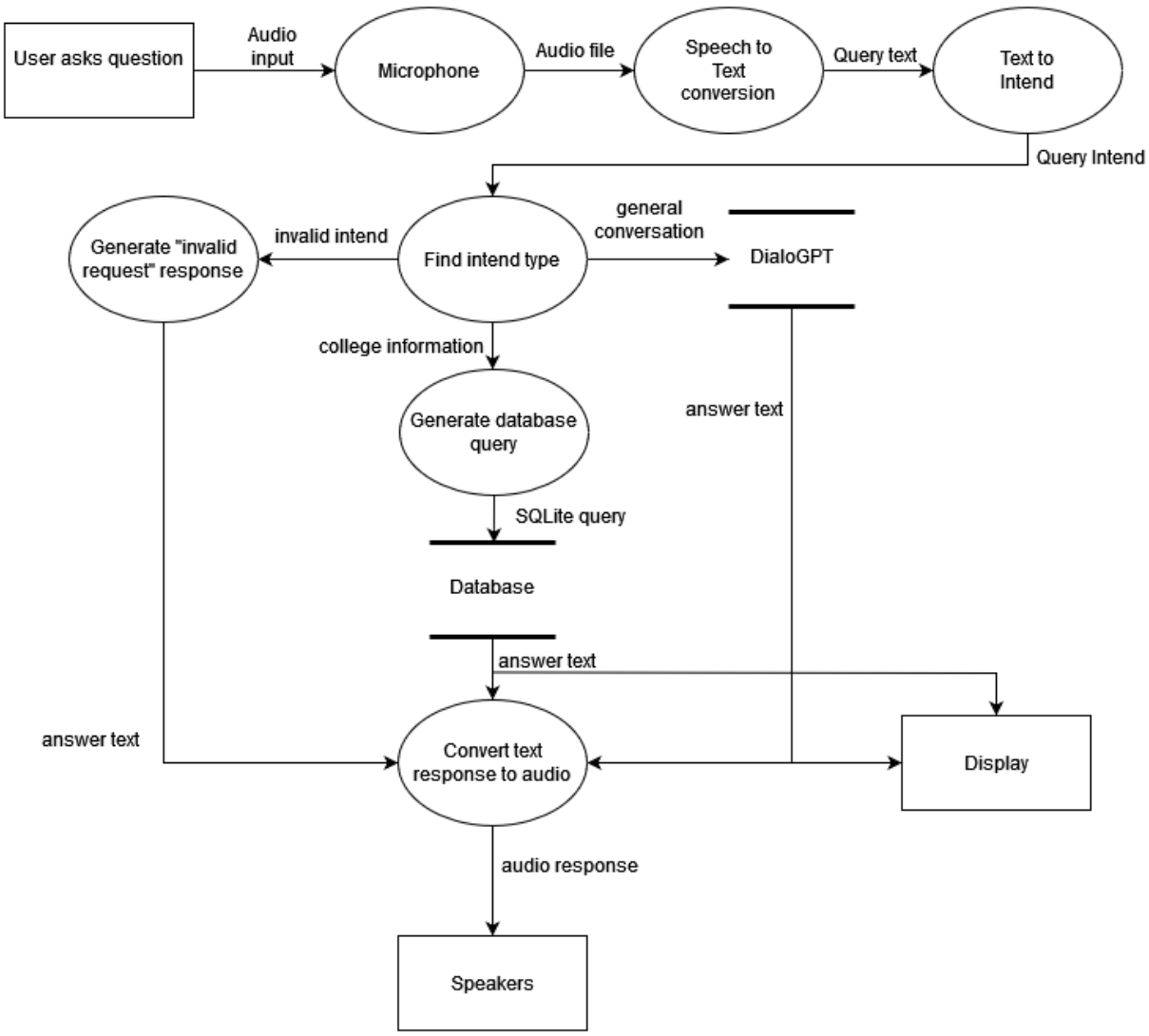


Figure 5.3: Level 2 DFD

5.3 WORK PLAN

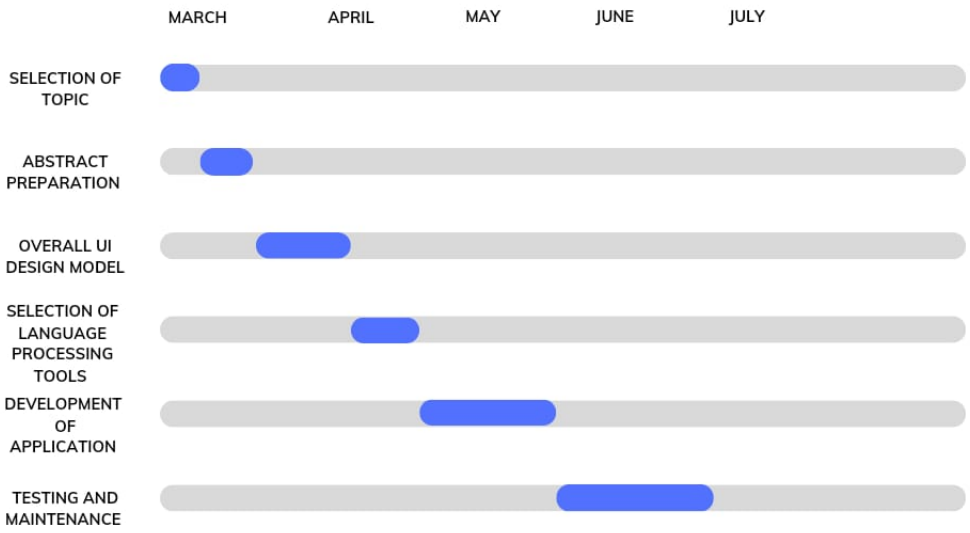


Figure 5.4: Work plan

CHAPTER 6
CODING

6.1 INTRODUCTION

A brief introduction to each of the frameworks/software tools/Specific libraries used and Data sets used, stating the company/organization which provides it, its merits/why used it for your project and its limitations, if any.

The framework used to build EVA is Flutter. Flutter is a cross-platform app development framework developed by Google. It is one of the best frameworks available for the development of hybrid apps. The SQFlite package provided by Flutter has been utilized for database management. SQFlite is a plugin provided by Flutter to use the SQLite tool in the Flutter apps. SQLite is the distribution of SQL database management tools for working with mobile applications.

The device's inbuilt Speech to Text and Text to Speech capabilities have been utilized for audio communication with the user. EVA's communication skills have been further expanded by the integration of DialoGPT API. DialoGPT allows EVA to respond to basic communications such as "How are you", providing it a human-like feel.

6.1.1 Frameworks

(i) Flutter

Flutter is the most popular cross-platform app development platform. The easy learning curve makes it the perfect choice for beginners in the field of mobile development. The availability of vast amount of community developed plugins allows developers to build

apps without worrying about the underlying complexities in native implementation of features. Flutter apps can be exported to multiple platforms - Android, iOS, Windows, Linux and Web. It also allows powerful tools to make debugging easier. Flutter provides more performance in comparison to other such frameworks.

The Dart language used by Flutter is also simple to learn and powerful to create efficient code. Latest features such as null safety are provided by Dart which makes development easier.

(ii) SQLite

SQLite is an implementation of SQL toolchain that is designed specifically for mobile devices. It is a light weight tool which allows it to be exported along with the application easily. In Flutter, SQLite is implemented through the sqflite package. It provides access to all the powerful capabilites of SQLite in database management.

6.1.2 Tools

(i) Android Studio

Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It also provides the ability to build and run Flutter apps. The extensive tools provided by Android Studio makes it the prime choice for development of mobile apps

6.1.3 ML Library

(i) DeBERTa-v3-base-mnli-fever-anli

This is a zeroshot classification model hosted by huggingface. It classifies the given text based on the given parameters. This model was trained on the MultiNLI, Fever-NLI and Adversarial-NLI (ANLI) datasets, which comprise 763 913 NLI hypothesis-premise pairs. This base model outperforms almost all large models on the ANLI benchmark. The base model is DeBERTa-v3-base from Microsoft. The v3 variant of DeBERTa substantially outperforms previous versions of the model by including a different pre-training objective

6.1.4 Audio/video processing library

(i) speech_to_text

A Flutter library that exposes device specific speech recognition capability. This plugin contains a set of classes that make it easy to use the speech recognition capabilities of the underlying platform in Flutter.

(ii) flutter_tts

It is a community developed flutter plugin. It allows developers to access the Text to Speech functionality of the device. The high level functions of the plugin abstracts the underlying technology vastly. The easiness of use makes it one of the best choices for TTS.

6.1.5 Data Set

The dataset was prepared solely for EVA. It contains answers to possible queries that may arise from the user. The database has been split into various tables each containing answer regarding a specific domain.

6.2 SAMPLE CODES

6.2.1 Database module

```
getAnswerFromDB(List<String> intents) async {
  if (intents.contains('general conversation')) {
    await getSmartReply(intents);
    return;
  }
  var queryString = "";
  if (tables.contains(intents[0]) && queries.contains(intents[1])) {
    queryString =
        "select answer from ${intents[0]} where query='${intents[1]}';";
  } else if (tables.contains(intents[1]) && queries.contains(intents[0])) {
    queryString =
        "select answer from ${intents[1]} where query='${intents[0]}';";
  }

  if (queryString != "") {
    print(queryString);
    var res = await db.rawQuery(queryString);
    print(res);
    if (res.isEmpty) {
      homeController.updateAnsState(AnswerState.Error);
      homeController.updateAnswer('SORRY I CANNOT ANSWER YOU');
      await homeController.updateIsAnswerReady(true);
    } else {
      var queryRes = res.first;
      homeController.updateAnswer(
          queryRes['answer'].toString() ?? 'SORRY I CANNOT ANSWER YOU');
      print(homeController.answer.value);
      if (homeController.answer.value.length < 100) {
        homeController.updateAnsState(AnswerState.Available);
      } else {
        homeController.updateAnsState(AnswerState.Generic);
      }
      await homeController.updateIsAnswerReady(true);
      print("DB RES: ${queryRes["answer"] ?? 'SORRY I CANNOT ANSWER YOU'}");
    }
  } else {
    homeController.updateAnsState(AnswerState.Error);
    homeController.updateAnswer('SORRY I CANNOT ANSWER YOU');
    await homeController.updateIsAnswerReady(true);
  }
}
```

Figure 6.1: Database module 1

```
  Future<void> getSmartReply(List<String> intents) async {
    final smartReply = SmartReply();
    smartReply.addMessageToConversationFromRemoteUser(
        homeController.userPrompt.value,
        DateTime.now().millisecondsSinceEpoch,
        "1");
    final response = await smartReply.suggestReplies();
    print(response.suggestions);
    homeController.updateAnswer(response.suggestions[0]);
    homeController.updateAnsState(AnswerState.Available);
    await homeController.updateIsAnswerReady(true);
  }
}
```

Figure 6.2: Database module 2

6.2.2 Recording and Parsing module

```
recordMic() async {
  //then async await
  _isRec = !(_isRec);
  print("Staring record");
  if (_isRec) {
    homeController.resetAnswer();
    print("Recording");
    if (homeController.isAvailable.value) {
      await speech.listen(
          partialResults: false,
          listenMode: ListenMode.dictation,
          onResult: (result) async {
            await homeController.updateUserPrompt(result.recognizedWords);
          });
    } else {
      print("speech recognition not available.");
      ScaffoldMessenger.of(context).showSnackBar(
          SnackBar(content: Text("speech recognition not available")));
    }
    // some time later...

    // await recorder.start();
    homeController.updateIsQueryReady(false);
    homeController.isAnswerReady(false);
    //print('tempPath/myFile.m4a');
  }
  if (!_isRec) {
    print("stopping rec..");
    await speech.stop();
    await Future.delayed(Duration(seconds: 3));
    print("stopped rec..");
    print(homeController.userPrompt.value);
    if (homeController.userPrompt.value.isNotEmpty) {
      // var bartUrl = Uri.parse(
      //     "https://api-inference.huggingface.co/models/facebook/bart-large-mnli");
      var bartUrl = Uri.parse(
          "https://api-inference.huggingface.co/models/MoritzLaurer/DeBERTa-v3-base-mnli-fever-anli");
      var payloadRaw = {
        "inputs": homeController.userPrompt.value,
        "parameters": {
          "candidate_labels": intents,
        }
      };
```

Figure 6.3: Database module 3

```
var payload = jsonEncode(payloadRaw);
print(payload);
var bartRes = await http.post(bartUrl,
    headers: {
      "Authorization": "Bearer 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"
    },
    body: payload);

print(bartRes.body);
Map<String, dynamic> bartBody = jsonDecode(bartRes.body);
if (bartBody.containsKey('error')) {
  ScaffoldMessenger.of(context).showSnackBar(const SnackBar(
    content: Text(
      "DeBERTa is being initialized. Try again Please wait 20 secs",
      style: TextStyle(color: Colors.white),
    ),
    backgroundColor: blue,
  ));
  print("API Error: API is being initialized. Please wait");

  await Future.delayed(Duration(seconds: 25));
  bartRes = await http.post(bartUrl,
      headers: {
        "Authorization": "Bearer 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"
      },
      body: payload);

  print(bartRes.body);
  bartBody = jsonDecode(bartRes.body);

  if (bartBody.containsKey('error')) {
    ScaffoldMessenger.of(context).showSnackBar(const SnackBar(
      content: Text(
        "DeBERTa is unavailable.",
        style: TextStyle(color: Colors.white),
      ),
      backgroundColor: blue,
    ));
    print("API Error: API is unavailable.");
    return;
  }
```

Figure 6.4: Database module 4

Figure 6.5: Database module 5

6.3 SUMMARY

The functionalities of the application where implemented using the Flutter framework and Dart programming language. Speech to Text and Text to Speech functionalities were implemented using the speech_to_text and flutter_speech plugins respectively, in Flutter. Using the sqflite package enabled the usage of databases in the mobile application. The usage of Flutter framework allows the application to be exported to multiple platforms hence improving the portability of the application. The full capabilities of DialoGPT API were not utilized in the application. It is possible to enable contextual conversations in the application by utilizing DialoGPT, but since this was out of the scope of functional requirements demanded by the application it is not implemented. Creating the dataset was the major challenge in the development process. Identifying the possible areas where

queries might occur and design of the database accordingly were key to the implementation of the application. The problem of generation of responses and presentation of the responses to the user required intense brainstorming to reach potential solutions.

The API to be used for Text to Intend was not an easy choice. The lack of proper free tier hosted models became a problem. The unavailability of servers still pose an issue the application might face since usage of free tier servers does not guarantee availability of services all the time. It can be resolved by hosting a model on device but that is not feasible due to technical difficulties in hosting machine learning model on an Android platform.

CHAPTER 7
RESULTS AND DISCUSSION

7.1 INTRODUCTION

EVA has been successfully developed using Flutter framework and has been exported to the Android platform. All the functional requirements of the application which were planned has been successfully implemented and validated.

User interface provided is simple, aesthetic and intuitive. It allows users to get the result with least interaction and has been designed to user friendly

Application is fast and responsive, it interprets the users input accurately and generates output in the least possible time. The generated results are displayed in an unambiguous manner. The text to speech functionality adds significantly to the user experience.

7.2 RESULTS

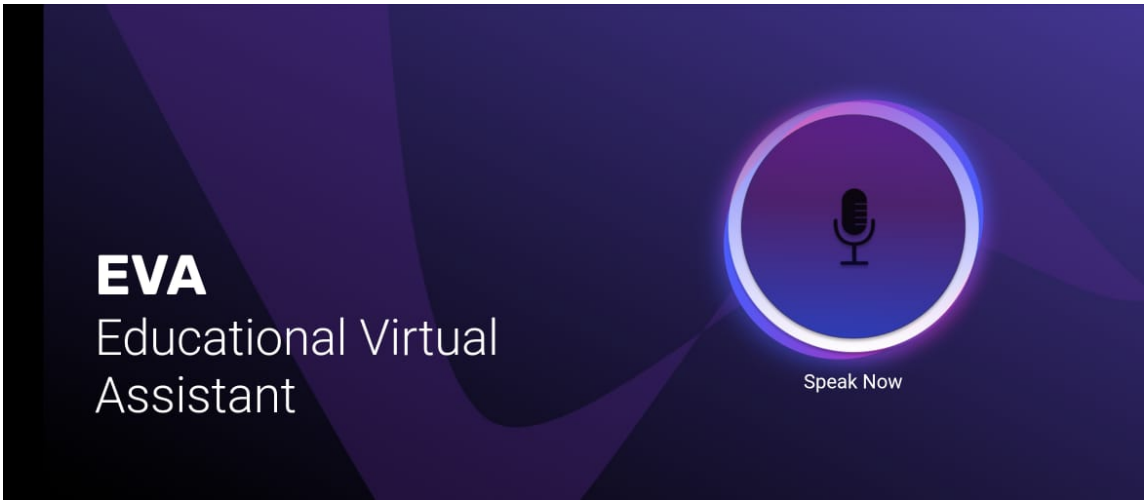


Figure 7.1: Homepage

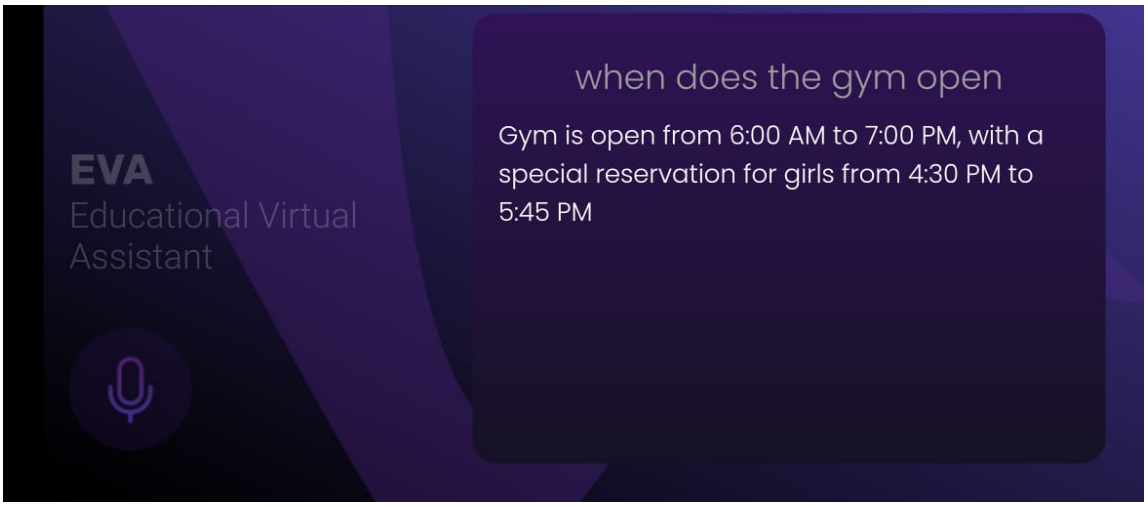


Figure 7.2: Answerpage (result from database)

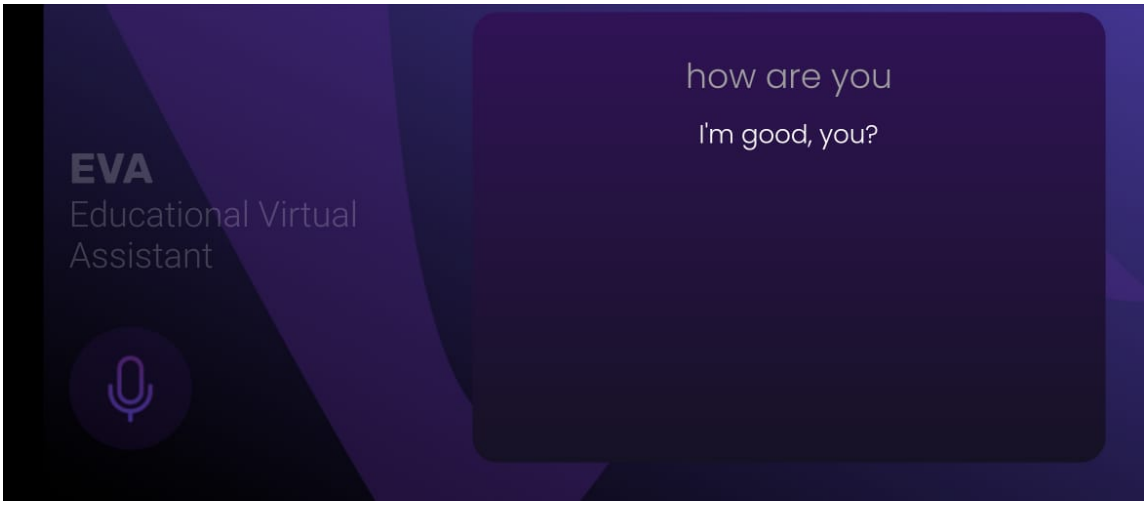


Figure 7.3: Answerpage (result from DialoGPT)

7.3 RESULT ANALYSIS

In the best case, where answer is available in the local database, EVA takes an average of 5 seconds to get the answer. In the cases where answers have to be fetched from DialoGPT, EVA takes an average of 7 seconds to produce the answer. These times also depend on the network speed.

It may be noted that Text to Intend API (Deberta-v3) takes around 20 seconds to initialize which adds to the time required to generate response incases where the API must be initialized.

Out of the 30 questions queried, 24 where answered correctly. 5 were unanswered and 1 was answered incorrectly. It was found that the number of unanswered questions can be reduced by improving the database while the number of incorrect answers indicate the inaccuracy of the system.

CHAPTER 8
DOCUMENTATION

8.1 INTRODUCTION

There is a growing demand for a more effective and accessible solution to manage queries. This project aims to address this need by introducing an apt platform for college students, faculties, and visitors. This report provides a comprehensive overview of the project, outlining its objectives, development process, key features, and potential impact on the academic community. By combining AI capabilities with user interactions, the Educational Virtual Assistant aims to empower users by offering valuable insights, and easy access to information. Throughout this report, we delve into the project's architecture, user interfaces, data management, and future prospects of the Educational Virtual Assistant, highlighting its potential to create a more efficient, inclusive, and technologically-driven environment.

8.2 CONTACT

For more information contact:

- Diya Baiju N - diyanoopuram@gmail.com
- Amaan Zain N - zainamaanaz@gmail.com
- Adith Koliyot - koliyotadith4@gmail.com
- Akash P T - akashpt248@gmail.com

CHAPTER 9
CONCLUSION AND FUTURE WORK

9.1 CONCLUSION

By adopting an Agile software development approach, we have successfully crafted a dynamic platform that provides personalized assistance and intelligent responses to user queries. The seamless integration of Natural Language Processing technology has empowered the application to understand and respond effectively to inputs, ensuring an intuitive user experience. Throughout the project, the requirement analysis, along with user feedback, has played a pivotal role in shaping the application's design and functionality, aligning it closely with the needs of the community. The educational virtual assistant application stands as a testament to our commitment to excellence, and efficiency, as it enhances the academic journey for all stakeholders involved. As we move forward, we remain dedicated to the continuous improvement and refinement of the application, adapting to the evolving needs of the education landscape and cementing our position at the forefront of technology-enabled support.

9.2 ADVANTAGES

- Fast Response and Real-Time Service: The system's quick response time ensures that users receive timely and relevant information. The integration of cutting-edge technologies, such as AI and NLP, allows the virtual assistant to process user queries rapidly and deliver real-time responses. This real-time service provision ensures that students, faculties, and visitors can access essential information and support without delay, enhancing their productivity and overall experience with the

platform.

- Easy to Use and Simple Interface: The system's user-friendly and intuitive interface makes it easy for all users to interact with the platform effortlessly. The simple design and clear navigation enable users to access features and functionalities without encountering complexities.

- Efficient and Time-Saving: Using speech as input allows users to interact with the system quickly, saving time compared to typing or navigating through menus. For individuals on different literacy levels or with physical disabilities or limited motor skills, voice-based interaction can be a more efficient and convenient method of accessing information and completing tasks.

9.3 LIMITATIONS

- Recognition Accuracy: The accuracy of speech recognition technology can vary based on factors such as accent, pronunciation, and background noise. In some cases, the system may misinterpret or fail to recognize certain commands or responses accurately, leading to potential errors in the information provided or actions taken.

- Data Maintenance and Cleanup: Over time, the database may accumulate redundant or outdated information. Implementing effective data maintenance and cleanup processes is necessary to ensure data integrity, minimize storage overhead, and enhance database performance.

9.4 FUTURE EXPANSION

While the current implementation of EVA includes significant milestones, there are several avenues for future improvement and expansion. Some potential areas for future work include:

- Enhanced Voice Assistant Capabilities: Continuously improving the speech processing and natural language understanding capabilities of EVA to provide more accurate and comprehensive responses to user queries and commands.

- Personalization and Customization: Implementing features that allow users to personalize their EVA experience, such as setting preferences, receiving personalized notifications, and tailoring information based on individual needs and interests.

- Continuous Maintenance and Updates: Regularly updating and maintaining the EVA system to address any bugs, security vulnerabilities, and evolving user needs, ensuring a smooth and reliable user experience.

- Wake word detection: A wake word is a specific phrase or keyword that triggers the voice assistant to start listening. You will need to implement a wake word detector to recognize when the user is addressing the voice assistant and activate the speech recognition module.

- Integration with external APIs: Depending on the features one wants to include in his voice assistant, one may need to integrate with external APIs to access additional data or services. For example, one could integrate with a weather API to retrieve current weather conditions for the user's location.

REFERENCES

1. **B. Arend** (2018). Hey siri, what can i tell about sancho panza in my presentation? investigating siri as a virtual assistant in a learning context? URL https://api.semanticscholar.org/CorpusID:59453852.

2. **B. G. Bob Strahan** (2019). Building a multi-channel q and a chatbot at saint louis university using the open source qnabot.

3. **B. Derbyshire** (2021). Transforming ada the college digital assistant, into ada the digital human.

4. **A. J. Jean-Charles** (2018). Internet of things in education: Artificial intelligence voice assistant in the classroom. URL https://api.semanticscholar.org/CorpusID:196001229.

5. **C. Lakshmi Chandana**, **V. Ashita**, **G. Neha**, **K. Sravan Kumar**, **D. Suresh Babu**, **G. Krishna Kishore** and **Y. Vijaya Bharathi** (2022). Voice-enabled virtual assistant. **P. Karrupusamy**, **V. E. Balas** and **Y. Shi**, editors, Sustainable Communication Networks and Application, 335–346. Springer Nature Singapore.