# Virtual Sharable Personality

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

# Bachelor of Technology

in

# Electronics and Instrumentation Engineering

*by*


**Vinayak Sharma**

**19BEI0084**

**Under the guidance of**

**Dr. Marimuthu R.**

**School of Electrical Engineering**

**VIT,Vellore.**


Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

**April,2023**

**DECLARATION**

I here by declare that the thesis entitled "**Virtual Sharable Personality"** submitted by me, for the award of the degree of *Bachelor of Technology in Electronics and Instrumentation Engineering* to VIT University is a record of bonafide work carried out by me under the supervision of **Dr. Marimuthu R.**, Associate Professor, School of Electrical Engineering, VIT University, Vellore.

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place: Vellore

Date: 19/04/23 **Signature of the Candidate**

# CERTIFICATE

This is to certify that the thesis entitled "**Virtual Sharable Personality"** submitted by **Vinayak Sharma (19BEI0084)**, School of Electrical Engineering, VIT, Vellore, for the award of the degree of *Bachelor of Technology in Electronics and Instrumentation Engineering* , is a record of bonafide work carried out by him under my supervision during the period, 07.10.2022 to 03.05.2023,, as per the VIT code of academic and research ethics.

The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university. The thesis fulfills the requirements and regulations of the University and in my opinion meets the necessary standards for submission.

Place: Vellore

Date: 19/04/2023

**Signature of the Guide**

**(Dr. Marimuthu R.)**

**Head of the Department**

**EIE**

**DEAN**

**SELECT**

**Acknowledgement**

With immense pleasure and deep sense of gratitude, I wish to express my sincere thanks to my supervisor **Dr. Marimuthu R.**, Associate Professor, School of Electrical Engineering, VIT University, without his motivation and continuous encouragement, this project work would not have been successfully completed.

I am grateful to the Chancellor of VIT University, **Dr. G.Viswanathan**, the Vice Presidents, the Vice Chancellor for motivating me to carry out research in the VIT University and also for providing me with infrastructural facilities and many other resources needed for my project.

I express my sincere thanks to **Dr. Mathew M. Noel** , Dean, School of Electrical Engineering, VIT University for his kind words of support and encouragement. I like to acknowledge the support rendered by **my colleagues** in several ways throughout my project work.

I wish to extend my profound sense of gratitude to **my parents** for all the sacrifices they made during my research and also providing me with moral support and encouragement whenever required.

Place: Vellore

Date: 19/04/2023 **Vinayak Sharma**

**Executive Summary**

The aim of this thesis is to create a system that can be used to mimic a user's personality and also allows for securely sharing this virtual personality.
This is accomplished in a three part process:

1. a user interacts with an Artificial Intelligence model via conversations to create a virtual personality,
2. the model then saves this personality on to a blockchain network,
3. upon sharing this personality, another user can extract meaningful information from it or build upon it further.

The conversational aspect of the system is handled using a *Natural Language Processing (NLP)* subsystem. Specifically, the *Bidirectional Encoder Representations from Transformers (BERT)* model is employed which effectively converts the conversations into *tokens*.
These tokens are then appended to a custom *blockchain*. All the conversations that make up a virtual personality is stored on this blockchain and when any further interaction occurs, this blockchain is looked up and a relevant answer is prepared as a response.
When a user desires to share a virtual personality, they can simply share this private blockchain to allow another user to interact with their virual personality.

Some possible applications of this implementation are:

- Medical Field: Analysis of a patient's mental state.
- Digital Secretary: Preparation of a summary of events.
- Context Search: Searching text by meaning instead of keywords.

**Keywords:** *blockchain*, *NLP*, *BERT*.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

## List of Terms and Abbreviations

**API** Application Programming Interface

**BERT** Bidirectional Encoder Representations from Transformers

**NLP** Natural Language Processing

**SQUAD** Stanford Question Answering Dataset

**AI** Artificial Intelligence

**UI** User Interface

**IoT** Internet of Things

CHAPTER 1

# INTRODUCTION

## 1.1 Objective

The primary objective of this thesis is to allow any user to train a system to mimic their personality using simple conversations and to able to securely share this personality at will.

## 1.2 Motivation

Pre-trained language models have existed for a while now, and some are quite impressive in their capabilities (Guven and Unalir (2022)). There also exists various technologies for secure transmission, processing and storage of data. This thesis intends to bring these two technological advancements under a single roof. By combining the powers of AI and Blockchain, it is possible to create secure and private services that utilise the benefits of AI based tools.

## 1.3 Background

Most commonly available public implementations of large language models aim to be like a Swiss Army Knife, i.e. they try to be good at everything. For such models to be able to handle a variety of tasks, they have to be trained on a large amount of data which is usually obtained by web scraping. This inherently makes them prone to being inaccurate at personalised tasks. Furthermore, such implementations heavily rely on obtaining data from their users for further training of these models as elucidated in Harris and Waggoner (2019), resulting is a loss of privacy. **Virtual Shareable Personality** aims to tackle these issues head-on. Firstly, by allowing the user to develop a virtual personality from scratch, i.e. providing a blank canvas for the user to start with and then building up the personality block by block with each conversation, the ability to answer personalised questions is very high. Secondly, by building the personality on a private blockchain, our system is inherently private and secure.

# CHAPTER 2

# PROJECT DESCRIPTION AND GOALS

## 2.1 Review of Literature

- Dinh and Thai (2018)
  This paper talks about how Artificial Intelligence (AI) and Blockchain are some of the most distruptive technologies to emerge in recent times and how these technologies can have a profound impact on our daily lives. The authors describe in detail, the ongoing work in these domains and how these have a very promising future. They also talk about how these domains can evolve in the near future to provide even more value by integration with other systems. With a quote, ”UNLESS THE ADVERSARY OWNS THE MAJORITY MINING POWER, BLOCKCHAIN IS ALMOST IMPOSSIBLE TO HACK”, it is explained how blockchain systems are inherently self-correcting and secure. The authors also discuss how decentralised blockchains function with miners keeping these systems alive. Topics like data-sharing, privacy and personalisation are also discusses to provide a top-level overview.

- Deepa et al. (2022)
  This paper discusses how the blockchain technology can be used in making big data systems more decentralised and secure. It describes in brief, blockchain, big data systems and the benefit of integrating the two. It then, presents an analysis of various blockchain services integrated with key components of a big data system like data acquisition, data storage, data analytics, and data privacy preservation. The authors then review latest advancements in the integration of blockchain and big data applications like smart city, smart healthcare, smart transportation, and smart grid. All in all, a complete overview of how blockchain systems augment and can augment big data applications is presented.

- Harris and Waggoner (2019)
  This paper discusses how large advancements in artificial intelligence have recently been made possible by machine learning, however these tend to be very centralised. It explains how predictions are monetised and big datasets that are

hard to maintain are usually the ones in demand. The authors then offer a system that allows members to create a collaborative dataset that is maintaned by a community. Additions are made using smart contracts and the dataset is publicly availble for use. They then also discuss incentivising users to contribute good data. Detailed mathematical models are presented to compute incentives for the maintainers and contributors. Three incentive mechanisms are elucidated: ”Gamification”, ”Rewards Mechanism Based on Prediction Markets” and ”Deposit, Refund, and Take”. The authors then discuss possible implementation techniques.

- Guven and Unalir (2022)
  This paper evaluates RNP methods and how these supplement pre-trained BERT model pre-trained with the Stanford Question Answering Dataset (SQUAD) dataset. The authors first use the SQUAD dataset to benchmark the capabilites of the BERT model. Then RNP methods are used along with BERT in hopes of improving its accuracy. The authors do discover that such models improve the accuracy of the Bidirectional Encoder Representations from Transformers (BERT) model by 1.1% to 8.76% depending on the exact implementation. As these methods don’t need to be pre-trained, these can be used along with any other model to improve accuracy. The paper also demonstrates that these methods specifically make the BERT model better at sentence selection and answer detection tasks.

- Xia et al. (2017)
  This paper brings to light the fact that users would rather withhold information than sharing them with an entity that may not be able to keep that information secured. To address this challenge, the authors present a model for blockchain based data sharing system. Though the interest of this project was in implementation of such a system in the health care sector, the proposed model is dynamic enough to be applied in various other applications. The presented model relies on the fact that blockchains are immutable and highly secure. This makes them a primary candidate for storing sensitive information. The authors suggest using cryptographically authenticated access roles to access a blockchain. This not only guarantees that the data is secure but also that it cannot be accessed even in a protected form by adversaries. Furthermore, possible implementation details are shared for deployment of such a system with well described processes and use of protocols. It is important to note that the proposed system is scalable in nature making it very feasible for large scale deployments.

## 2.2 Project Description

This project has two major scopes of work: Software and Hardware. The hardware dictates which software components can be run on a specific device. With various deployment options, a large array of devices can be supported.

### 2.2.1 Software

The backend system is divided into three components:

1. Natural Language Processing (NLP) Subsystem: The NLP subsystem is responsible for utilising the BERT model for powering conversations. It understands the user's input by tokenising the statements and builds an appropriate response.

2. Blockchain Subsystem: The Blockchain subsystem is responsible for storing and retrieving the tokenised conversations along with maintaining integrity over the data.

3. Application Programming Interface (API) Subsystem: The API subsystem provides a simple method for devices to communicate with a server to send user inputs and receive the generated responses.

The frontend system may contain the following components:

- User Interface (UI): A user interface may be displayed to a user for devices with a screen. It may contain features such as on screen typing for conversing, a log of previous conversations and sharing features.

- Text to Speech: Text to Speech may be used to read out the generated responses to the user.

- Speech Recognition: Speech recognition may be used to record the user's voice and convert the audio into texual conversation.

- Sharing: An intuitive method may be used for the user to share their personality over a blockchain network like numbered codes or usernames and passwords.

### 2.2.2 Hardware

To promote ease of use, we shall aim to support as many types of devices as possible, ranging from embedded system to wearables. Since the performance capabilities of such a diverse range of devices varies largely, the deployment to a device has to be very specific in nature. For the deployment of this project, devices can be classified based on the following criteria:

- Connectivity: A device with internet connectivity can benefit from server side processing, cloud storage and easy sharing options. Whereas a device without internet connectivity will have to process conversations and store them on device and will also lack sharing ability.
- Computational Performance: A device with enough computational performance to run a NLP model locally will benefit from quicker responses and will have the ability to maintain full security and privacy over its data. A weaker device would need internet connectivity and shall communicate with a server that processes conversations on its behalf. It is important to note that running the model locally also requires significant free disk space.
- Storage: A device with ample amount of storage would be able to run the pre-trained BERT model locally and would also be able to store a dense personality with numerous large conversations it has been trained with. On the contrary, a device with low storage availability will need internet connectivity and will have to resort to server side processing.
- Audio/Visual: A device with support for microphones and speakers can record interactions and announce responses and a device with a display can show these conversations on screen. In short, the user's interaction depends on the audio/visual capabilities of the device.

Considering these limitations, it is advisable to run the model locally on single board computers, PCs and automation systems whereas low power devices such as mobile phones, wearables and embedded systems shall prefer server side processing.

## 2.3 Goals

This project aims to allow the user to perform the following with a virtual personality:

- train,
- interact,
- and share;

all while maintaining personalised responses, privacy and security.

CHAPTER 3

# TECHNICAL SPECIFICATION

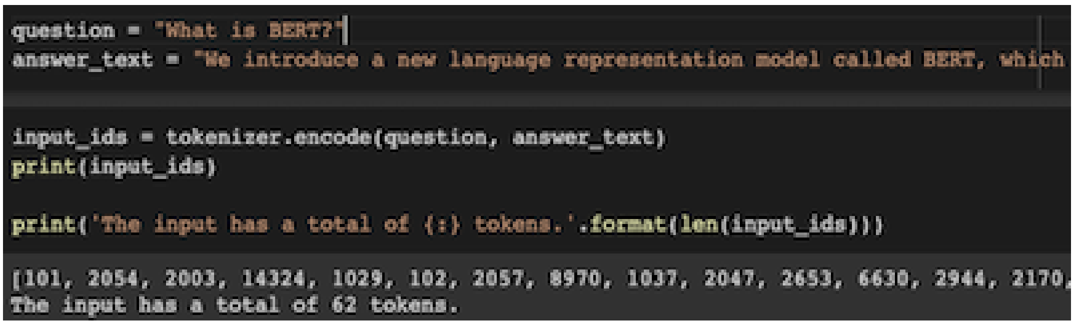
## 3.1 Software Specification

The backend for this project is developed in Python. This allows us to quickly prototype the deploy on virtually any environment.

### 3.1.1 NLP Subsystem

First, we load the BERT model fine tuned with the SQUAD 2.0 dataset. This equips our system with the ability to carry out conversations in a question and answer format.

Then, the BERT model is used to tokenize user inputs and the response is decoded in the reverse manner for displaying to the user.



```
question = "What is BERT?"
answer_text = "We introduce a new language representation model called BERT, which

input_ids = tokenizer.encode(question, answer_text)
print(input_ids)

print('The input has a total of {:} tokens.'.format(len(input_ids)))

[101, 2054, 2003, 14324, 1029, 102, 2057, 8970, 1037, 2047, 2653, 6630, 2944, 2170,
The input has a total of 62 tokens.
```

**Fig. 3.1** BERT Model Tokenising Inputs

### 3.1.2 Blockchain

A custom blockchain is defined for use in this project. The structure of a block is as follows:

$index$: the position of the block in the blockchain,

$timestamp$: the timestamp of the instant when the block was created,

$data$: the encoded tokens of the conversation,

$prevhash$: the hash of the previous block in the blockchain,

$hash$: the hash of this block.

The SHA265 algorithm is used for hashing blocks, ensuring integrity of the stored information. Each conversation is stored in this blockchain as a separate block in the form tokens. When a response is required, this blockchain is read and the tokens are derived to construct a response which is again stored into the blockchain as a separate block.

An instance of a RedisDB database is used as a store for the blockchain. Since RedisDB is an in-memory database it is relatively faster in performing transactions.


**Fig. 3.2** Some Blocks in the Blockchain

### 3.1.3 API

A GraphQL api is developed to serve as the link between a front end user interface and the NLP subsystem. The API is served via the https protocol, allowing virtually any platform with internet connectivity to communicate with the backend. An example of a developed query is as follows:

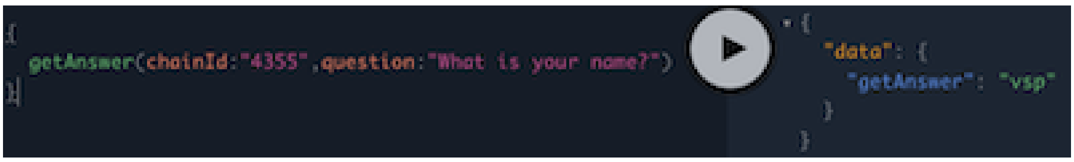



**Fig. 3.3** Example Query

### 3.1.4 User Interface

A common user interface is developed using the Flutter Framework. This allows us to run an app on Android, iOS, Web, Windows, macOS and Linux platforms, all with a single code base. The app contains two text fields. One to train the personality, named ”Tell”. Text entered in this field acts as training data. The other field is named

”Ask”, this field can be used to extract information out of the trained personality. The fields also have appropriate controls for speech recognition and text to speech features. However it is important to note that a User Interface can be developed with various other technologies as long as they support making http calls or can interoperate with python or an equivalent platform for hosting the NLP and Blockchain subsystems.

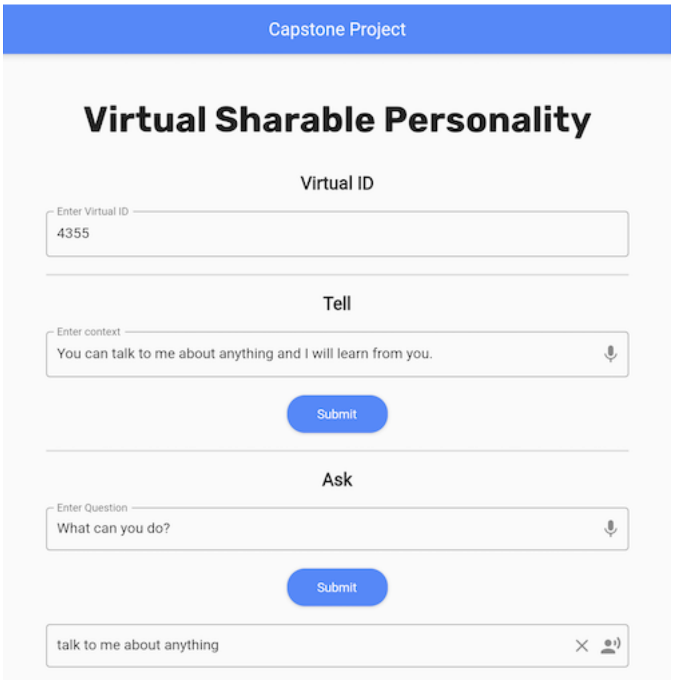



**Fig. 3.4** user Interface

## 3.2 Hardware Specification

The frontend for this project can be materialised with various technologies as the backend is very versatile in compatibility. But the choice of this final component of tech stack is decided by the limitations of the hardware it has to be deployed on. The technologies can hence be broadly categorised based on deployment hardware:

- Embedded Systems: The ideal form of user interface for embedded device is voice control. The system records the user’s input and converts the speech into text using voice recognition. This information is then sent to the backend service which processes the received text and sends back a response which the device converts into a voice using a text to speech feature.

- Wearables: These devices can offer both, a voice control and a visual interface for the users to interact with. Though the functionality of a visual interface is limited due to the small sizes of such devices but at the very least these may be able to display responses on a display.

- PCs and Mobile Devices: These are the most commonly used devices and can fully feature all the quirks of a system like this, including modern user interfaces, speech recognition and text to speech functionality, integration with host operating system or other apps, etc. They also have the performance capability to run a model like this locally.

CHAPTER 4

# DESIGN APPROACHES AND METHODS

## 4.1 Possible Design Approaches

**A few design approaches are described here that cover the various types of deployments possible for a system like such.**.

### 4.1.1 Embedded Systems

This section refers to devices that are already in use towards a purpose and can be repurposed or additionally expanded in scope to incorporate a system like this. For example, a digital control on a washing machine or a refrigerator, etc. These devices are often limited in their functionality but the more recent ones usually feature Internet of Things (IoT) applications, this inherently gives them the ability to stay connected to the internet. This allows us to use these devices and connect to our backend API for transmitting conversations.

### 4.1.2 Custom Handheld Device

A custom handheld device can be built that is capable of running all the subsystems locally. This allows the device to power conversations even off grid and sync up with the blockhain on the server when online and if required. These devices can be as small as the size of credit cards and are very efficient in power usage. The ideal choice of hardware for a device like this is a Single Board Computer along with either a touchscreen or a microphone and a speaker combo.

### 4.1.3 Modern Generic Devices

This section targets all the PCs, Mobile Phones and Wearable technologies. The hardware on these devices usually supports a wide array of platforms and in our case, we use the most common technologies which guarantees compatibility. A python program can be directly run on these devices if there is a need to run the NLP and Blockchian subsystems locally. A Flutter app can also be run on these devices to server a frontend user interface to the user.

## 4.2 Constraints, Alternatives and Tradeoffs

### 4.2.1 Constraints

- Devices running the backend locally must have enough processing power to run BERT model in real time.
- Devices running the backend locally must also have enough storage capacity to host the BERT model pre-trained with SQUAD 2.0 dataset and additional conversations the user may have.
- Devices relying on speech recognition and text to speech must obviously have a microphone and speaker attached on present on board.
- Devices lacking the audio functionality or requiring visual feedback may have a touchscreen or a digital screen with any other form of input device.
- The process to use/carry/maintain the device must not be cumbersome for the user.

### 4.2.2 Alternatives

- To BERT Model: other large language models could also be used for this task. A very popular example is GPT-3 and its newest iteration GPT-4.
- To Blockchain: end to end encryption can be employed to protect the conversations of each user.
- To GraphQL: there are various technologies that can serve an API, the other most common one is RESTful APIs.
- To RedisDB: there are plenty more database services, few common ones are SQLite, MongoDB, MySQL, SupraDB, PostgreSQL.
- To Flutter: numerous other frameworks can be used to deploy a font end for the user to interact with, a few cross platform alternatives are: React Native, Xamarin Forms.

### 4.2.3 Tradeoffs

Two major tradeoffs considered while building this project are:

- By maintaining fully personalised responses, we lose the ability for the model to reply to general questions without training.
- If we choose a fully local deployment to guarantee privacy and security, we lose the ability to share personalities.

CHAPTER 5

# SCHEDULE, TASKS AND MILESTONES

## 5.1 Schedule

**The work schedule and deadlines decided in the beginning per the requirements of this project is described in Fig. 5.1 and Table 5.1**.


**Fig. 5.1** Work Schedule

**Table 5.1** Deadlines

| | Review 1 | Review 2 | Review 3 |
|---|---|---|---|
| Deadline | 27.01.23 | 13.03.23 | 03.05.23 |
| Primary Milestone | Neural Network and Natural Language Processing based subsystem that can carry out conversations | Exporting/Importing neural network weights to share the "brain" of the model | Implementation of Speech to Text and Text to Speech |
| Secondary Milestone | Basic User Interface with core backend features | Sharing the "brain" via blockchain networks | Deployment on portable devices |

## CHAPTER 6

# PROJECT DEMONSTRATION

## 6.1 NLP Subsystem

### 6.1.1 Tokenizing Input

| | |
|---|---|
| [CLS] | 101 |
| what | 2,054 |
| is | 2,003 |
| bert | 14,324 |
| ? | 1,029 |
| [SEP] | 102 |
| we | 2,057 |
| introduce | 8,970 |
| a | 1,037 |
| new | 2,047 |
| language | 2,653 |
| representation | 6,630 |
| model | 2,944 |
| called | 2,170 |
| bert | 14,324 |
| , | 1,010 |
| which | 2,029 |
| stands | 4,832 |
| for | 2,005 |
| bid | 7,226 |
| ##ire | 7,442 |
| ##ction | 7,542 |
| ##al | 2,389 |
| en | 4,372 |
| ##code | 16,044 |
| ##r | 2,099 |
| representations | 15,066 |
| from | 2,013 |
| transformers | 19,081 |
| . | 1,012 |

**Fig. 6.1** Definition of BERT tokenised by BERT

### 6.1.2 Testing the Model

```
[ ]  context="""
     My name is VSP.
     """
     question = "What is your name?"
     answer_question(question,context)

     Query has 14 tokens.

     Answer: "vsp"
     'vsp'
```

**Fig. 6.2** Testing BERT

## 6.2 Blockchain Subsystem

### 6.2.1 Training

```
{
  addContext(chainId:"4355",context:"My name is VSP.")
}
```

**Fig. 6.3** Train Query

```
{
  "data": {
    "addContext": "True"
  }
}
```

**Fig. 6.4** Train Response

### 6.2.2 Testing

```
{
  getAnswer(chainId:"4355",question:"What is your name?")
}
```

**Fig. 6.5** Test Query

```
{
  "data": {
    "getAnswer": "vsp"
  }
}
```


**Fig. 6.6** Test Response

## 6.3 Cross Platform App

Accessing the same blockchain form a cross platform app.

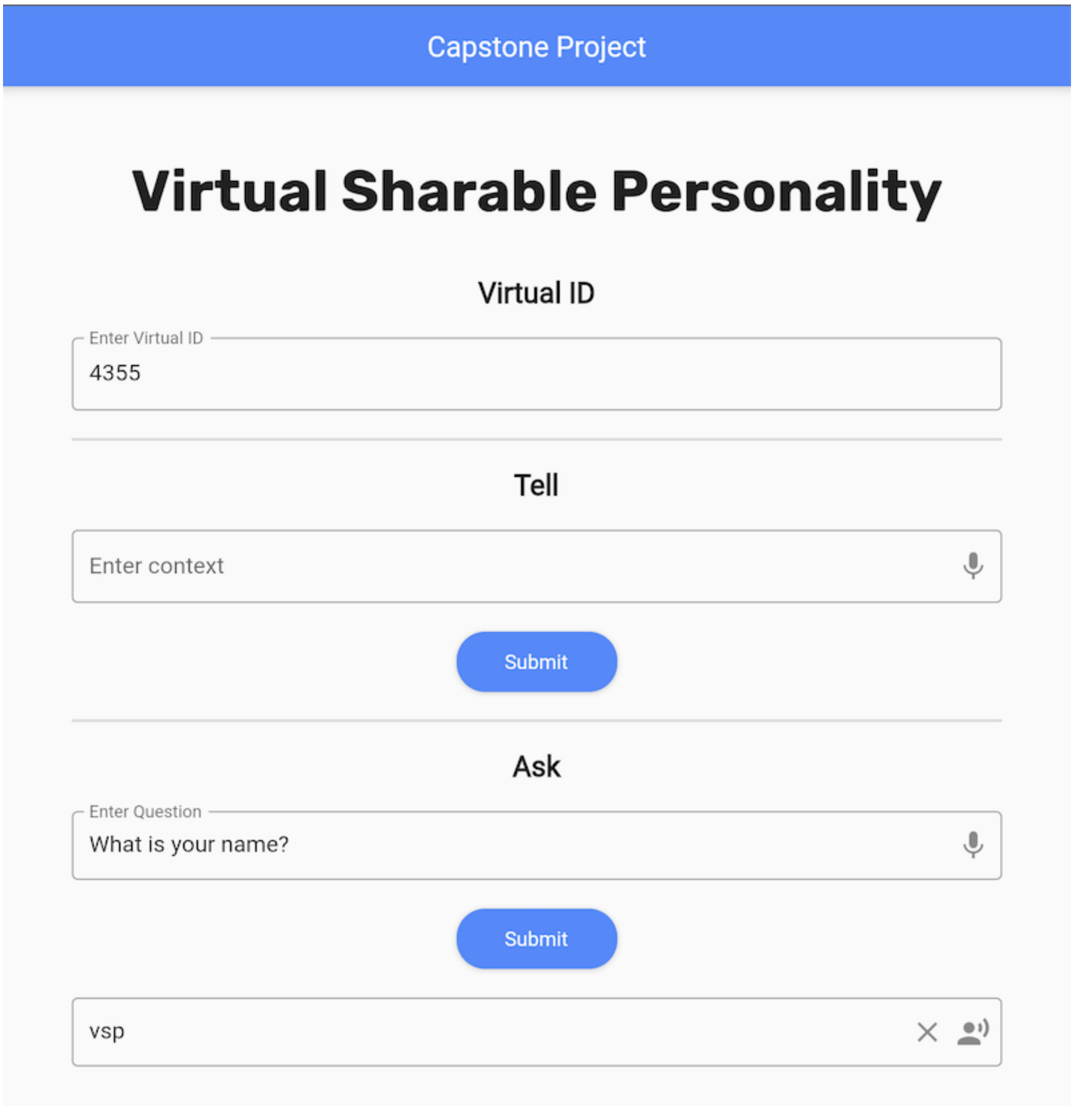



**Fig. 6.7** Cross Platform App Response

## CHAPTER 7

# COST ANALYSIS / RESULT AND DISCUSSION

## 7.1 Cost Analysis

### 7.1.1 Software

The backend subsystems require a host to run on. The ideal choice for such a deployment is a Virtual Private Server. The costs incurred to run such a server depends on the following:

- number of Virtual CPU cores,
- amount of RAM (usually in GB),
- storage size (usually in GB),
- and network traffic bandwidth limits (usually in TB).

The average cost to deploy a minimum viable product would be approximately $20/month.

### 7.1.2 Hardware

The cost of hardware largely depends on the use case. To run the system on a regular device, no extra costs are associated but to build a custom device using a single board computer can range between $50 to $300 which includes the cost of the single board computer itself, may include a touchscreen display and speaker/microphone combo.

## 7.2 Results and Discussion

The first evident observation from this project refers to Fig. 6.2, where we see that the BERT model is excellent at understanding languages to generate an appropriate response as opposed to performing a simple search and returning the results. We confirm this from the fact that the response to the question ”What is your name?” was not ”My name is VSP.” but instead was just ”vsp”.

The next observation stems from the fact that the blockchain subsystem works flawlessly to provide the BERT model with the necessary historic context. This asserts that blockchains can be very useful in storing data that is incremental in nature.

Finally, we also observe the ease of development that cross platform frameworks like Flutter provide. This makes the process of building apps and deploying them on to a wide variety of devices very easy.

All in all, a platform is created that allows a user to create their own virtual personality and share it as they like.

A system like such is very important for the current times and even more so for the times to come, especially because our right to privacy is always at risk with he rise of more digital services. We must devote necessary resources to build secure systems for everyone.

CHAPTER 8

# SUMMARY

## 8.1 Summary

- BERT model fine tuned with the SQUAD 2.0 dataset is used to serve as the NLP subsystem for the project.
- The tokens generated by the BERT model for each conversation are stored on a custom Blockchain.
- A GraphQL API is developed that serves as the middleware between frontend applications and the NLP and Blockchain subsystems.
- A Cross Platform Flutter App is developed to run on various types of devices.
- Combined, a system is built that allows a user to train, share and interact with a virtual personality.

# REFERENCES


Deepa, N., Pham, Q.-V., Nguyen, D. C., Bhattacharya, S., Prabadevi, B., Gadekallu, T. R., Maddikunta, P. K. R., Fang, F. and Pathirana, P. N. (2022), 'A survey on blockchain for big data: Approaches, opportunities, and future directions', *Future Generation Computer Systems* **131**, 209–226.
**URL:** *https://www.sciencedirect.com/science/article/pii/S0167739X22000243*

Dinh, T. N. and Thai, M. T. (2018), 'Ai and blockchain: A disruptive integration', *Computer* **51**(9), 48–53.

Guven, Z. A. and Unalir, M. O. (2022), 'Natural language based analysis of squad: An analytical approach for bert', *Expert Systems with Applications* **195**, 116592.
**URL:** *https://www.sciencedirect.com/science/article/pii/S0957417422000884*

Harris, J. D. and Waggoner, B. (2019), Decentralized and collaborative ai on blockchain, *in* '2019 IEEE International Conference on Blockchain (Blockchain)', pp. 368–375.

Xia, Q., Sifah, E. B., Smahi, A., Amofa, S. and Zhang, X. (2017), 'Bbds: Blockchain-based data sharing for electronic medical records in cloud environments', *Information* **8**(2), 44.