

# AURO

## MINI PROJECT REPORT

*Submitted By*

**GINU MATHEW      14019917**

*In partial fulfillment for the award of degree of*

**Bachelor of Technology**

in

**Computer Science & Engineering**

**Mahatma Gandhi University, Kottayam**



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**VISWAJYOTHI COLLEGE OF ENGINEERING & TECHNOLOGY,  
VAZHAKULAM**

# AURO

## MINI PROJECT REPORT

*Submitted By*

**GINU MATHEW      14019917**

*In partial fulfillment for the award of degree of*

**Bachelor of Technology**

in

**Computer Science & Engineering**

**Mahatma Gandhi University, Kottayam**

*Under the guidance of*

**Mrs. RITTY JACOB**

**Assistant Professor, Department of CSE**



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**VISWAJYOTHI COLLEGE OF ENGINEERING & TECHNOLOGY,  
VAZHAKULAM**

**April 2017**

# **VISWAJYOTHI COLLEGE OF ENGINEERING & TECHNOLOGY, VAZHAKULAM**

## **Department of Computer Science and Engineering**

### **Mission**

1. Foster the principles and practices of computer science to empower life-long learning and build careers in software and hardware development.
2. Impart value education to elevate students to be successful, ethical and effective problem-solvers to serve the needs of the industry, government, society and the scientific community.
3. Promote industry interaction to pursue new technologies in Computer Science and provide excellent infrastructure to engage faculty and students in scholarly research activities.

### **Program Educational Objectives**

#### **Our Graduates**

1. Shall have creative and critical reasoning skills to solve technical problems ethically and responsibly to serve the society.
2. Shall have competency to collaborate as a team member and team leader to address social, technical and engineering challenges.
3. Shall have ability to contribute to the development of the next generation of information technology either through innovative research or through practice in a corporate setting.
4. Shall have potential to build start-up companies with the foundations, knowledge and experience they acquired from undergraduate education.

### **Program Outcomes**

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 analyze 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 principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. Life-long 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.

#### Program Specific Outcomes

1. Ability to integrate theory and practice to construct software systems of varying complexity.
2. Able to Apply Computer Science skills, tools and mathematical techniques to analyze, design and model complex systems.
3. Ability to design and manage small-scale projects to develop a career in a related industry.

## DECLARATION BY THE CANDIDATE

I hereby declare that the mini-project report entitled “**AURO**” submitted by me to the Department of Computer Science & Engineering, Viswajyothi College of Engineering & Technology, Vazhakulam in partial fulfillment of the requirement for the award of the degree of **Bachelor of Technology in Computer Science & Engineering** is a record of bonafide project work carried out by us under the guidance of **Mrs. Ritty Jacob**. I further declare that the work reported in this project has not been submitted, either in part or in full, for the award of any other degree in this college.

**Ginu Mathew**

**VISWAJYOTHI COLLEGE OF ENGINEERING & TECHNOLOGY,  
VAZHAKULAM**

**Department of Computer Science & Engineering**



**BONAFIDE CERTIFICATE**

This is to certify that the Mini Project report entitled “**AURO**” is a bonafide record of the work by **GINU MATHEW (14019917)** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science & Engineering** of Mahatma Gandhi University, Kottayam.



Date:

**Mrs. Ritty Jacob**

Place: Vazhakulam

**Mini Project Guide**

**Mrs. Nimmy George**

**Dr. K.N Ramachandran Nair**

**Mini Project Coordinator**

**Professor and HOD**

**Internal Examiner**

**Internal Examiner**

## ACKNOWLEDGEMENT

First and foremost I thank God Almighty for His divine grace and blessings in making this project possible. May he continue to lead me in the years to come. It is my privilege to render heartfelt thanks and gratitude to our most beloved Manager, **Msgr. George Oliapuram** and Principal, **Dr. JosephKunju Paul C.** for providing the opportunity to do this project during the third year (2017) of our B.Tech degree course. I am deeply thankful to the Head of the Department, **Dr. K.N Ramachandran Nair** for his support and encouragement. I would like to express my sincere gratitude to my project guide

**Mrs. Ritty Jacob**, Asst. Professor, Department of Computer Science and Engineering for her motivation, assistance and help for the project. I also express sincere thanks to the project coordinator **Mrs. Nimmy George**, Asst. Professor, Department of Computer Science and Engineering for her guidance and support. I also thank all the staff members of the Computer Science Department for providing their assistance and support. Last, but not the least I thank all my friends and family for their valuable feedback from time to time as well as their help and encouragement. I hereby apologize for any kind of omissions that may have taken place by mistake.

**Ginu Mathew**

## ABSTRACT

Our common understanding of a personal assistant is that of a person (or an agent) who is able to provide distinct help at a given time and in a given activity context. We must be very much familiar with the commercial products such as Apple's Siri, Google's Now and api.ai's Speak to It Assistant which are the necessary technology inventions (ie., composition of Natural Language Processing, Auto Speech synthesis, Dialogue management and Text-speech Processing) during the past 5 years. The possible application scenario have progressively been changing and now involves domains such as healthCare, weather forecast, navigation, translation, information and tutoring. This personal assistant (AURO) can be used to respond to simple questions and arithmetic calculations.

# Contents

<b>1</b>	<b>PREAMBLE</b>	<b>1</b>
1.1	Purpose . . . . .	1
1.2	Objective . . . . .	1
1.3	Product Scope . . . . .	2
<b>2</b>	<b>SYSTEM STUDY</b>	<b>3</b>
2.1	PROOF OF CONCEPTS . . . . .	3
2.1.1	Need of proof of concepts . . . . .	3
2.2	Result of proof of concept . . . . .	4
2.3	SYSTEM FEATURES . . . . .	4
2.3.1	Tokenize the inputs from the user . . . . .	5
2.3.2	Filter appropriate Keywords from the tokenized input . . . . .	5
2.3.3	Form and process Query . . . . .	5
2.4	NON-FUNCTIONAL REQUIREMENTS . . . . .	6
2.4.1	Performance Requirements . . . . .	6

2.4.2	Safety Requirements . . . . .	6
2.4.3	Security Requirements . . . . .	6
2.4.4	Software Quality Attributes . . . . .	6
2.5	REQUIREMENT ANALYSIS . . . . .	6
2.5.1	Hardware Requirements . . . . .	6
2.5.2	Software Requirements . . . . .	7
2.6	SOFTWARE DESCRIPTION . . . . .	7
2.6.1	Django . . . . .	7
2.6.2	jQuery . . . . .	8
2.6.3	Heroku . . . . .	8
2.6.4	Natural Language Toolkit . . . . .	9
2.6.5	Git . . . . .	9
<b>3</b>	<b>SYSTEM DESIGN AND MODELLING</b>	<b>10</b>
3.1	SYSTEM ARCHITECTURE . . . . .	10
3.2	CONTROL FLOW DIAGRAM . . . . .	11
3.3	DATA FLOW DIAGRAM . . . . .	11
3.4	USE CASE DIAGRAM . . . . .	13
3.5	FLOWCHART . . . . .	15
<b>4</b>	<b>IMPLEMENTATION</b>	<b>16</b>

4.1	INTRODUCTION . . . . .	16
4.2	IMPLEMENTATION PLAN . . . . .	17
<b>5</b>	<b>TESTING</b>	<b>18</b>
5.1	INTRODUCTION . . . . .	18
5.2	TEST CASES . . . . .	23
<b>6</b>	<b>CONCLUSION</b>	<b>24</b>
6.1	ADVANTAGES AND DISADVANTAGES OF AURO . . . . .	24
6.1.1	Advantages . . . . .	24
6.1.2	Disadvantages . . . . .	24
6.2	FUTURE SCOPE . . . . .	25

# List of Figures

3.1	Architecture for Auro . . . . .	11
3.2	Control Flow Diagram for Auro . . . . .	11
3.3	Level 0 DFD . . . . .	12
3.4	Level 1 DFD . . . . .	13
3.5	Use Case Digram . . . . .	14
3.6	Flowchart . . . . .	15

## **LIST OF ABBREVIATIONS**

SRS    Software Requirements Specification

DFD    Data Flow Diagram



# Chapter 1

## PREAMBLE

### 1.1 Purpose

We would have wished to use a single program just to get the answers in one go. But what we do nowadays is that we have to scroll through different applications or different websites and search hard to get what we actually want. Our proposed system is a solution to this problem. The project is to build a Personal Digital Assistant which is able to accept queries in the form of text (English) and respond back accordingly. This application can be used to get or fetch the user required processes based on the input provided. This document is intended for both the stakeholders and the developers of the software and will be proposed to the committee for its approval.

### 1.2 Objective

The main objective of our proposed project is to build a simple personal assistant which is able to accept commands in the form of text (natural language: English) and respond back again in text (natural language). This project is implemented by Machine Learning (ML) and Natural Language Processing (NLP).

### **1.3 Product Scope**

The scope of this product is to provide an efficient and enhanced tool for the users to perform various operations quickly and more efficiently. It finds application in many fields such as medical, educational etc. Our project is actually a demonstration of Machine learning and Natural Language Processing. The main objective of the proposed system is to convert machines to user friendly devices through the use the use of Machine Learning and Natural Language processing.

## Chapter 2

# SYSTEM STUDY

### 2.1 PROOF OF CONCEPTS

Proof of concept is a short and/or incomplete realization (or synopsis) of a certain method or idea(s) to demonstrate its feasibility, or a demonstration in principle, whose purpose is to verify that some concept or theory is probably capable of being useful. A related (somewhat synonymous) term is "proof of principle". The proof of concept is usually considered a milestone on the way to a fully functioning prototype. A proof of concept can refer to a partial solution that involves a relatively small number of users acting in business roles to establish whether the system satisfies some aspect of the requirements. The proof of concept is the first step we did to confirm that our project can go forward. It is an important part of the project that determines whether the project will reach a success

#### 2.1.1 Need of proof of concepts

Our main aim of this project 'Personal Digital Assistant'(PDA) is to provide assistance to its users in their work.PDA is a software that will accept query from the user and then tokenizes the query, filters the keywords and displays the result that the system gets through searching process if the keywords are valid i.e;present in the database. Our main aims:

1. Input the query in the form of text in English language.
2. This query text is tokenized, that is separated as tokens word by word.
3. Natural Language Processing(NLP) is done with the query from user.
4. NLP makes the interaction between computers and human(natural) language possible.
5. The necessary output will be displayed on a window after searching on web using internet.

## 2.2 Result of proof of concept

Our proof of concept was a success and we were allowed to continue with our project. In the present concept our project has wide scope as usage and internet has increased rapidly for searching and getting details about various things around.

We proposed a system which acts as a PDA(Personal Digital Assistant) to the all users. This helps the users in so many ways as when we use it we can avoid searching hard to get proper answers.

The results obtained are:

1. The input was accepted in English language.
2. The input was processed and search was done with Google Wikipedia as databases.
3. The required output was displayed to the user in the AURO window in English language.

## 2.3 SYSTEM FEATURES

**AURO** is an application that accepts commands from the user in the form of text (natural language) and reply back again in text (natural language).

### **2.3.1 Tokenize the inputs from the user**

#### **Description and priority**

In lexical analysis, tokenization is the process of breaking a stream of text up into words, phrases, symbols, or other meaningful elements called tokens. The list of tokens becomes input for further processing such as parsing or text mining. This feature is very much important as each word has to be extracted separately in order to choose and analyze the question of the user.

#### **Functional Requirement**

FREQ-1: Any question or input provided by the user.

### **2.3.2 Filter appropriate Keywords from the tokenized input**

#### **Description and Priority**

This feature analyzes the important terms or words from the tokenized user input so as to analyze the need of the user and to provide the user with the required output.

### **2.3.3 Form and process Query**

#### **Description and Priority**

This feature forms and process the query which will be compared with data in the database and give required output.

## 2.4 NON-FUNCTIONAL REQUIREMENTS

### 2.4.1 Performance Requirements

The performance of system lies in the way it is handled. The other factor of performance is the absence of any suggested requirements.

### 2.4.2 Safety Requirements

No such safety requirement is needed.

### 2.4.3 Security Requirements

No such security requirement is needed.

### 2.4.4 Software Quality Attributes

1. **Easy to operate:** The system should be easy to operate and limited to the budget of the user.
2. **Accuracy:** The accuracy of the proposed system is moderate.

## 2.5 REQUIREMENT ANALYSIS

### 2.5.1 Hardware Requirements

- Processor: Pentium IV or above
- RAM: 2GB or above
- Monitor: Min 14inch standard monitor

- Keyboard: Standard Keyboard
- Mouse: Standard mouse

### 2.5.2 Software Requirements

- Operating System: Ubuntu 16.10
- Python 2.7+
- git

## 2.6 SOFTWARE DESCRIPTION

The selection of an appropriate software for development is an important task, since completion of the system is greatly dependent on software selected. The different software used AURO includes :

### 2.6.1 Django

Django is a free and open source web application framework, written in Python. A web framework is a set of components that helps you to develop websites faster and easier. When you're building a website, you always need a similar set of components: a way to handle user authentication (signing up, signing in, signing out), a management panel for your website, forms, a way to upload files, etc. Django give us the ready made components to use. These framework help to reduce the burden of building a new site. The web server reads the letter and then sends a response with a webpage. The django is used to create the content of the webpage.

When a request comes to a web server, it's passed to Django which tries to figure out what is actually requested. It takes a web page address first and tries to figure out what to do. This part is done by Django's urlresolver. It is not very smart – it takes a list of

patterns and tries to match the URL. Django checks patterns from top to bottom and if something is matched, then Django passes the request to the associated function (which is called view).

### 2.6.2 jQuery

jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers.

The purpose of jQuery is to make it much easier to use JavaScript on your website. jQuery takes a lot of common tasks that require many lines of JavaScript code to accomplish, and wraps them into methods that you can call with a single line of code.

### 2.6.3 Heroku

Heroku is a cloud Platform-as-a-Service (PaaS) supporting several programming languages that is used as a web application deployment model. Heroku, one of the first cloud platforms, has been in development since June 2007, when it supported only the Ruby programming language, but now supports Java, Node.js, Scala, Clojure, Python, PHP, and Go. For this reason, Heroku is said to be a polyglot platform as it lets the developer build, run and scale applications in a similar manner across all the languages.

Applications that are run from the Heroku server use the Heroku DNS Server to direct to the application domain (typically "applicationname.herokuapp.com"). Each of the application containers, or dynos, are spread across a "dyno grid" which consists of several servers. Heroku's Git server handles application repository pushes from permitted users.



### 2.6.4 Natural Language Toolkit

The Natural Language Toolkit (NLTK) is a Python package for natural language processing. NLTK requires Python 2.7, or 3.4+. NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries.

NLTK is suitable for linguists, engineers, students, educators, researchers, and industry users alike. NLTK is available for Windows, Mac OS X, and Linux.

### 2.6.5 Git

Git is a version control system(VCS) for tracking changes in computer files and coordinating work on those files among multiple people. It is primarily used for software development, but it can be used to keep track of changes in any files. As a distributed revision control system it is aimed at speed, data integrity, and support for distributed, non-linear workflows.

Git was created by Linus Torvalds in 2005 for development of the Linux kernel, with other kernel developers contributing to its initial development. Its current maintainer since 2005 is Junio Hamano.

As with most other distributed version control systems, and unlike most client – server systems, every Git directory on every computer is a full-fledged repository with complete history and full version tracking abilities, independent of network access or a central server. Like the Linux kernel, Git is free software distributed under the terms of the GNU General Public License version 2.

## Chapter 3

# SYSTEM DESIGN AND MODELLING

### 3.1 SYSTEM ARCHITECTURE

A system architecture can comprise system components that will work together to implement the overall system. A system architecture is a conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system.

In our system architecture, the main component is the AURO API. It is the only component which is performing all the operations. Input is accepted from the user through the AURO window. User can input the query using any device that has internet access.

Query or the input is accepted only in English language and the answer or output will be displayed after searching for it in the databases used such as Google Wikipedia, in the same AURO window. No more components are involved here.

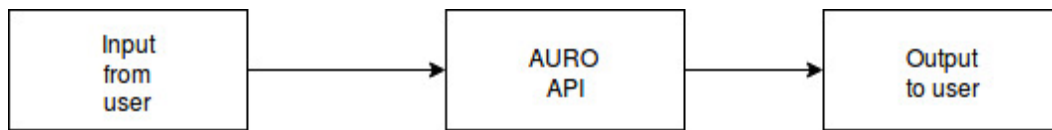


Figure 3.1: Architecture for Auro

## 3.2 CONTROL FLOW DIAGRAM

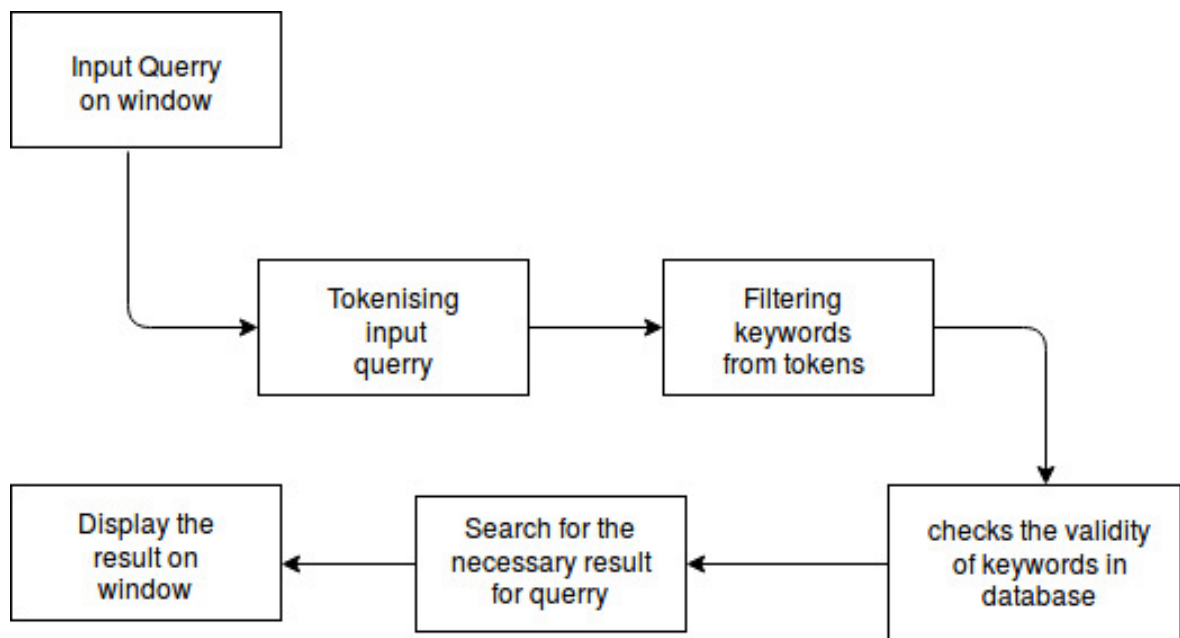


Figure 3.2: Control Flow Diagram for Auro

## 3.3 DATA FLOW DIAGRAM

A data flow diagram (DFD) is a graphical representation of the flow of data through a system. A DFD is often used as a preliminary step to create an overview of the system without going into detail. It is used in the problem analysis. It views system as a function that transforms input into the desired outputs. The data undergoes a series of transformations before it becomes the output. The data flow diagram aims to capture transformations that take place within the system to the input data so that eventually the output is produced. The agent that performs the transformation is the process. So, a DFD shows the movement of data

through different processes or transformations in the system.

### Level 0 Data Flow Diagram

The level 0 data flow diagram shown in Figure 3.3 contain an entity, user and the main process of the system. The user provide commands in the form of text. These commands are then converted into natural language. Further it search the database for the corresponding answer. The system retrieves the answer to user as final result.

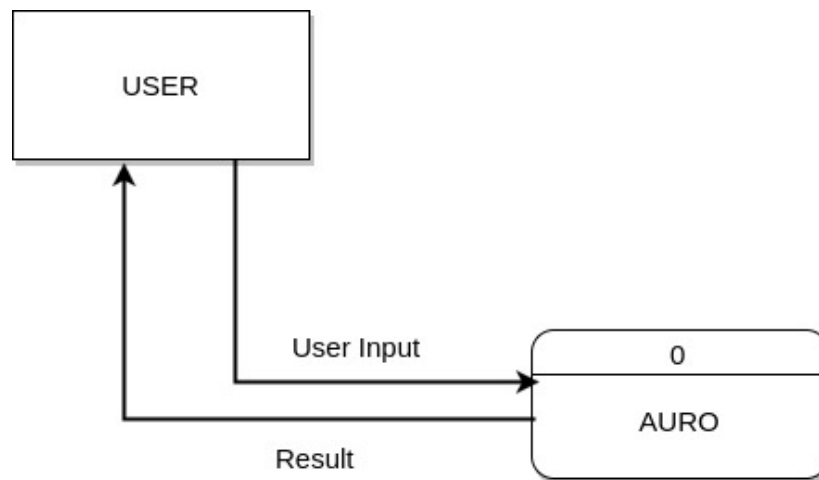


Figure 3.3: Level 0 DFD

### Level 1 Data Flow Diagram

A level 1 DFD represents the system's major process, data flows and the data stores at a high level of detail. The level 1 DFD shown in Figure 3.4 contain one external entity the user. The text command provided by the user is tokenized into words which forms group of words. From this group, the required keywords are filtered necessary for searching the database. The system search database for the corresponding answer and is given to the user as the final result.

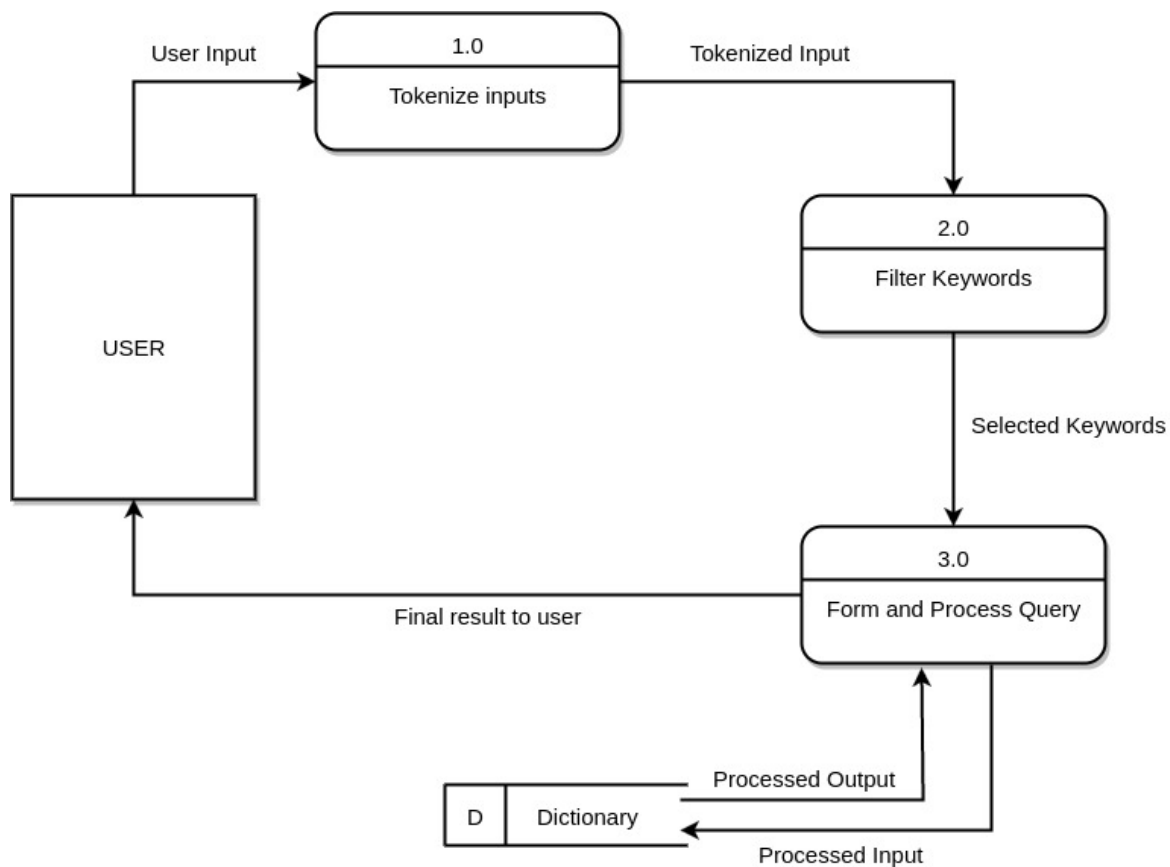


Figure 3.4: Level 1 DFD

### 3.4 USE CASE DIAGRAM

A use case diagram shown in Figure 3.5 is the representation of the user's interaction with the system that shows the relationship between the user and the different use cases. The use case diagrams are also referred to as behaviour diagrams used to describe a set of actions (use cases) that some systems or should or can perform in collaboration with one or more external users of the system. Each use case should provide some observable and valuable result to the stakeholders of the system.

External entities are referred to as actors which can be human users, external hardware or other systems. An actor is a named stick figure, or a class rectangle with <<actor>> keyword. A use case is a single unit of meaningful work. The purpose of the use case diagrams

is simply to provide the high level view of the system and convey the requirements. The use case is denoted by an ellipse.

The use case diagram of AURO consist of one external entity the user and four use cases, user input, processing the text, search database and display the result. The user provide input to the system which is processed into natural language by the use case processing the text.

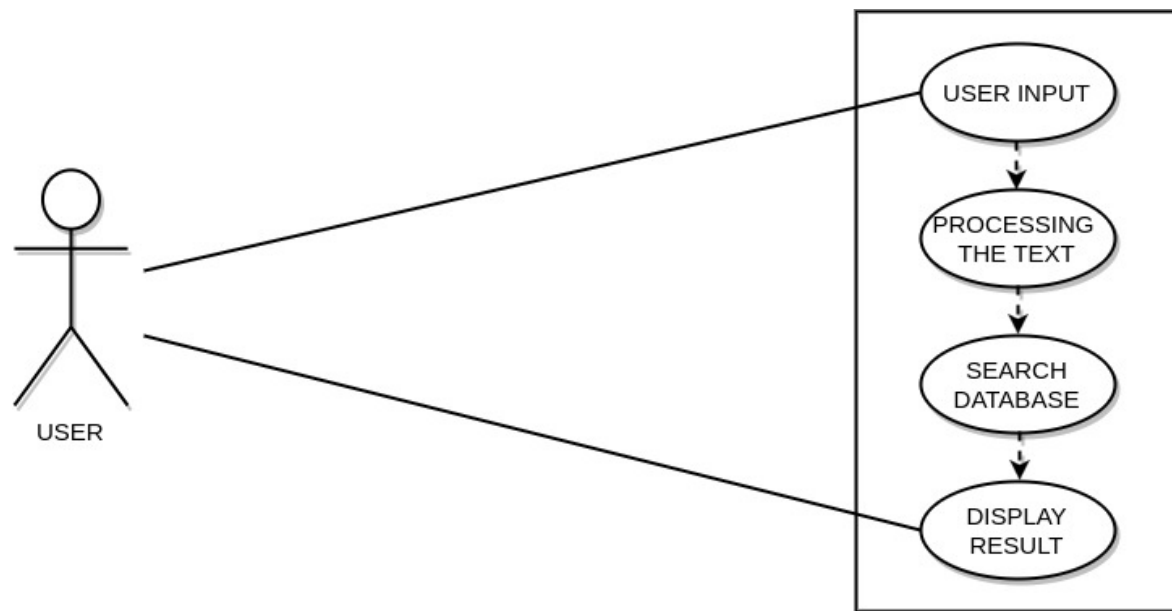


Figure 3.5: Use Case Diagram

### 3.5 FLOWCHART

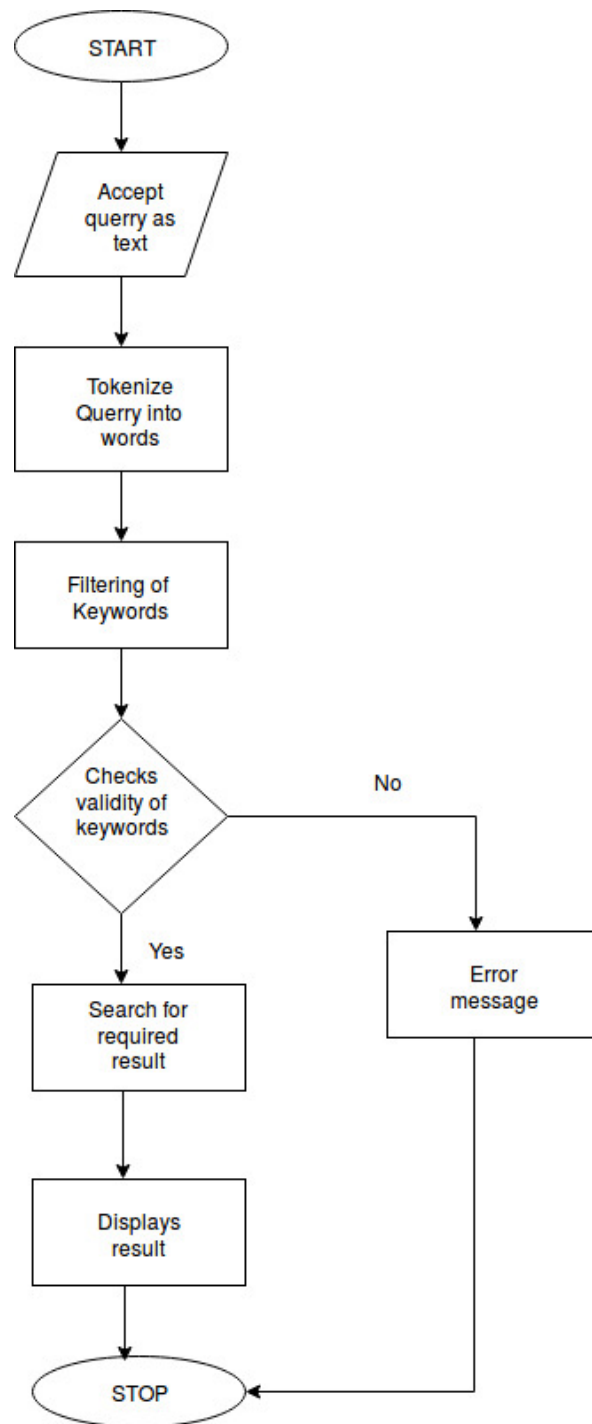


Figure 3.6: Flowchart

## Chapter 4

# IMPLEMENTATION

### 4.1 INTRODUCTION

Implementation is the stage in the project where the theoretical design is turned into a working system and is giving confidence on the new system for the users that it will work efficiently and effectively. It involves careful planning, investigation of the current system and its constraints on implementation, design of methods to achieve the change over, an evaluation, of change over methods. Apart from planning major task of preparing the implementation are education and training of users. The more complex system being implemented, the more involved will be the system analysis and the design effort required just for implementation. An implementation coordinating committee based on policies of individual organization has been appointed. The implementation process begins with preparing a plan for the implementation of the system. According to this plan, the activities are to be carried out, discussions made regarding the equipment and resources and the additional equipment has to be acquired to implement the new system. Implementation is the final and important phase.

The most critical stage in achieving a successful new system and in giving the users confidence that the new system will work and be effective. The system can be implemented only after thorough testing is done and if it found to working according to the specification.



This method also offers the greatest security since the old system can take over if the errors are found or inability to handle certain type of transactions while using the new system.

## **4.2 IMPLEMENTATION PLAN**

Implementation is the final and important phase. The most critical stage in achieving a successful new system and in giving the users confidence that the new system will work and be effective. The system can be implemented only after thorough testing is done and if it found to working according to the specification.

In this project, query input of our system will be tokenized word by word and from this list of tokens the NN,NNS,NNP,NNPS and CD will be identified. After filtering the keywords the necessary output will be displayed to the user.

This can be used in any field of work as it makes the searching kind of operations much more easier and faster than usual.The database can be designed specially and specifically for our particular field of work also.This software is simple user-friendly and can be installed in personal computers.

## Chapter 5

# TESTING

### 5.1 INTRODUCTION

Testing is the process of executing a program with the intent of finding any errors. A good test of course has the high probability of finding a yet undiscovered error. A successful testing is the one that uncovers a yet undiscovered error. A test is vital to the success of the system. System test makes a logical assumption that if all parts of the system are correct, then goal will be successfully achieved. The candidate system is subjected to a variety of tests online like responsiveness, its value, stress and security. A series of tests are performed before the system is ready for user acceptance testing.

The success of testing in revealing errors depends on the test cases. Testing should help locate errors, not just detect their presence. Test should be organized in a way that helps isolate errors.

Thus testing should be considered only one of the means to analyze the behavior of a system and should be integrated with other verification techniques in order to enhance our confidence in system qualities as much as possible.

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System

testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic.

Black-box testing is a method of software testing that tests the functionality of an application as opposed to its internal structures or workings. Specific knowledge of the application's code/internal structure and programming knowledge in general is not required. Test cases are built around specifications and requirements, i.e., what the application is supposed to do. It uses external descriptions of the software, including specifications, requirements, and designs to derive test cases. These tests can be functional or non-functional, though usually functional.

The test designer selects valid and invalid inputs and determines the correct output. There is no knowledge of the test object's internal structure. This method of test can be applied to all levels of software testing: unit, integration, functional, system and acceptance. It typically comprises most if not all testing at higher levels, but can also dominate unit testing as well.

## UNIT TESTING

In unit testing different modules are tested against the specification produced during the design of modules. Unit testing is essential for verification during coding phase. The aim is to test the internal logic of the modules. The tests carried out during the programming stage itself.

This enables the tester to detect errors in coding and logic that are contained within that module alone. Those resulting from the interaction between modules are initially avoided. Unit test comprises the set of performed prior to integration of the unit in to the entire project. Four categories of tests are performed on each unit.

**Functional test:** The code is exercised with normal input values for which the expected results are shown, as well as boundary values and values on and just outside the functional boundaries and special values such as logically related inputs.

**Performance Test:** Performance test is done to determine the amount of execution time spent in various parts of the unit, program throughput and response time and device utilization by the program unit.

**Stress Test:** Stress test intentionally breaks the unit. This helps in learning about the strength and limitations of the program by examining the manner in which a program unit breaks.

**Structure Test:** Structure tests are used to test the internal logic of a program. The major activity involved in this is to decide which paths to exercise, deriving test to exercise those paths, determining the test coverage criteria to be used.

The unit testing we have to perform is the testing of laser spot detection module, messaging hardware module, serial message reception module message projection module. We perform all the four categories of unit testing on each module.

## Integration Testing

Integration testing focuses on the design and the construction of the software architecture. The data can be lost across the interface or one module can pose an adverse effect on another. The sub functions when combined may not produce the major function. Integration testing is a systematic technique for the program structure, while at the same conducting test to uncover errors associated with the interface. In this test, groups of the program modules are tested together to determine if they interface properly. Two types of integration testing are:

**Top down Integration:** This method is an incremental approach to one construction of program structure. Modules are integrated by moving downward through the control hierarchy, beginning with the main program module. The modules subordinates to the main program module are incorporated into the structure in either a depth first or breadth first manner.

**Bottom up Integration:** This method begins the construction and testing with the modules at the lowest level in the program structure. Since the modules are integrated

from the bottom up, processing required for modules subordinate to a given level is always available and the need for the stubs is eliminated.

We make use of bottom up integration testing .we begin from the cam detection, ie, the most basic module in the project. We integrate the cam detection and the laser spot detection module .We make sure that the cam is detected, video is converted to frames and we detect laser spots in the frame.

The next integration to be done is integrating the messaging hardware and the serial message reception unit with the cam detection and laser spot detection. By this integration we make sure that once laser spot is detected, the message written using the messaging hardware is received serially by the system.

Finally we integrate the present system with the message projection unit. Here after receiving the message we display the message in a window and it's this window we finally project on to the wall. Thus we have integrated the whole system in the increasing order of hierarchy.

## **System Testing**

System testing is a critical element of quality assurance and represents the ultimate review of analysis, design and coding. When a system is developed it is hoped that it performs, manual procedures, computer operations and control.

System testing is the process of checking whether the developed system is working according to the objective and requirement. All testing accordance to the test conditions specified earlier. This will ensure that the test coverage meets the requirements and that testing is done in a systematic manner.

In system testing we check whether the integrated system works according to the objectives. We make sure that the system detects cam, captures the video and converts it to frames. We have to detect laser spots if any in the frames. Our next objective was to receive the message from the messaging hardware if a laser spot was detected. Then we c heck into

whether the received message is displayed on the window. Thus through system testing we make sure that all our objectives are met.

## **Validation Testing**

At the culmination of the integration testing, software is completely assembled as a package; interfacing errors have been recovered and corrected and a final series of a software tests, validation test, begins. Validation testing can be defined in many ways but a simple definition is that validation succeeds when the software functions in a manner that can be reasonably expected by the customer. Verification validation is the process of checking that a software system meets specification and that it fulfils its intended purpose. Validation ensures that the product actually meets the users need, and that the specifications were correct in the first place, while verification is ensuring that the product has been built according to the requirements and design specifications.

In validation testing we check into whether the requirements are correctly met. To make sure that the requirements are properly met we check several things in the system. one case is we have to check if only the laser spots are detected. if any other red colored spots are detected we have to correct it.

The next case we have to check whether message is read from arduino board only after the message is written to it through the messaging hardware.ie we make sure a text message with some content is received and not a blank message.

Another case is that we have to check whether message window is displayed only if the spot is detected .displaying message window before detecting a laser spot doesn't meet or actual requirement. We have to check into all these cases and correct if the meet the actual requirements.

## Acceptance Testing

User acceptance testing of the system is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with the prospective system at that time of development and making change whenever required. Acceptance testing is performed to demonstrate to the client, on the real-life data of the client, the operation of the system.

In acceptance testing we let a user work with the system and check into their full satisfaction. We perform the system working with real life data. First step is to take the template of the laser spot from the environment we are to work with. Then we let the user to enter the text message in the keypad. The user can see the message in the LCD while he or she types just as in a mobile phone.

Instead of using a send button we let the user point the laser beam to a wall. Once the laser spot is detected we display the message on the messaging window. the user can view the message window in the monitor as well as view it on the wall to the position where the laser spot was shown.

## 5.2 TEST CASES

A test case in software engineering is a set of conditions or variables under which a tester will determine whether an application or software system is working correctly or not. It is the mechanism for determining whether an application or software

## Chapter 6

# CONCLUSION

### 6.1 ADVANTAGES AND DISADVANTAGES OF AURO

#### 6.1.1 Advantages

The main advantage of our project is that we have developed a personal assistant for the users.

1. We can make use of this project in many areas of work such as educational field, medical field.
2. We could make our searching operations more faster and easier.
3. This project gives us a simpler user- friendly system that acts as a question-answering system

#### 6.1.2 Disadvantages

The main disadvantage of our project is that we have developed a personal assistant that can accept text in English language only. Some of the other disadvantages are listed below:



1. It can only accept text inputs and not voice or gestures as inputs.
2. Proper checking should be done to ensure that the queries are answered in a right way or else it can be a inconvenience to the user.
3. Proper internet connection is required for efficient operation of the system.
4. Auro is not fully trained to respond to 100% correct answer.

## **6.2 FUTURE SCOPE**

Some of the main future scope is that this project can be further extended to one which can accept voice and gestures as inputs along with text in another languages as well. In this way this project can become a good help for the users especially the disabled ones and people who are not familiar with English language.

Also this project can be again extended to level in which we can upload images through the user interface window and image searches can be done which can help users in an advanced way. One of future enhancement refers to the modification or improvements to the currently developed system. System enhancement may be required if there is change in organizational requirements of the user priorities. To carry out these changes the system have to be reevaluated, programs need to be modified and then again tested for user acceptance. Future developements can be easily integrated to the system.

## APPENDIX A

### Main.js

```
(function() {
    var Message;
    Message = function(arg) {
        this.text = arg.text, this.message_side = arg.message_side;
        this.draw = function(_this) {
            return function() {
                var $message;
                $message = $($('.message_template').clone().html());
                $message.addClass(_this.message_side).find('.text').html(_this.text);
                $('.messages').append($message);
                return setTimeout(function() {
                    return $message.addClass('appeared');
                }, 0);
            };
        }(this);
        return this;
    };
    $(function() {
        var lastSend;
        var getMessageText, message_side;
        message_side = 'right';
        getMessageText = function() {
            var $message_input;
            $message_input = $('.message_input');
            return $message_input.val();
        };
    });
});
```

```

sendMessage = function(text) {
    var $messages, message;
    if (text.trim() === '') {
        return;
    }
    $('.message_input').val('');
    $messages = $('.messages');
    message_side = message_side === 'left' ? 'left' : 'left';
    message = new Message({
        text: text,
        message_side: message_side
    });
    message.draw();
    return $messages.animate({
        scrollTop: $messages.prop('scrollHeight')
    }, 300);
};

receiveMessage = function(text) {
    var $messages, message;
    if (text.trim() === '') {
        return;
    }
    $('.message_input').val('');
    $messages = $('.messages');
    message_side = message_side === 'right' ? 'right' : 'right';
    message = new Message({
        text: text,
        message_side: message_side
    });
    message.draw();

```

```

        return $messages.animate({
            scrollTop: $messages.prop('scrollHeight')
        }, 300);
    };
    // $('#send_message').click(function(e) {
    //     lastSend = $('#message_input').val();
    //     return receiveMessage(getMessageText());
    // });
    $('#message_input').keyup(function(e) {
        lastSend = $('#message_input').val();
        if (e.which === 13) {
            var user_data=lastSend;
            console.log(user_data);
            RequestAPI(user_data);
            return receiveMessage(getMessageText());
        }
    });
    sendMessage('Hello User! What would you like to know?');
    $('#send_button').on('click',function(){
        var user_data=lastSend;
        console.log(user_data);
        RequestAPI(user_data);
    });
});

}.call(this))
function RequestAPI(request_data){
    myURL = 'https://auro-api.herokuapp.com/api/?q=' + request_data;
    var api_reply;
    var ourRequest = new XMLHttpRequest();
    ourRequest.open('GET', myURL);

```

```
ourRequest.onreadystatechange=function(){
    if(this.readyState==4 && this.status==200){
        api_reply=this.responseText;
        // console.log(api_reply);
        sendMessage(api_reply);
    }
}
ourRequest.open('GET', myURL, true);
ourRequest.send();
}
```

## Code for User Interface Window

```
<!DOCTYPE html>

<html>

<head>
  <title>AURO</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">

  <link rel="stylesheet"
    href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">

  <script src="https://code.jquery.com/jquery-3.2.1.min.js"
    integrity="sha256-hwg4gsxgFZh0sEEamdOYGBf13FyQuiTwlAQgxVSNgt4="
    crossorigin="anonymous">
  </script>

  <link
    href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">

  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js">
  </script>

  <link rel="shortcut icon" type="image/x-icon" href="./res/logo.png" />

  <style type="text/css">
    * {
      box-sizing: border-box;
    }
    body {
```

```
background-color: #edeff2;
font-family: "Calibri", "Roboto", sans-serif;
}
.navbar {
padding-top: 7px;
padding-bottom: 7px;
border: 0;
border-radius: 0;
margin-bottom: 0;
font-size: 16spx;
letter-spacing: 5px;
}
.navbar-nav li a:hover {
color: #1abc9c !important;
}
.chat_window {
position: absolute;
width: calc(100% - 20px);
max-width: 800px;
height: 500px;
border-radius: 10px;
background-color: #fff;
left: 50%;
top: 50%;
transform: translateX(-50%) translateY(-50%);
box-shadow: 0 10px 20px rgba(0, 0, 0, 0.15);
background-color: #f8f8f8;
overflow: hidden;
}
```

```
.top_menu {  
    background-color: #fff;  
    width: 100%;  
    padding: 20px 0 15px;  
    box-shadow: 0 1px 30px rgba(0, 0, 0, 0.1);  
}
```

```
.top_menu .buttons {  
    margin: 3px 0 0 20px;  
    position: absolute;  
}
```

```
.top_menu .buttons .button {  
    width: 16px;  
    height: 16px;  
    border-radius: 50%;  
    display: inline-block;  
    margin-right: 10px;  
    position: relative;  
}
```

```
.top_menu .buttons .button.close {  
    background-color: #f5886e;  
}
```

```
.top_menu .buttons .button.minimize {  
    background-color: #fdbf68;  
}
```

```
.top_menu .buttons .button.maximize {
```



```
        background-color: #a3d063;
    }

    .top_menu .title {
        text-align: center;
        color: #bcbdc0;
        font-size: 20px;
    }

    .messages {
        position: relative;
        list-style: none;
        padding: 20px 10px 0 10px;
        margin: 0;
        height: 347px;
        overflow: scroll;
    }

    .messages .message {
        clear: both;
        overflow: hidden;
        margin-bottom: 20px;
        transition: all 0.5s linear;
        opacity: 0;
    }

    .messages .message.left .avatar {
        background-color: #f5886e;
        float: left;
    }
}
```

```
.messages .message.left .text_wrapper {  
    background-color: #ffe6cb;  
    margin-left: 20px;  
}  
  
.messages .message.left .text_wrapper::after,  
.messages .message.left .text_wrapper::before {  
    right: 100%;  
    border-right-color: #ffe6cb;  
}  
  
.messages .message.left .text {  
    color: #c48843;  
}  
  
.messages .message.right .avatar {  
    background-color: #fdbf68;  
    float: right;  
}  
  
.messages .message.right .text_wrapper {  
    background-color: #c7eafc;  
    margin-right: 20px;  
    float: right;  
}  
  
.messages .message.right .text_wrapper::after,  
.messages .message.right .text_wrapper::before {  
    left: 100%;
```

```
        border-left-color: #c7eafc;
    }

    .messages .message.right .text {
        color: #45829b;
    }

    .messages .message.appeared {
        opacity: 1;
    }

    .messages .message .avatar {
        width: 60px;
        height: 60px;
        border-radius: 50%;
        display: inline-block;
    }

    .messages .message .text_wrapper {
        display: inline-block;
        padding: 20px;
        border-radius: 6px;
        width: calc(100% - 85px);
        min-width: 100px;
        position: relative;
    }

    .messages .message .text_wrapper::after,
    .messages .message .text_wrapper:before {
        top: 18px;
```

```
border: solid transparent;
content: " ";
height: 0;
width: 0;
position: absolute;
pointer-events: none;
}

.messages .message .text_wrapper::after {
border-width: 13px;
margin-top: 0px;
}

.messages .message .text_wrapper::before {
border-width: 15px;
margin-top: -2px;
}

.messages .message .text_wrapper .text {
font-size: 18px;
font-weight: 300;
}

.bottom_wrapper {
position: relative;
width: 100%;
background-color: #fff;
padding: 20px 20px;
position: absolute;
bottom: 0;
```

```
}

.bottom_wrapper .message_input_wrapper {
    display: inline-block;
    height: 50px;
    border-radius: 25px;
    border: 1px solid #bcbdc0;
    width: calc(100% - 160px);
    position: relative;
    padding: 0 20px;
}

.bottom_wrapper .message_input_wrapper .message_input {
    border: none;
    height: 100%;
    box-sizing: border-box;
    width: calc(100% - 40px);
    position: absolute;
    outline-width: 0;
    color: gray;
}

.bottom_wrapper .send_message {
    width: 140px;
    height: 50px;
    display: inline-block;
    border-radius: 50px;
    background-color: #a3d063;
    border: 2px solid #a3d063;
    color: #fff;
```

```
        cursor: pointer;
        transition: all 0.2s linear;
        text-align: center;
        float: right;
    }

    .bottom_wrapper .send_message:hover {
        color: #a3d063;
        background-color: #fff;
    }

    .bottom_wrapper .send_message .text {
        font-size: 18px;
        font-weight: 300;
        display: inline-block;
        line-height: 48px;
    }

    .message_template {
        display: none;
    }

    .container-fluid {
        padding-top: 5px;
        padding-bottom: 5px;
    }
</style>
```

</head>

<body>

```

<nav class="navbar navbar-default">
  <div class="container">
    <div class="navbar-header">
      <a class="navbar-brand avatar" href="#">
        </a>
      </div>
    </div>
  </div>
</nav>
<div class="chat_window">
  <div class="top_menu">
    <div class="title">AURO - The Personal Digital Assistant</div>
  </div>
  <ul class="messages"></ul>
  <div class="bottom_wrapper clearfix">
    <div class="message_input_wrapper">
      <input class="message_input" placeholder="Type your message here..."
      id="user_input"/>
    </div>
    <div class="send_message" id="send_button">
      <div class="icon"></div>
      <div class="text">Send</div>
    </div>
  </div>
</div>
<div class="message_template">
  <li class="message">
    <div class="avatar" ></div>
  </li>
</div>

```

```
<div class="text_wrapper">
    <div class="text"><p id="reply"></p></div>
</div>
</li>
</div>

<footer class="container-fluid text-center">
    <p align="center">A MiniProject by
<a href="https://www.github.com/SooluThomas" target="_blank">Soolu</a>,
<a href="https://github.com/ginu123" target="_blank">Ginu</a> and
<a href="https://www.facebook.com/greeshma.mathew.79"
target="_blank">Greeshma</a>
</p>
</footer>

<script src = "../js/main.js"></script>
</body>

</html>
```



## Code for API

```
# -*- coding: utf-8 -*-
from __future__ import unicode_literals

from django.shortcuts import render
from django.http import HttpResponse

import wikipedia
import wolframalpha

from nltk.tokenize import PunktSentenceTokenizer
import nltk

from tensorflow.examples.tutorials.mnist import input_data
mnist = input_data.read_data_sets("MNIST_data/", one_hot=True)
import tensorflow as tf

def tokenStr():
    filename = "./Files/userinput.txt"
    filename1 = "./Files/dictionary.txt"
    user = input("Enter Data: ")

    with open (filename , "w") as fd:
        fd.write(user)

    fd = open(filename ,"r")
    print(fd.read())
    fd.close()

    fd = open(filename , "r")
```

```

fd1 = open(filename1 , "r")
train_text = fd.read()
# sample_tect = fd1.read()

custom_sent_tokenizer = PunktSentenceTokenizer(train_text)

tokenized = custom_sent_tokenizer.tokenize(train_text)

def process_content():
    try:
        for i in tokenized[:5]:
            words = nltk.word_tokenize(i)
            tagged = nltk.pos_tag(words)
            print(tagged)
            with open("./Files/tokenized.txt", "w") as fd,
            open(filename) as fd1, open("./Files/comp.txt") as fd2:
                # myWords = set(line.split(',') for line in fd1.readlines())
                with open("./Files/newtokens.txt", "w") as fd3:
                    for line in tagged:
                        print(line)
                        freq1 = line[1]
                        f.write(line + "\n")
            for line1 in fd2:
                freq= line1[1]
                if freq in freq1:
                    print(freq)
                    # fd.write(myWords)

            fd.close()
chunkGram = r"""Chunk: {<.*>+}}<VB.?|IN|DT|TO>+{

```

```

        chunkParser = nltk.RegexpParser(chunkGram)
        chunked = chunkParser.parse(tagged)
        print(chunked)

    except Exception as e:
        print(str(e))

process_content()
fd1.close()
fd.close()

def index(request):
    answer = newFn(request.GET.get('q', ''))
    return HttpResponse(answer)

def training():
    #Initializing

    x = tf.placeholder(tf.float32, [None, 784])
    W = tf.Variable(tf.zeros([784, 10]))
    b = tf.Variable(tf.zeros([10]))
    y = tf.nn.softmax(tf.matmul(x, W) + b)

    #Training

    cross_entropy = tf.reduce_mean(-tf.reduce_sum(y_ * tf.log(y), reduction_
    train_step = tf.train.GradientDescentOptimizer(0.5).minimize(cross_entropy)
    sess = tf.InteractiveSession()
    tf.global_variables_initializer().run()

```

```

for _ in range(1000):
    batch_xs, batch_ys = mnist.train.next_batch(100)
    sess.run(train_step, feed_dict={x: batch_xs, y_: batch_ys})

#Evaluating Model

correct_prediction = tf.equal(tf.argmax(y,1), tf.argmax(y_,1))
accuracy = tf.reduce_mean(tf.cast(correct_prediction, tf.float32))
print(sess.run(accuracy, feed_dict={x: mnist.test.images, y_: mnist.test

if __name__ == '__training__':
    parser = argparse.ArgumentParser()
    parser.add_argument('--data_dir', type=str,
                        default='/tmp/tensorflow/mnist/input_data',
                        help='Directory for storing input data')
    FLAGS, unparsed = parser.parse_known_args()
    tf.app.run(main=main, argv=[sys.argv[0]] + unparsed)

expected_questions = {'what is your name': 'Auro',
    'who developed you': 'Ginu, Greeshma and Soolu',
    'who developed you?': 'Ginu, Greeshma and Soolu',
    'who created you': 'Ginu, Greeshma and Soolu',
    'who created you?': 'Ginu, Greeshma and Soolu',
    'when were you developed': 'I am still under delopment',
    'what is auro': 'I am a personal Digital
                    Assistant developed by Ginu, Greeshma and Soolu',
    'who are you': 'I am Auro, the Personal Assistant',
    'do you know me': 'Very well...!! I am afraid to
                    reply to that. I am Auro, the Personal Assistant',
    'do you know me?': 'No, I\'m afraid you have me

```

```

        at a disadvantage there. I am Auro, the Personal Assistant
    'how old are you?': 'A few days ago',
    'how old are you': 'A few days ago',
    'what is your age?': 'A few days',
    'what is your age': 'A few days',
    'when was i born': '21st April 2017. My Project Review is still going on',
    'me': 'Auro',
    'you': 'Auro',
    'who is Soolu Thomas': 'Soolu, Ginu and Greeshma Developed me!',
    'who is Greeshma': 'Soolu, Ginu and Greeshma Developed me!',
    'who is Ginu': 'Soolu, Ginu and Greeshma Developed me!'
    'What programing language was used to develop you?': 'Python',
    'Development': 'Python',
    'what are you': 'A Personal Digital Assistant',}

```

```

def newFn(question):
    try:
        input = question
        input = input.strip(' ').lower()
        app_id = "587E79-677J3JTXE3"
        client = wolframalpha.Client(app_id)
        try:
            res_exp = expected_questions[input]
            if input not in res_exp :
                res = client.query(input)
                answer = next(res.results).text
                return answer
            else:
                answer = res_exp

```

```
        return answer

    except:

        res = client.query(input)
        answer = next(res.results).text
        return answer

except:

    wikipedia.set_lang("en")
    try:

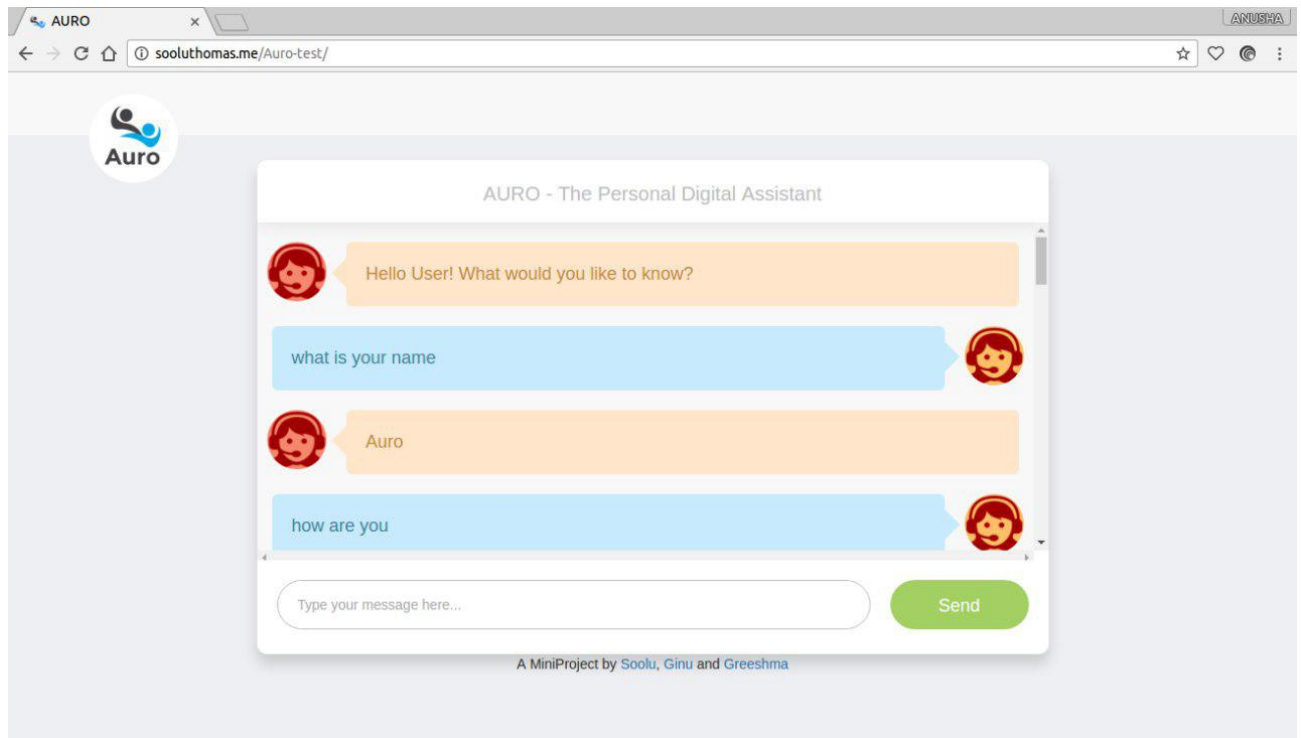
        return "Wikipedia says: " + wikipedia.summary(input, sen
    except wikipedia.exceptions.DisambiguationError as e:

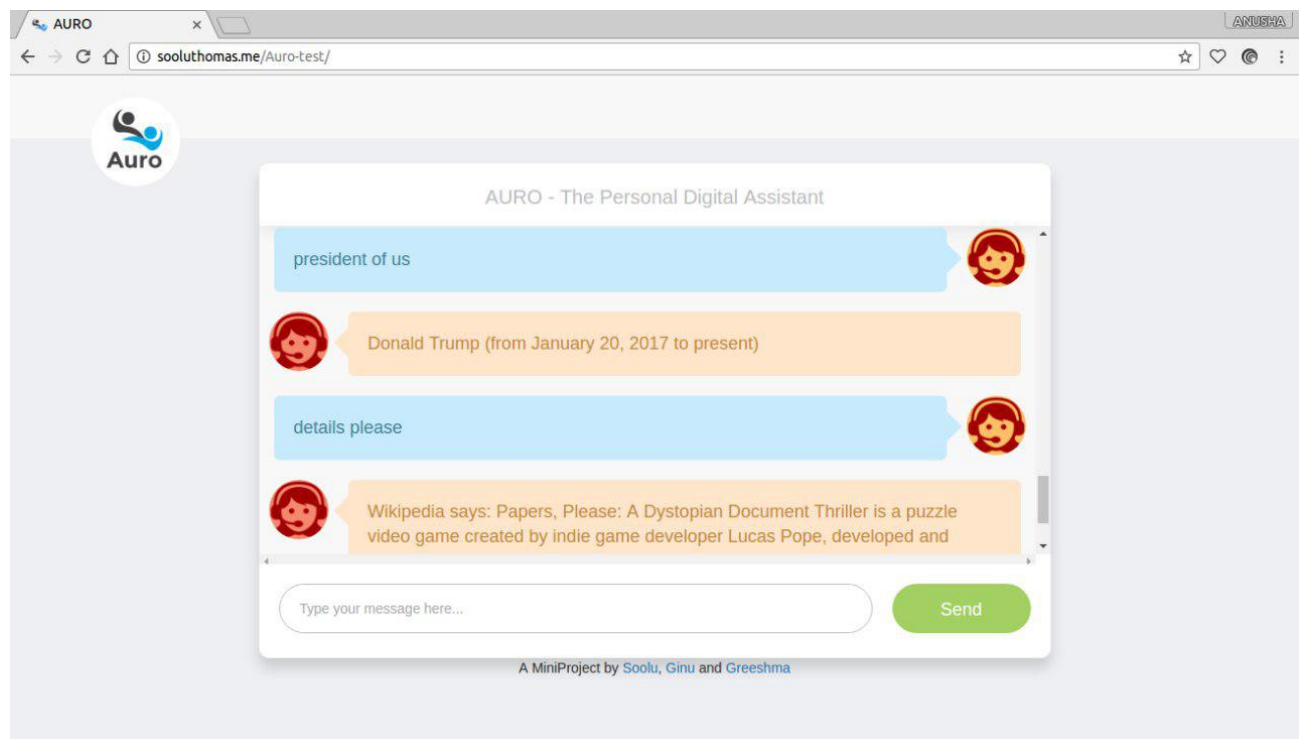
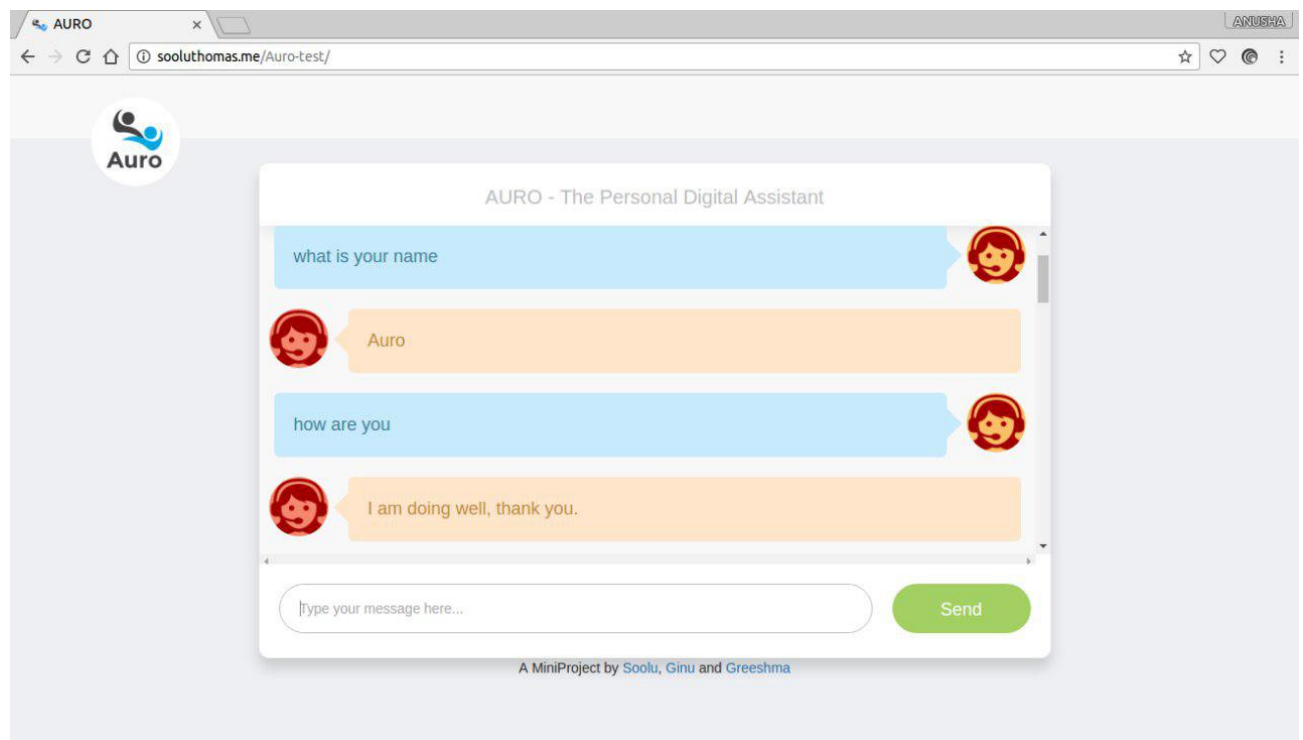
        return "Ambiguos question! Be more specific"
    except wikipedia.exceptions.PageError as pe:

        return "I am sorry. I didn't get your question"
```

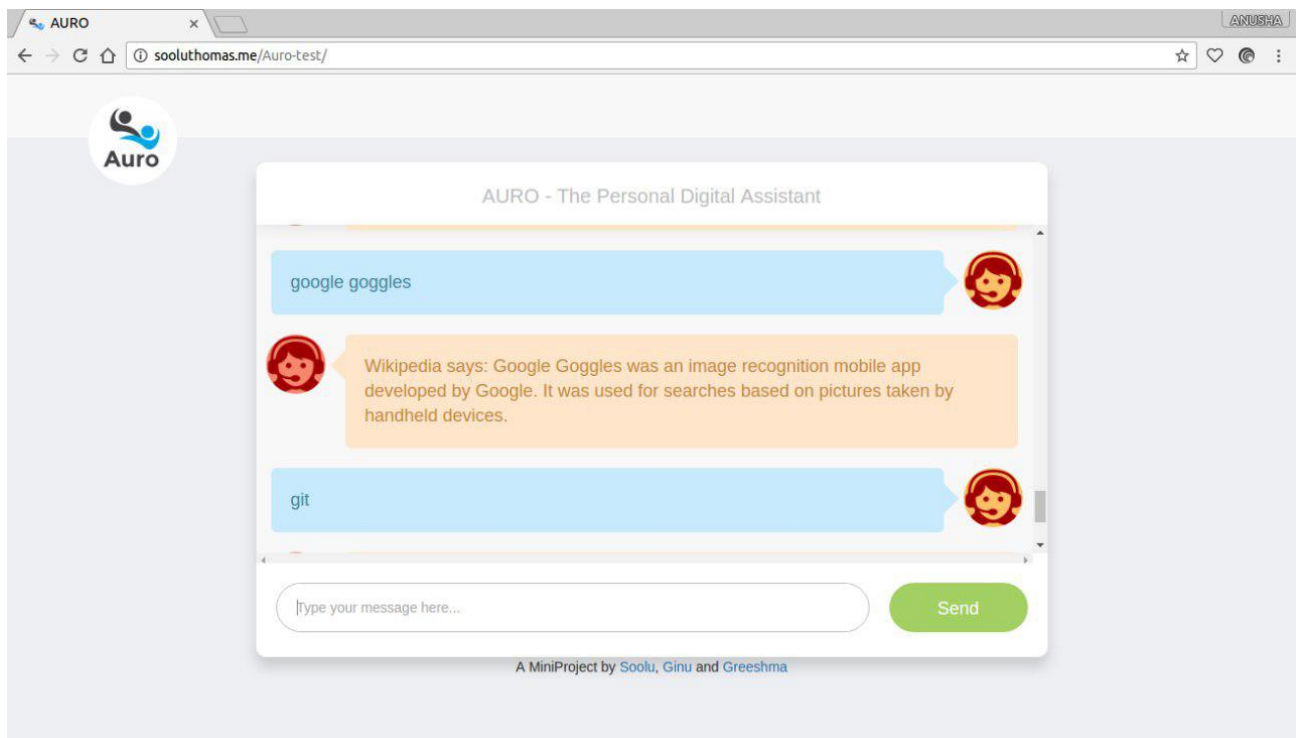
## APPENDIX B

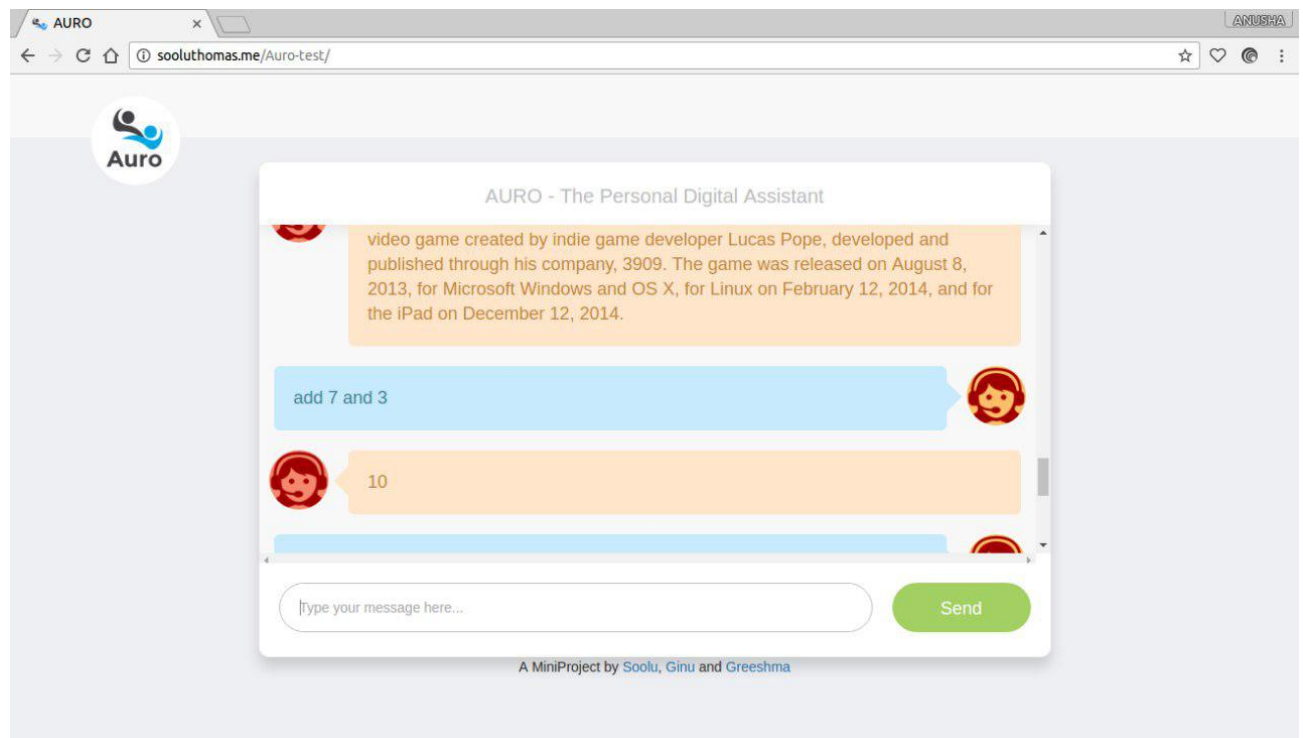
### screenshots

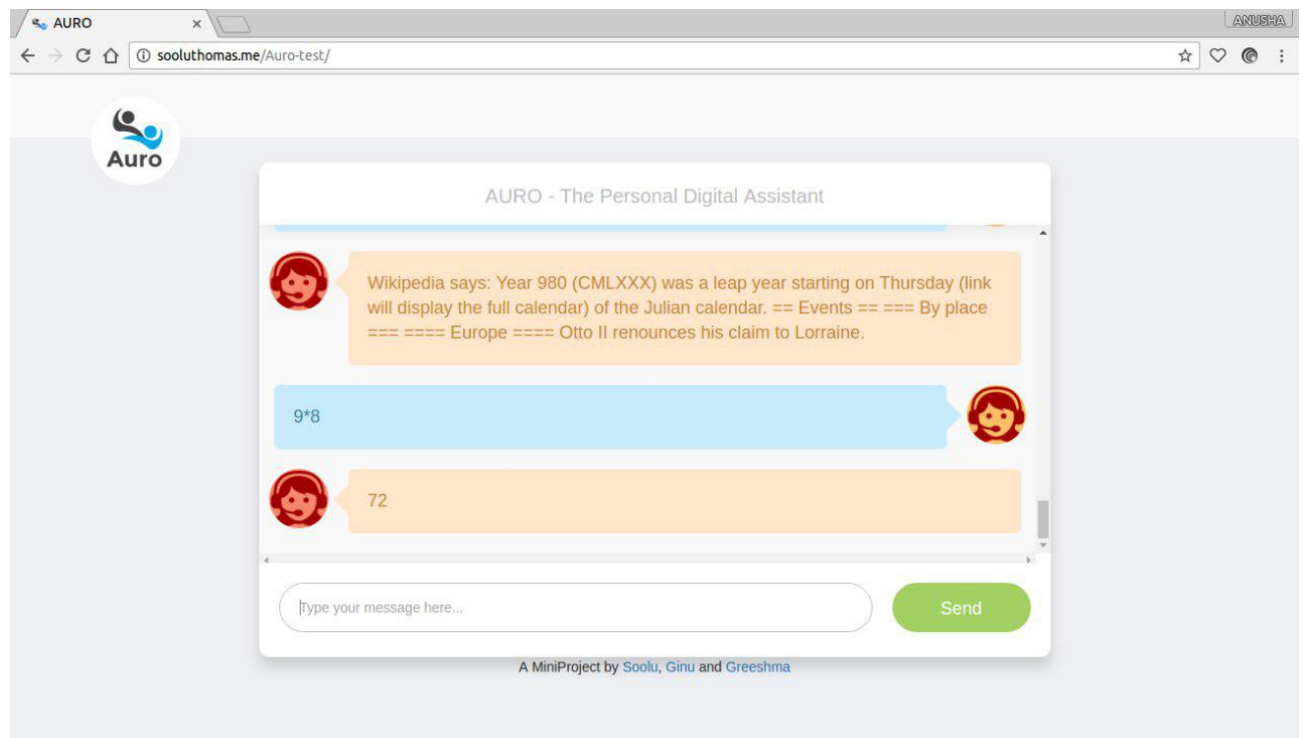












# References

- [1] [IEEE Standard 181-1998]: The standard followed by the current SRS
- [2] Roger.S.Pressman, Software Engineering: A Practitioner's Approach, McGraw Hill International edition, Seventh edition.
- [3] Synthesis Lectures on Human Language Technologies, Morgan & Claypool, 2015
- [4] Natural Language Processing Gobinda G. Chowdhury Dept. of Computer and Information Sciences University of Strathclyde, Glasgow G1 1XH, UK
- [5] Core Python Programming Wesley J. Chun Publisher: Prentice Hall PTR First Edition December 14, 2000
- [6] Website Reference for Natural Language Processing in Python. Link: [\*http://pythonprogramming.net\*](http://pythonprogramming.net)
- [7] Artificial Intelligence for Humans, Volume 3: Deep Learning and Neural Networks, Jeff Heaton, Copyright 2015 by Heaton Research, Inc.
- [8] Natural Language ToolKit for NLP in Python Link: [\*http://www.nltk.org\*](http://www.nltk.org)