A Project report on

**VOICE BASED EMAIL FOR VISUALLY CHALLENGED PEOPLE**

**submitted in partial fulfillment for the award of the degree of**

**BACHELOR OF TECHNOLOGY**

in

**Computer Science and Engineering**

by

**J. Devi Durga (17A81A0570)**

**Ch. Gopi (17A81A0561)**

**G. Prem Chandu (17A81A0565)**

**Ch. Pavan (17A81A0560)**

**Under the Esteemed Supervision of**

**Dr.D.JAYA KUMARI**

**M.TECH, Ph.D,**

**Professor & HOD**

****

**Department of Computer Science and Engineering (Accredited by NBA) SRI VASAVI ENGINEERING COLLEGE (Autonomous) (Affiliated to JNTU, Kakinada)**

**Pedatadepalli, Tadepalligudem-534101, A.P**

**2020-2021**

**SRI VASAVI ENGINEERING COLLEGE**

****

**DEPARTMENT OF**

**COMPUTER SCIENCE AND ENGINEERING**

**CERTIFICATE**

This is to certify that the project Report entitled "**VOICE BASED EMAIL FOR VISUALLY CHALLENGED PEOPLE**" submitted by **J. Devi Durga (17A81A0570), Ch. Gopi (17A81A0561), G. Prem Chandu (17A81A0565), Ch. Pavan (17A81A0560)** under the guidance and supervision in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering Department of Sri Vasavi Engineering College during 2020-21.

**Name of Project Guide Head of the Department Dr.K.SHIRIN BHANU Dr.D.JAYA KUMARI M.Tech, Ph.D, M.Tech, Ph.D, Professor Professor & HOD**

**External Examiner**

**DECLARATION BY THE STUDENTS**

We hereby declare that the project report entitled “**VOICE BASED EMAIL FOR VISUALLY CHALLENGED PEOPLE**" submitted by us to Sri Vasavi Engineering College, Tadepalligudem, affiliated to JNTU Kakinada in partial fulfillment of the requirement for the award of the degree of **B.Tech Degree** in **Computer Science and Engineering** is a record of bonafide project work carried out by us under the guidance of **Dr.K.SHIRIN BHANU**, we further declare that the work reported in this project has not been submitted and will not be submitted either in part or in full, for the award of any other degree in this institute or any other institute on University.

**Project Associates:**

**J. Devi Durga (17A81A0570) Ch. Gopi (17A81A0561) G. Prem Chandu (17A81A0565) Ch. Pavan (17A81A0560)**

**ACKNOWLEDGEMENT**

First and foremost we sincerely salute to our esteemed institute Sri Vasavi Engineering college for giving this golden opportunity for fulfilling our warm dream to become engineer.

We consider it as our privilege to express our gratitude and respect to all those who guided inspired and helped us in completion of this project we wish to express our sincere gratitude to our project guide, Dr.K.SHIRIN BHANU, Professor, Dept of Computer Science and Engineering, SRI VASAVI ENGINEERING COLLEGE, Pedatadepalli, for this timely cooperation and her valuable suggestions while carrying out this project.

We express our sincere thanks and heartful gratitude to Dr.D.Jaya kumari, HOD, Dept of Computer Science and engineering, SRI VASAVI ENGINEERING COLLEGE, Pedatadepalli, for permitting us to do our project.

We express our sincere thanks and heartful gratitude to Dr.Guduru VNSR Ratnakara Rao. PRINCIPAL of SRI VASAVI ENGINEERING COLLEGE, Pedatadepalli for supporting us during this work assisted us in this project work.

We would like to express our gratitude to our parents, friends those who helped to complete this course.

**Project Associates:**

**J.Devi Durga (17A81A0570)**

**Ch.Gopi (17A81A0561)**

**G.Prem Chandu(17A81A0565)**

**Ch.Pavan (17A81A0560)**

**ABSTRACT**

As the technology is enhancing, people are coming more closer to digital life and digital communication. There are many ways to communicate with others through internet in this new advanced era. Most of them are choosing the easiest way of communication i.e., electronic mail (E-mail).

E-mail is the technology that enables user to contact with others by sending mails and also helps in business world communication. There are people who cannot use these technologies because either they are illiterate or do not have ability to see the screen. So, to make this technology closer to visually challenged people, we proposed a Voice Based E-mail System.

This system provides them the facility of communication and make them much stronger and independent. This architecture will help blind people to access e-mail and other multimedia functions. Leaving behind the old techniques, this voice-based email system will be containing new technologies that will be easily acceptable by visually

**INDEX**

**CHAPTERNO. NAME OF THE TOPIC PAGENO**

CHAPTER 1 INRODUCTION 1-2 CHAPTER 2 LITERATURE SURVEY 3-4 CHAPTER 3 SYSTEM STUDY AND ANALYSIS 5-12

3.1 Existing System 6

3.2 Proposed System 6-7

3.3 Feasibility Study 7-8

3.3.1 Operational Feasibility 7

3.2.2 Economic Feasibility 7-8

3.3.3 Technical Feasibility 8

3.4 Requirements 8-11

3.4.1 Functional Requirements 8-9

3.4.2 Non-Functional Requirements 9-11

3.5 System Requirements 12

3.5.1 Hardware requirements 12

3.5.2 Software Requirements 12

3.5.3 Deployment Platform 12

CHAPTER 4 SYSTEM DESIGN 13-23

4.1 System Architecture Design 14

4.2 Design Diagrams using UML Approach 15

4.2.1 UseCase Diagram 15-17

4.2.2 Class Diagram 18

4.3 Detailed Design 19-23

4.3.1 Sequence Diagram 19

4.3.2 Activity Diagram 20

4.3.3 Component Diagram 21

4.3.4 Deployment Diagram 22-23

CHAPTER 5 SOFTWARE ENVIRONMENT 24-42 CHAPTER 6 IMPLEMENTATION 43-46

6.1 Modules 43

6.2 Module Description 43-46

CHAPTER 7 TESTING 47-52

7.1 Introduction 48

7.2 Testing Strategies 48-51

7.2.1 Unit Testing 48-49

7.2.2 Integration Testing 49-50

7.2.3 System Testing 50

7.2.4 White Box Testing 51

7.2.5 Black Box Testing 51

7.3 Testcases Design 52

CHAPTER 8 SCREENSHOTS 53-56

CHAPTER 9 CONCLUSION 57-58

CHAPTER 10 REFERENCES 59-61

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **TITLE** | **PAGE NO** |
| 4.1.1 | SystemArchitecture | 15 |
| 4.2.1.1 | Usecase diagram for Voice Based Email System | 16 |
| 4.2.2.1 | Class diagram for Voiced Based Email System | 18 |
| 4.3.1.1 | Sequence diagram for Voice Based Email System | 19 |
| 4.3.2.1 | Activity diagram for Voice Based Email System | 20 |
| 4.3.3.1 | Component diagram for Voice Based Email System | 21 |
| 4.3.4.1 | Deployment diagram for Voice Based Email System | 23 |
| 8.1 | Login page | 54 |
| 8.2 | Registration page | 54 |
| 8.3 | Home page | 55 |
| 8.4 | Compose page | 55 |
| 8.5 | Inbox page | 56 |
|  |  |  |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **TABLE NO** | **TITLE** | **PAGE NO.** |
| 7.3.1 | Testcase Design | 52 |

**CHAPTER-1**

**INTRODUCTION**

**1.INTRODUCTION**

Internet is considered as the most important means of information and has become de facto methods used in communication. Email is one of the most common form of communication. However, it is completely useless for visually impaired and illiterate people.

Currently available systems like screen readers TTS (Text-To- Speech) and ASR (Automatic Speech Recognition) does not provide full efficiency to the blind people to use internet. As nearly 285 million people worldwide are visually impaired so it is necessary to make internet facilities for communication usable for them.

Therefore, in this project we will be developing a voice based email system which will aid the visually impaired people who are naive to computer systems to use email facilities with ease. All the functions are based on simple voice commands making it very easy for any type of user to use this system.

**The main benefit of this system is that the use of keyboard is completely eliminated, the user will have to respond through voice and mouse click only.**

**CHAPTER-2**

**LITERATURE SURVEY**

**2.LITERATURE SURVEY**

The systems available nowadays uses screen readers which read information displayed on desktop or it prints information on Braille printer. ASR (Automatic speech recognizer) and TTS (text to speech) get used for converting speech to text and vice versa. Although these technologies are being improved continuously, some major problems still persist which make them unusable as a way of accessing email to a large segment of Blind people.

These systems have following drawbacks.

1) With the help of screen readers it is difficult for blind person to access E-mail system and computer operating easily because it has noisy audio interface.

2) ASR is still in development stage. In case of noisy environment performance of ASR degrade

3) Both ASR and TTS are highly language dependent. So the system developed for one language is not applicable to other.

4) Now a day’s mobile is very common word it is known to almost all peoples even school goers also use mobile. Moreover, tools and technologies above for the blind users are unavailable for mobile devices.

5) These systems are not very much useful for small scale application for E-mail.

6) These available systems require use of keyboard which is very difficult for blind people to recognize and remember characters of keyboard.

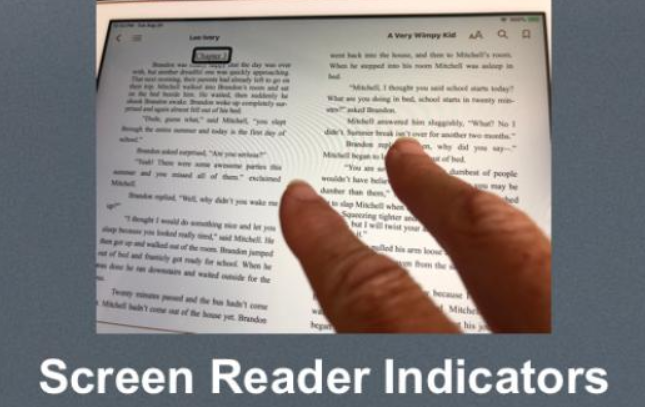
**** 

Fig1:Braille printer Fig2:screen readers

**CHAPTER -3**

**SYSTEM STUDY AND ANALYSIS**

**3.SYSTEM STUDY AND ANALYSIS**

**3.1 Existing System**

There are 4.1 billion email accounts created until 2014 and there will be estimated 5.2 billion accounts by end of 2018. This makes emails the most common form of communication. The most common mail services that we use in our day to day life cannot be used by visually challenged people because they do not provide any facility so that the person in front can hear out the content of the screen, As they cannot visualize what is already present on screen they cannot make out where to click in order to perform the required operations.

For a visually challenged person using a computer for the first time is not that convenient as it is for a normal user even though it is user friendly. Although there are screen readers available still these people face minor difficulties. Screen readers read out whatever content is there on the screen and to perform those actions the person will have to use keyboard shortcuts as mouse location cannot be traced by the screen readers. A user is new to computer can therefore not use this service as they are not aware of the key locations.

The screen readers read out the content in sequential manner and therefore user can make out the contents of the screen only if they are in basic HTML format. Thus the new advanced web pages which do not follow this paradigm in order to make the website more user-friendly only create extra hassles for these people. Our project aims to overcome these drawbacks.

**3.2 Proposed System**

Keeping in view all of the drawbacks of the existing system, goal of our project is to reduce limitations and problems in the current systems. In this system, we are trying to make a system for the blind people through which they can easily use an important feature such as email in a very interactive manner. Here, we are designing a system which will work on the voice commands and prompts for confirmations of actions. The user can also add attachments, create labels, etc in the system.

This system has following advantages:

1) Browser is used and done via a desktop application.

2) The system provides an intuitive, interactive and easy to use GUI that can be easily used by a blind user even if they are not computer literate.

3) The system help not only for blind user to access Email, but it may also help other sighted people who can’t type text due to illiteracy.

**3.3 FEASIBILITY STUDY**

An important outcome of preliminary investigation is the determination that the system request is feasible. This is possible only if it is feasible within limited resource and time. The different feasibilities that have to be analyzed are

∙ **Operational Feasibility**

∙ **Economic Feasibility**

∙ **Technical Feasibility**

**3.3.1 Operational Feasibility**

Operational Feasibility deals with the study of prospects of the system to be developed. This system operationally eliminates all the tensions of the Admin and helps him in effectively tracking the project progress. This kind of automation will surely reduce the time and energy, which previously consumed in manual work. Based on the study, the system is proved to be operationally feasible.

**3.3.2 Economic Feasibility**

Economic Feasibility or Cost-benefit is an assessment of the economic justification for a computer based project. As hardware was installed from the beginning & for lots of purposes thus the cost on project of hardware is low. Since the system is a network based, any number of employees connected to the LAN within that organization can use this tool from at anytime. The Virtual Private Network is to be developed using the existing resources of the organization. So the project is economically feasible.

**3.3.3 Technical Feasibility**

According to Roger S. Pressman, Technical Feasibility is the assessment of the technical resources of the organization. The system is developed for platform Independent environment. JavaScript, HTML5, PHP mail server, SMTP,POP3 and IMAP protocols are used to develop the system. The technical feasibility has been carried out. The system is technically feasible for development and can be developed with the existing facility

**3.4 REQUIREMENTS**

**3.4.1 Functional Requirements:**

In software engineering, a **functional requirement** defines a function of a software system or its component. A function is described as a set of inputs, the behavior, and outputs (see also software). Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define *what* a system is supposed to accomplish.

Functional requirements drive the application architecture of a system. A requirements analyst generates use cases after gathering and validating a set of functional requirements. Functional requirements may be technical details, data manipulation and other specific functionality of the project is to provide the information to the user.

The following are the Functional requirements of our system:

1. When user will visit our site he would first have to register in our website through registration form. User will be very well guided with the help of voice commands.

2. IVR (Interactive voice response): IVR is an advanced technology describes the interaction between the user and the system in the way of responding by using respective voice message.

3.Audio voice to further assist users on how to proceed. Annyang is a tiny Java script library that lets your visitors control your site with voice commands.

4.Responsive Voice is a HTML5-based Text-To-Speech library designed to add voice features to all smartphone, tablet and desktop devices.

5.PHP mail is the built in PHP function that is used to send emails. It’s a cost effective way of notifying users on important events. We can use it in a web application to easily send out emails using a configured SMTP server.

**3.4.2 Non-Functional Requirements**

In systems engineering and requirements engineering, a **non-functional requirement** is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors**.** The project non functional requirements include the following.

**Availability:** A system’s “availability” or “uptime” is the amount of time that is operational and available for use. Application can be made use of at any time in the system having PHP, HTML5, Javascript and its relative packages installed. As our system will be used by thousands of users at any time our system must be available always. If there are any cases of updations they must be performed in a short interval of time without interrupting the normal services made available to the users.

**Efficiency:** Specifies how well the software utilizes scarce resources: CPU cycles, disk space, memory, bandwidth etc. All of the above mentioned resources can be effectively used by performing most of the validations at client side and reducing the workload on server by using PHP, Javascript libraries, SMTP, POP3 and IMAP protocols which are used for implementation .

**Flexibility:** If the organization intends to increase or extend the functionality of the software after it is deployed, that should be planned from the beginning; it influences choices made during the design, development, testing and deployment of the system. New modules can be easily integrated to our system without disturbing the existing modules or modifying the logical database schema of the existing applications.

**Portability:** Portability specifies the ease with which the software can be installed on all necessary platforms, and the platforms on which it is expected to run. By using appropriate server versions released for different platforms our project can be easily operated on any operating system, hence can be said highly portable. This system can be run on any operating system including Windows, Linux and Mac.

**6**

**Scalability:** Software that is scalable has the ability to handle a wide variety of system configuration sizes. The nonfunctional requirements should specify the ways in which the system may be expected to scale up (by increasing hardware capacity, adding machines etc.). Our system can be easily expandable. Any additional requirements such as hardware or software which increase the performance of the system can be easily added. An additional server would be useful to speed up the application.

**Integrity:** Integrity requirements define the security attributes of the system, restricting access to features or data to certain users and protecting the privacy of data entered into the software. Certain features access must be disabled to normal users such as adding the details of files, searching etc which is the sole responsibility of the server. Access can be disabled by providing appropriate logins to the users for only access.

**Usability:** Ease-of-use requirements address the factors that constitute the capacity of the software to be understood, learned, and used by its intended users. A system that has high usability coefficient makes the work of the user easier.

**Performance:** The performance constraints specify the timing characteristics of the software. Making the application form filling process through online and providing the invigilation list information and examination hall list is given high priority compared to other services and can be identified as the critical aspect of the system

1. In our system introduced user specific search performance.

2. The query related search is effective it provide within short period results, so the speed of system is very high.

**Maintainability:**

Maintainability means fixing, updating, servicing and to modify the system or update the software for performance improvements or for the correction of faults. Maintenance is easy and economical.

**Security Requirements:**

Application requires user to login with his mail credentials as specified in code.

**Objective And Scope**

**Objective**

This project aims at developing an email system that will help even a naïve, visually impaired person to use the services for communication without previous training. The system does not require the use of keyboard. Instead it will work only on voice commands. This system can also be used by any normal person, for instance, by someone who is unable to read.

**Scope**

The proposed system will meet the needs of the user by providing following features:

* The visually impaired person can efficiently access the mail services.
* This system will focus more on the user friendliness of all types of people
* including normal, visually impaired as well as illiterate people.
* The system will be highly efficient and easily understandable
* The user can compose mail and listen to mail through voice commands
* The project can work on any system

**3.5 System Requirements**

**3.5.1 Hardware Requirements**

* 2GB RAM
* 250 GB HDD
* MICROPHONE
* SPEAKER

**3.5.2 Software Requirements**

* Windows platform, vs code editor
* Front End : HTML ,CSS, JAVASCRIPT
* Backend : php
* Database : MySQL
* Local Server :XAMPP

**3.5.3 Deployment platform**

* Laptops
* Mobiles

**CHAPTER-4**

**SYSTEM DESIGN**

**4.SYSTEM DESIGN**

**DESIGN**

DESIGN: The design of this project is divided into three phases as described below:

1. UI design: In this phase the UI or the user interface of the project is developed. That is, the designing of the web pages which the user will use to interact.
2. Database design: The database is considered to be the main pillars of every project. In our application, database is used to store user details such as name, age etc.
3. System design: In this phase a complete flow diagram of the working system is designed. This flow diagram will show the details of all the events like actions to be performed for an event.

**4.1 System Architecture Design**

A system Architecture is the conceptual model that defines the structure, behaviour, and more views of the system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviours of the system. A system architecture can consists of system components and the sub-systems developed, that will work together the overall system.

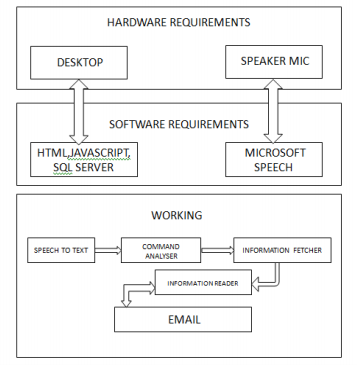


Fig 4.1.1 System Architecture

**4.2 Design Diagrams Using UML Approach**

UML is a method for describing the system architecture in detail using the blue print. UML represents a collection of best engineering practice that has proven successful in the modelling of large and complex systems. The UML is very important parts of developing object oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects. Using the helps UML helps project teams communicate explore potential designs and validate the architectural design of the software.

**4.2.1Usecase Diagram**

Use case diagram represents the functionality of the system. Use case focus on the behaviour of the system from external point of view. Actors are external

entities that interact with the system.

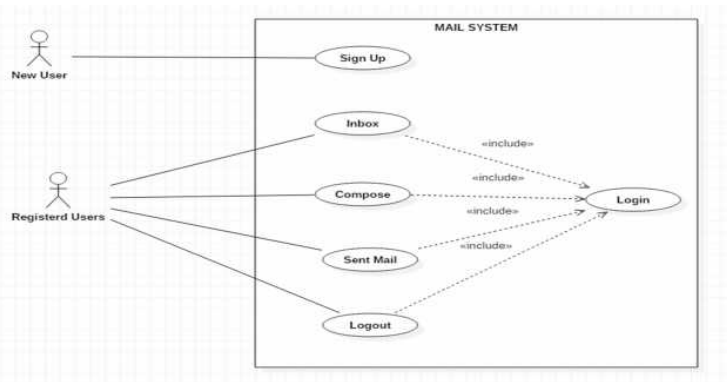


Fig 4.2.1.1 usecase diagram for Voice Base Email System

**Usecases**:

A use case describes a sequence of actions that provide something of measurable value to an actor and is drawn as a horizontal ellipse.

**Actors:**

An actor is a person, organization, or external system that plays a role in one or more interactions with the system.

System boundary boxes (optional): A rectangle is drawn around the use cases, called the system boundary box, to indicate the scope of system. Anything within the box represents functionality that is in scope and anything outside the box is not. Four relationships among use cases are used often in practice.

**Include:**

In one form of interaction, a given use case may include another. "Include is a Directed Relationship between two use cases, implying that the behaviour of the included use case is inserted into the behaviour of the including use case. The first use case often depends on the outcome of the included use case. This is useful for extracting truly common behaviours from multiple use cases into a single description. The notation is a dashed arrow from the including to the included use case, with the label "«include»". There are no parameters or return values. To specify the location in a flow of events in which the base use case includes the behaviour of another, you simply write include followed by the name of use case you want to include, as in the following flow for track order.

**Extend:**

In another form of interaction, a given use case (the extension) may extend another. This relationship indicates that the behaviour of the extension use case may be inserted in the extended use case under some conditions. The notation is a dashed arrow from the extension to the extended use case, with the label "«extend»". Modelers use the «extend» relationship to indicate use cases that are "optional" to the base use case.

**Generalization:**

In the third form of relationship among use cases, a generalization/specialization relationship exists. A given use case may have common behaviours, requirements, constraints, and assumptions with a more general use case. In this case, describe them once, and deal with it in the same way, describing any differences in the specialized cases. The notation is a solid line ending in a hollow triangle drawn from the specialized to the more general use case (following the standard generalization notation.

**Associations:**

Associations between actors and use cases are indicated in use case diagrams by solid lines. An association exists whenever an actor is involved with an interaction described by a use case. Associations are modelled as lines connecting use cases and actors to one another, with an optional arrowhead on one end of the line. The arrowhead is often used to indicating the direction of the initial invocation of the relationship or to indicate the primary actor within the use case.

**Identified UseCases:**

The “user model view” encompasses a problem and solution from the preservative of those individuals whose problem the solution addresses. The view presents the goals and objectives of the problem owners and their requirements of the solution. This view is composed of “Usecase diagrams”. These diagrams describe the functionality provided by a system to external integrators. These diagrams contain actors, use cases, and their relationships.

**4.2.2 Class Diagram**

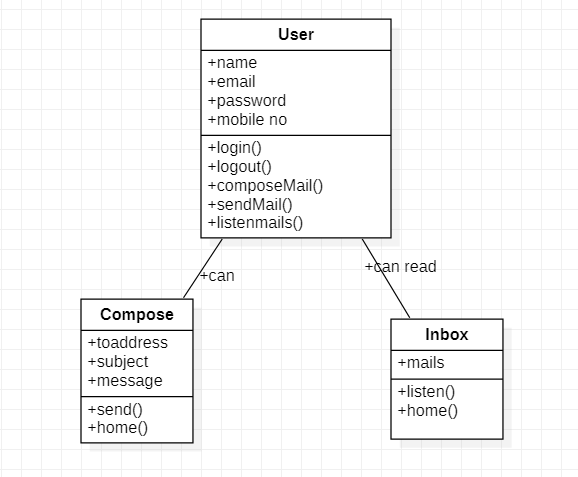
****

Fig 4.2.2.1 class diagram for voice based email system

Class-based Modelling, or more commonly class-orientation, refers to the style of object-oriented programming in which inheritance is achieved by defining classes of objects; as opposed to the objects themselves (compare Prototype-based programming). The most popular and developed model of OOP is a class-based model, as opposed to an object based model. In this model, objects are entities that combine state (i.e., data), behaviour (i.e., procedures, or methods) and identity (unique existence among all other objects). The structure and behaviour of an object are defined by a class, which is a definition, or blueprint, of all objects of a specific type. An object must be explicitly created based on a class and an object thus created is considered to be an instance of that class. An object is similar to a structure, with the addition of method pointers, member access control, and an implicit data member which locates instances of the class (i.e. actual objects of that class) in the class hierarchy (essential for runtime inheritance features).

**4.3 Detailed Design**

**4.3.1 Sequence Diagram**

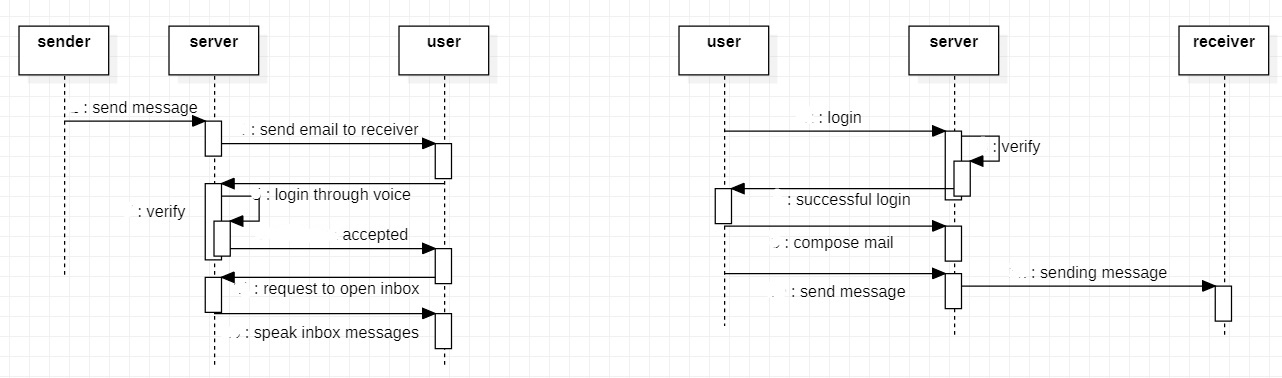


Fig 4.3.1.1 sequence diagram for voice based email system

A sequence diagram in Unified Modelling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams. A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner. If the lifeline is that of an object, it demonstrates a role. Note that leaving the instance name blank can represent anonymous and unnamed instances. In order to display interaction, messages are used. These are horizontal arrows with the message name written above them. Solid arrows with full heads are synchronous calls, solid arrows with stick heads are asynchronous calls and dashed arrows with stick heads are return messages. Activation boxes, or method-call boxes, are opaque rectangles drawn on top of lifelines to represent that processes are being performed in response to the message (Execution Specifications in UML). Objects calling methods on themselves use messages and add new activation boxes on top of any others to indicate a further level of processing. When an object is destroyed (removed from memory), an X is drawn on top of the lifeline, and the dashed line ceases to be drawn below it (this is not the case in the first example though). It should be the result of a message, either from the object itself, or another. A message sent from outside the diagram can be represented by a message originating from a filled-in circle (found message in UML) or from a border of sequence diagram (gate in UML).

**4.3.2 Activity Diagram**

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

Activity diagrams are constructed from a limited repertoire of shapes, connected with arrows.

The most important shape types:

• rounded rectangles represent activities;

• diamonds represent decisions;

• bars represent the start (split) or end (join) of concurrent activities;

• a black circle represents the start (initial state) of the workflow;

• An encircled black circle represents the end (final state).

Arrows run from the start towards the end and represent the order in which activities happen. However, the join and split symbols in activity diagrams only resolve this for simple cases; the meaning of the model is not clear when they are arbitrarily combined with the decisions or loops.

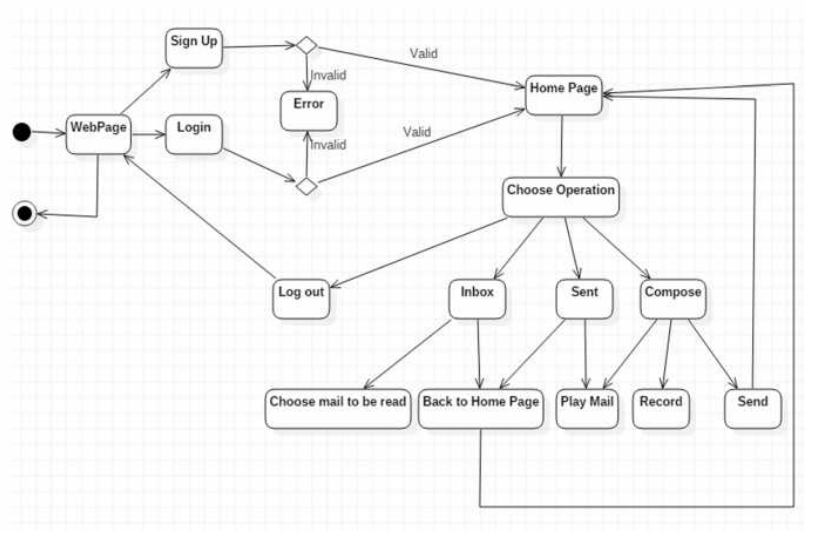


Fig 4.3.2.1 Activity diagram for voice based email system

4.3.3 **Component Diagram**

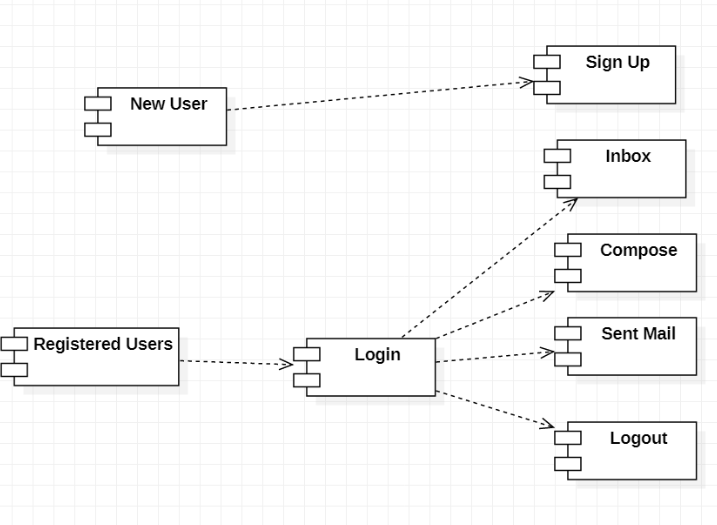
****

Fig 4.3.3.1 component diagram for voice based email system

**Component Level Class Design**

This chapter discusses the portion of the software development process where the design is elaborated and the individual data elements and operations are designed in detail. First, different views of a “component” are introduced. Guidelines for the design of object-oriented and traditional (conventional) program components are presented.

**What is a Component?**

This section defines the term component and discusses the differences between object oriented, traditional, and process related views of component level design. Object Management Group OMG UML defines a component as “… a modular, deployable, and replaceable part of a system that encapsulates implementation and exposes a set of interfaces.”

**An Object Oriented View**

A component contains a set of collaborating classes. Each class within a component has been fully elaborated to include all attributes and operations that are relevant to its implementation. As part of the design elaboration, all interfaces (messages) that enable the classes to communicate and collaborate with other design classes must also be defined. To accomplish this, the designer begins with the analysis model and elaborates analysis classes (for components that relate to the problem domain) and infrastructure classes (or components that provide support services for the problem domain).

**4.3.4 Deployment Diagram:**

Deployment diagrams are used to visualize the topology of the physical components of a system where the software components are deployed. So deployment diagrams are used to describe the static deployment view of a system. Deployment diagrams consist of nodes and their relationships. Purpose: The name Deployment itself describes the purpose of the diagram. Deployment diagrams are used for describing the hardware components where software components are deployed. Component diagrams and deployment diagrams are closely related. Component diagrams are used to describe the components and deployment diagrams shows how they are deployed in hardware. UML is mainly designed to focus on software artifacts of a system. But these two diagrams are special diagrams used to focus on software components and hardware components.

So most of the UML diagrams are used to handle logical components but deployment diagrams are made to focus on hardware topology of a system. Deployment diagrams are used by the system engineers.

The purpose of deployment diagrams can be described as:

• Visualize hardware topology of a system.

• Describe the hardware components used to deploy software components.

• Describe runtime processing nodes.

How to draw Deployment Diagram?

Deployment diagram represents the deployment view of a system. It is related to the component diagram. Because the components are deployed using the deployment diagrams. A deployment diagram consists of nodes. Nodes are nothing but physical hardware’s used to deploy the application. Deployment diagrams are useful for system engineers. An efficient deployment diagram is very important because it controls the following parameters

• Performance

• Scalability

• Maintainability

• Portability

So before drawing a deployment diagram the following artifacts should be identified:

• Nodes

• Relationships among nodes

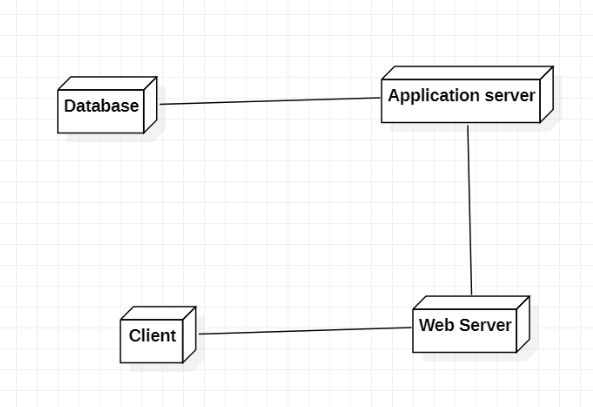
****

Fig 4.3.4.1 Deployment diagram for voice based email System

**CHAPTER-5**

**SOFTWARE ENVIRONMENT**

**5.SOFTWARE ENVIRONMENT**

**5.1 INTRODUCTION TO JAVASCRIPT**

**History of JavaScript**

JavaScript is everywhere, and for the seventh year in a row, it has been ranked [the most commonly used programming language](https://fossbytes.com/javascript-most-popular-programming-language-stack-overflow/), with 67.8% of developers employing it in 2019. Its ascent to the world’s most [popular programming language](https://www.springboard.com/blog/best-programming-language-for-ai/) is synonymous with the rise of the internet itself.

Created out of necessity, it is used to build 95.2% (1.52 billion) of websites today, including some of the world’s largest, like Facebook and YouTube. Without it, we would not have popular and useful web apps such as Google Maps and eBay.

So, without further ado, let’s take a look at what JavaScript is,  how and why it was created, and what’s next for the language.

## What Is JavaScript?

JavaScript is a scripting language that is one of the three core languages used to develop websites. Whereas HTML and CSS give a website structure and style, JavaScript lets you add functionality and behaviours to your website, allowing your website’s visitors to interact with content in many imaginative ways.

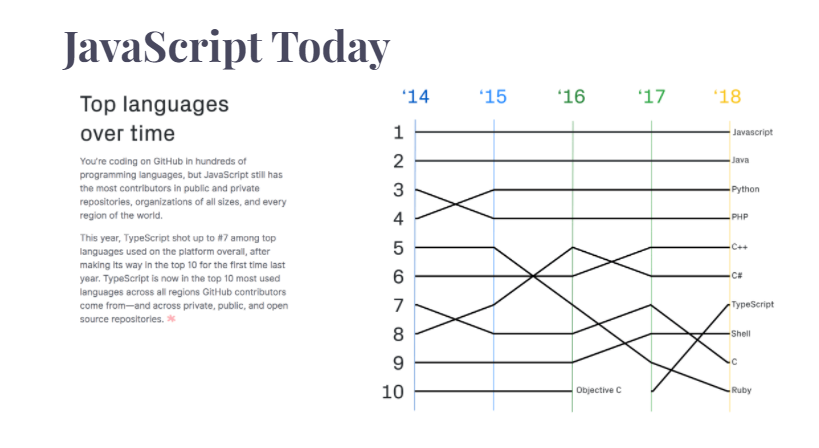
JavaScript is primarily a client-side language, meaning it runs on your computer within your browser. However, more recently the introduction of Node.js has allowed JavaScript to also execute code on servers.

Since its release, JavaScript has surpassed Java, Flash, and [other languages](https://www.springboard.com/blog/best-programming-language-for-ai/) because it is relatively easy to learn, has a free and open community, and, most importantly, is incredibly useful, allowing developers to quickly create apps with audiences in the millions.

## JavaScript vs. Java

There’s often some confusion about the two, but JavaScript and Java have almost nothing in common. The name JavaScript came from Netscape’s support of Java applets within its browser. Many say it was also a marketing tactic to divert some attention from Java, which was the most buzzed-about language at the time. To run Java programs, the code must be first compiled into an executable form. On the other hand, JavaScript was created to be interpreted at run time, making it much more dynamic (these days the boundary of the two methods is a lot more blurred).

JavaScript didn’t exactly get off to the best start. It didn’t perform as well, and those developing in Java considered JavaScript more of a “UI glue” to be used mostly by designers and other non-engineers. But the reality is that having a “glue” language allowed the internet to really flourish. Programmers could react better to use events and compose interactive components. And due to that, JavaScript spread like wildfire and very quickly became the lingua franca of the web.



From its slightly rocky start, JavaScript has risen to be the most [popular programming language](https://www.springboard.com/blog/best-programming-language-for-ai/) in the world. According to GitHub’s 2018 [Octoverse report](https://octoverse.github.com/" \t "_blank), there are more JavaScript code repositories than any other language—and that number is steadily on the rise.

A series of JavaScript frameworks and libraries, such as Ember, Angular, React, and Vue, have been developed to allow powerful and complicated web applications to be written using small teams within short time spans. Alongside client and server software, it is now even possible to write native mobile apps using JavaScript. Unsurprisingly, this is becoming increasingly popular due to the ability to share code between the worlds of mobile and web.

With all this choice, it’s somewhat understandable that there has also been a movement toward a more grassroots, “vanilla” implementation of JavaScript. Web components, small reusable custom browser elements, are the latest challenger aiming to be the next breakthrough in the JavaScript world. Whatever the next big thing is, it’s clear that JavaScript is going to be with us for many years to come.

**Advantages of JavaScript**

The biggest advantages to a JavaScript having a ability to produce the same result on all modern browsers.

**Client-Side execution**: No matter where you host JavaScript, Execute always on client environment to save a bandwidth and make execution process fast.

**User Interface Interactivity**: JavaScript used to fill web page data dynamically such as drop-down list for a Country and State. Base on selected Country, State drop down list dynamically filled. Another one is Form validation, missing/incorrect fields you can alert to a users using alert box.

**Rapid Development**: JavaScript syntax's are easy and flexible for the developers. JavaScript small bit of code you can test easily on Console Panel (inside Developer Tools) at a time browser interpret return output result. In-short easy language to get pick up in development.

**Browser Compatible**: The biggest advantages to a JavaScript having a ability to support all modern browser and produce the same result.

**Make XMLHttpRequest() Object**: [XMLHttpRequest](https://way2tutorial.com/ajax/ajax-xmlhttprequest-object.php" \t "_blank) is special JavaScript object that was designed by Microsoft. XMLHttpRequest object call as a asynchronous HTTP request to the Server for transferring data both side without reloading the page.

**5.2 Annyang Javascript Library For Speech Processing**

* Annyang is a tiny Java script library that lets your visitors control your site with voice commands.
* Annyang supports multiple languages(Arabic , English , French .. … )
* In order to use this library we have to include the tag and use the required functions in our project
* <script src ="[//cdnjs.cloudflare.com/ajax/libs/annyang/2.6.0/annyang.min.js](https://cdnjs.cloudflare.com/ajax/libs/annyang/2.6.0/annyang.min.js)"></script>

Annyang plays nicely with all browsers, progressively enhancing browsers that support Speech Recognition, while leaving users with older browsers unaffected.

**Voice Control for websites using Annyang**

it's increasingly common to control your devices using speech.

Google has speech recognition built into the Chrome browser. Apple and other Android manufacturers have voice control built into their cell phones.

Now, thanks to [annyang!](https://www.talater.com/annyang/" \t "_blank), you can add voice control to your website. annyang! is a tiny javascript library that's really easy to add to your site.

### **Step #1. Load annyang!**

Inside any HTML file, add the following code just before the closing head tag:

<script src ="[//cdnjs.cloudflare.com/ajax/libs/annyang/2.6.0/annyang.min.js](https://cdnjs.cloudflare.com/ajax/libs/annyang/2.6.0/annyang.min.js)"></script>

### **Step #2. Adding Voice Commands**

Now that you've loaded annyang! you can now add your voice commands.

For this example we're going use the voice command 'Hello' to get a response from the website. Add this code immediately after the code you added in Step 1:

<script>

window.onload = function(){

if (annyang) {

var commands = {

'Hello': function() {

alert('Hi! I can hear you.');

}

};

annyang.addCommands(commands);

annyang.start();

}

}

</script>

This is how we can add voice recognition to our website by using Annyang

**5.3 Responsive Voice Javascript Library For Voice Response**

* Responsive Voice is a HTML5-based Text-To-Speech library designed to add voice features to all smartphone, tablet and desktop devices.
* 51 languages supported through 168 voices(languages include UK English, US English, Spanish..)
* In order to use this library we have to include this tag and use the required functions in our project
* <script src=’https://code.responsivevoice.org/responsivevoice.js?key=YOUR\_UNIQUE\_KEY’></script>
* It will supported by all browsers

**Method**

## speak(string text, [string voice], [object parameters])

## responsiveVoice.speak("hello world"); -speaks the given string

**5.4 PHP :Hypertext Preprocessor**

PHP is a popular general-purpose scripting language that is especially suited to web development.

Fast, flexible and pragmatic, PHP powers everything from your blog to the most popular websites in the world.

PHP is a server scripting language, and a powerful tool for making dynamic and interactive Web pages.

PHP is a widely-used, free, and efficient alternative to competitors such as Microsoft's ASP.

What is a PHP File?

* PHP files can contain text, HTML, CSS, JavaScript, and PHP code
* PHP code is executed on the server, and the result is returned to the browser as plain HTML
* PHP files have extension ".php"

What Can PHP Do?

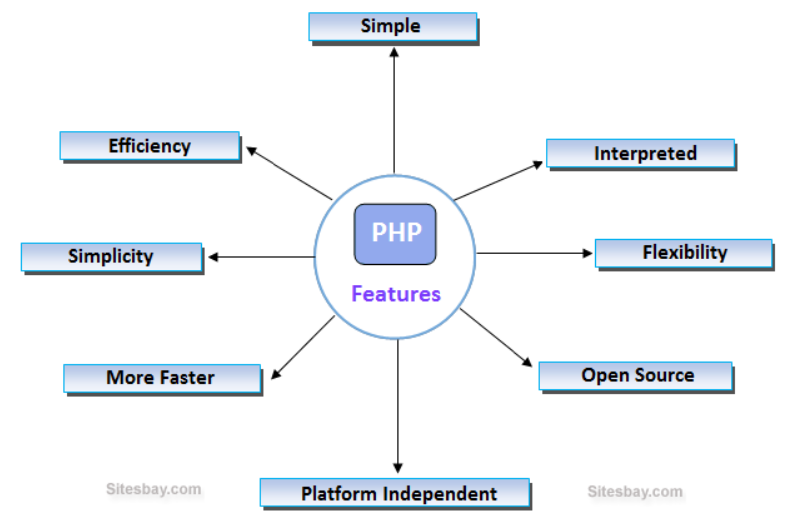
* PHP can generate dynamic page content
* PHP can create, open, read, write, delete, and close files on the server
* PHP can collect form data
* PHP can send and receive cookies
* PHP can add, delete, modify data in your database
* PHP can be used to control user-access
* PHP can encrypt data

With PHP you are not limited to output HTML. You can output images, PDF files, and even Flash movies. You can also output any text, such as XHTML and XML.

## Why PHP?

* PHP runs on various platforms (Windows, Linux, Unix, Mac OS X, etc.)
* PHP is compatible with almost all servers used today (Apache, IIS, etc.)
* PHP supports a wide range of databases
* PHP is free. Download it from the official PHP resource: [www.php.net](http://www.php.net/)
* PHP is easy to learn and runs efficiently on the server side

Features of PHP



**5.5 PHP Mailer**

## What is PHPMailer ?

[PHPMailer](https://github.com/PHPMailer/PHPMailer) is the classic email sending library for PHP. It supports several ways of sending email messages such as mail(), Sendmail, qmail, and direct dispatch to SMTP servers. In addition, it provides a list of advanced features:

* SMTP authentication
* secure/MIME encryption
* support of TLS and SSL protocols
* HTML content along with plain text
* multiple fs, string, and binary attachments
* [embedded images](https://mailtrap.io/blog/embedding-images-in-html-email-have-the-rules-changed/) support

## Sending Email With PHPMailer and SMTP

To send emails with PHPMailer and SMTP, you need to install PHPMailer and configure SMTP settings first.

### How to install PHPMailer

Up to version 5, PHPMailer was providing “PHPMailerAutoload.php” file, so all that was needed was to include it in your script and create a PHPMailer instance. Starting from PHPMailer 6.0 release in August 2017, you need to install it, preferably via [Composer](https://getcomposer.org/), a dependency manager for PHP (this way is recommended by PHPMailer’s creators on Github). After installing Composer, add this line to your composer.json file:

"phpmailer/phpmailer": "~6.1"

or run

composer require phpmailer/phpmailer

If you don’t want to install Composer, for example, while working within a testing environment, you can add PHPMailer manually. Download files with PHPMailer source code, then copy the contents of the PHPMailer folder to one of the include\_path directories specified in your PHP configuration, and load each class file manually:

<?php

use PHPMailer\PHPMailer\PHPMailer;

use PHPMailer\PHPMailer\Exception;

require 'path/to/PHPMailer/src/Exception.php';

require 'path/to/PHPMailer/src/PHPMailer.php';

require 'path/to/PHPMailer/src/SMTP.php';

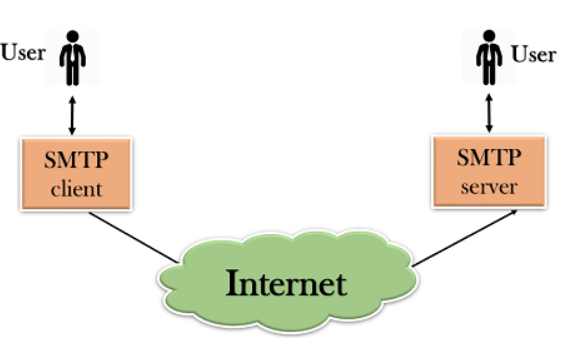
?>

Adding Exception class will help you handle errors and debug them. In PHP it works similarly to the other programming languages. So, without it, if there is an error in your email sending code, you will just see a message saying Exception class is not found, but you won’t be provided with any details on how to debug it. We will describe debugging is a separate section of this post.

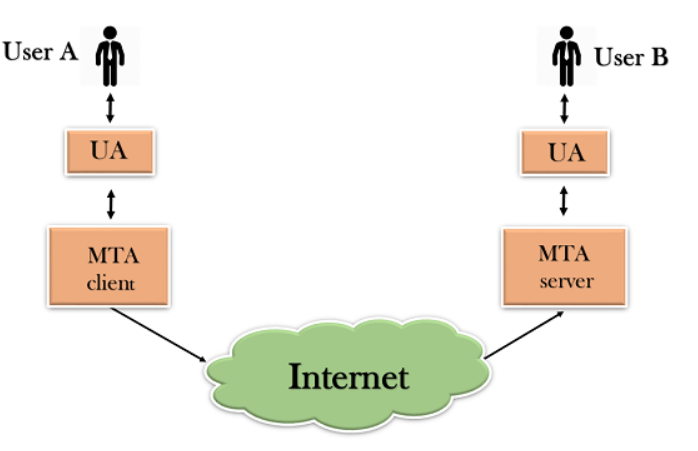
**5.6 SMTP Protocal**

* SMTP stands for Simple Mail Transfer Protocol.
* SMTP is a set of communication guidelines that allow software to transmit an electronic mail over the internet is called **Simple Mail Transfer Protocol**.
* It is a program used for sending messages to other computer users based on e-mail addresses.
* It provides a mail exchange between users on the same or different computers, and it also supports:
  + It can send a single message to one or more recipients.
  + Sending message can include text, voice, video or graphics.
  + It can also send the messages on networks outside the internet.
* The main purpose of SMTP is used to set up communication rules between servers. The servers have a way of identifying themselves and announcing what kind of communication they are trying to perform. They also have a way of handling the errors such as incorrect email address. For example, if the recipient address is wrong, then receiving server reply with an error message of some kind.

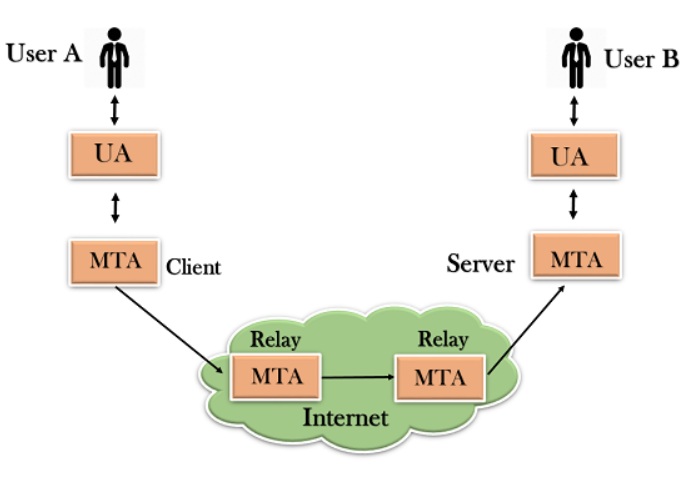
**Components of SMTP**

****

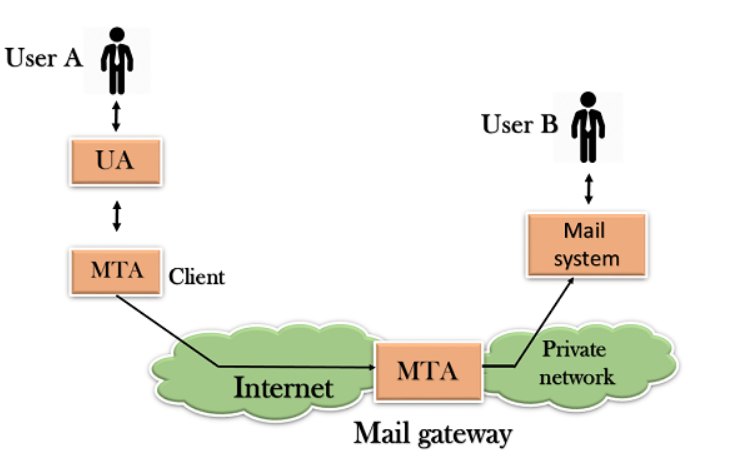
* First, we will break the SMTP client and SMTP server into two components such as user agent (UA) and mail transfer agent (MTA). The user agent (UA) prepares the message, creates the envelope and then puts the message in the envelope. The mail transfer agent (MTA) transfers this mail across the internet.

****

* SMTP allows a more complex system by adding a relaying system. Instead of just having one MTA at sending side and one at receiving side, more MTAs can be added, acting either as a client or server to relay the email.

****

* The relaying system without TCP/IP protocol can also be used to send the emails to users, and this is achieved by the use of the mail gateway. The mail gateway is a relay MTA that can be used to receive an email.



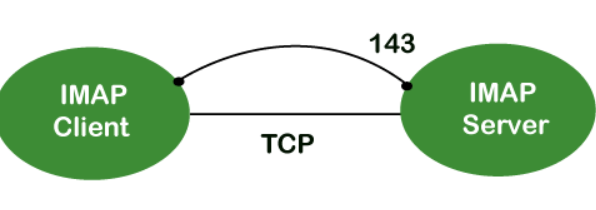
**Working of SMTP**

1. **Composition of Mail:** A user sends an e-mail by composing an electronic mail message using a Mail User Agent (MUA). Mail User Agent is a program which is used to send and receive mail. The message contains two parts: body and header. The body is the main part of the message while the header includes information such as the sender and recipient address. The header also includes descriptive information such as the subject of the message. In this case, the message body is like a letter and header is like an envelope that contains the recipient's address.
2. **Submission of Mail:** After composing an email, the mail client then submits the completed e-mail to the SMTP server by using SMTP on TCP port 25.
3. **Delivery of Mail:** E-mail addresses contain two parts: username of the recipient and domain name. For example, vivek@gmail.com, where "vivek" is the username of the recipient and "gmail.com" is the domain name.  
   If the domain name of the recipient's email address is different from the sender's domain name, then MSA will send the mail to the Mail Transfer Agent (MTA). To relay the email, the MTA will find the target domain. It checks the MX record from Domain Name System to obtain the target domain. The MX record contains the domain name and IP address of the recipient's domain. Once the record is located, MTA connects to the exchange server to relay the message.
4. **Receipt and Processing of Mail:** Once the incoming message is received, the exchange server delivers it to the incoming server (Mail Delivery Agent) which stores the e-mail where it waits for the user to retrieve it.
5. **Access and Retrieval of Mail:** The stored email in MDA can be retrieved by using MUA (Mail User Agent). MUA can be accessed by using login and password.

**5.7 IMAP Protocal**

IMAP stands for **Internet Message Access Protocol**. It is an application layer protocol which is used to receive the emails from the mail server. It is the most commonly used protocols like POP3 for retrieving the emails.

It also follows the client/server model. On one side, we have an IMAP client, which is a process running on a computer. On the other side, we have an IMAP server, which is also a process running on another computer. Both computers are connected through a network.



The IMAP protocol resides on the [TCP/IP](https://www.javatpoint.com/computer-network-tcp-ip-model) [transport layer](https://www.javatpoint.com/computer-network-transport-layer) which means that it implicitly uses the reliability of the protocol. Once the [TCP](https://www.javatpoint.com/tcp) connection is established between the IMAP client and IMAP server, the IMAP server listens to the port 143 by default, but this port number can also be changed.

By default, there are two ports used by IMAP:

* Port 143: It is a non-encrypted IMAP port.
* Port 993: This port is used when IMAP client wants to connect through IMAP securely.

### **Why should we use IMAP instead of POP3 protocol?**

POP3 is becoming the most popular protocol for accessing the TCP/IP mailboxes. It implements the offline mail access model, which means that the mails are retrieved from the mail server on the local machine, and then deleted from the mail server. Nowadays, millions of users use the [POP3 protocol](https://www.javatpoint.com/pop-protocol) to access the incoming mails. Due to the offline mail access model, it cannot be used as much. The online model we would prefer in the ideal world. In the online model, we need to be connected to the internet always. The biggest problem with the offline access using POP3 is that the mails are permanently removed from the server, so multiple computers cannot access the mails. The solution to this problem is to store the mails at the remote server rather than on the local server. The POP3 also faces another issue, i.e., data security and safety. The solution to this problem is to use the disconnected access model, which provides the benefits of both online and offline access. In the disconnected access model, the user can retrieve the mail for local use as in the POP3 protocol, and the user does not need to be connected to the internet continuously. However, the changes made to the mailboxes are synchronized between the client and the server. The mail remains on the server so different applications in the future can access it. When developers recognized these benefits, they made some attempts to implement the disconnected access model. This is implemented by using the POP3 commands that provide the option to leave the mails on the server. This works, but only to a limited extent, for example, keeping track of which messages are new or old become an issue when both are retrieved and left on the server. So, the POP3 lacks some features which are required for the proper disconnected access model.

In the mid-1980s, the development began at Stanford University on a new protocol that would provide a more capable way of accessing the user mailboxes. The result was the development of the interactive mail access protocol, which was later renamed as **Internet Message Access Protocol**.

### **MAP History and Standards**

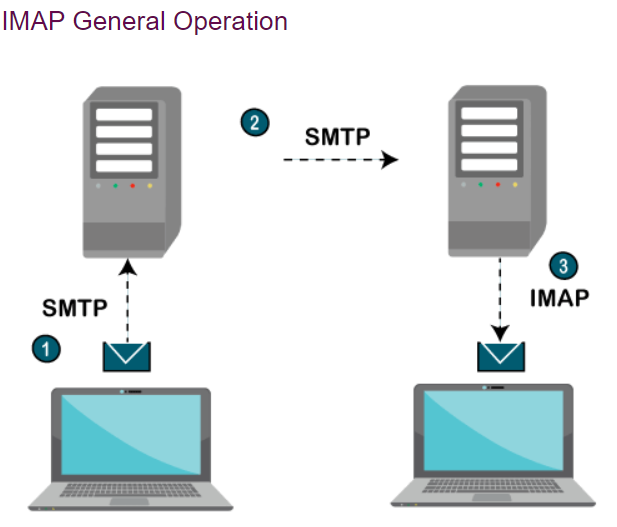
The first version of IMAP was formally documented as an internet standard was IMAP version 2, and in RFC 1064, and was published in July 1988. It was updated in RFC 1176, August 1990, retaining the same version. So they created a new document of version 3 known as IMAP3. In RFC 1203, which was published in February 1991. However, IMAP3 was never accepted by the market place, so people kept using IMAP2. The extension to the protocol was later created called IMAPbis, which added support for Multipurpose Internet Mail Extensions (MIME) to IMAP. This was a very important development due to the usefulness of MIME. Despite this, IMAPbis was never published as an RFC. This may be due to the problems associated with the IMAP3. In December 1994, IMAP version 4, i.e., IMAP4 was published in two RFCs, i.e., RFC 1730 describing the main protocol and RFC 1731 describing the authentication mechanism for IMAP 4. IMAP 4 is the current version of IMAP, which is widely used today. It continues to be refined, and its latest version is actually known as IMAP4rev1 and is defined in RFC 2060. It is most recently updated in RFC 3501.

### **IMAP Features**

IMAP was designed for a specific purpose that provides a more flexible way of how the user accesses the mailbox. It can operate in any of the three modes, i.e., online, offline, and disconnected mode. Out of these, offline and disconnected modes are of interest to most users of the protocol.

The following are the features of an IMAP protocol:

* Access and retrieve mail from remote server: The user can access the mail from the remote server while retaining the mails in the remote server.
* Set message flags: The message flag is set so that the user can keep track of which message he has already seen.
* Manage multiple mailboxes: The user can manage multiple mailboxes and transfer messages from one mailbox to another. The user can organize them into various categories for those who are working on various projects.
* Determine information prior to downloading: It decides whether to retrieve or not before downloading the mail from the mail server.
* Downloads a portion of a message: It allows you to download the portion of a message, such as one body part from the mime-multi part. This can be useful when there are large multimedia files in a short-text element of a message.
* Organize mails on the server: In case of POP3, the user is not allowed to manage the mails on the server. On the other hand, the users can organize the mails on the server according to their requirements like they can create, delete or rename the mailbox on the server.
* Search: Users can search for the contents of the emails.
* Check email-header: Users can also check the email-header prior to downloading.
* Create hierarchy: Users can also create the folders to organize the mails in a hierarchy.



1. The IMAP is a client-server protocol like POP3 and most other TCP/IP application protocols. The IMAP4 protocol functions only when the IMAP4 must reside on the server where the user mailboxes are located. In c the POP3 does not necessarily require the same physical server that provides the SMTP services. Therefore, in the case of the IMAP protocol, the mailbox must be accessible to both SMTP for incoming mails and IMAP for retrieval and modifications.
2. The IMAP uses the Transmission Control Protocol (TCP) for communication to ensure the delivery of data and also received in the order.
3. The IMAP4 listens on a well-known port, i.e., port number 143, for an incoming connection request from the IMAP4 client.

**5.8 XAMPP**

XAMPP is one of the widely used cross-platform web servers, which helps developers to create and test their programs on a local webserver. It was developed by the **Apache Friends**, and its native source code can be revised or modified by the audience. It consists of **Apache HTTP Server, MariaDB, and interpreter** for the different programming languages like PHP and Perl. It is available in 11 languages and supported by different platforms such as the IA-32 package of Windows & x64 package of macOS and Linux.

XAMPP is an abbreviation where X stands for Cross-Platform, A stands for Apache, M stands for [*MYSQL*](https://www.javatpoint.com/mysql-tutorial), and the Ps stand for PHP and Perl, respectively. It is an open-source package of web solutions that includes Apache distribution for many servers and command-line executables along with modules such as Apache server, [MariaDB](https://www.javatpoint.com/mariadb-tutorial), PHP, and Perl.

XAMPP helps a local host or server to test its website and clients via computers and laptops before releasing it to the main server. It is a platform that furnishes a suitable environment to test and verify the working of projects based on Apache, Perl, MySQL database, and PHP through the system of the host itself. Among these technologies, [Perl](https://www.javatpoint.com/perl-tutorial) is a programming language used for web development, [PHP](https://www.javatpoint.com/php-tutorial) is a backend scripting language, and MariaDB is the most vividly used database developed by MySQL. The detailed description of these components is given below.

## Components of XAMPP

As defined earlier, XAMPP is used to symbolize the classification of solutions for different technologies. It provides a base for testing of projects based on different technologies through a personal server. XAMPP is an abbreviated form of each alphabet representing each of its major components. This collection of software contains a web server named **Apache**, a database management system named **MariaDB** and scripting/ programming languages such as **PHP** and **Perl**. X denotes Cross-platform, which means that it can work on different platforms such as [Windows](https://www.javatpoint.com/windows), [Linux](https://www.javatpoint.com/linux-tutorial), and macOS.

**5.9 MySQL Database**

MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation.

MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MySQL is developed, marketed and supported by MySQL AB, which is a Swedish company. MySQL is becoming so popular because of many good reasons −

* MySQL is released under an open-source license. So you have nothing to pay to use it.
* MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.
* MySQL uses a standard form of the well-known SQL data language.
* MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc.
* MySQL works very quickly and works well even with large data sets.
* MySQL is very friendly to PHP, the most appreciated language for web development.
* MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).
* MySQL is customizable. The open-source GPL license allows programmers to modify the MySQL software to fit their own specific environments.
  1. **Hyper Text Markup Language :**
* Hypertext Markup Language (HTML), the languages of the World Wide Web (WWW), allows users to produces Web pages that include text, graphics and pointer to other Web pages (Hyperlinks).
* HTML is not a programming language but it is an application of ISO Standard 8879, SGML (Standard Generalized Markup Language), but specialized to hypertext and adapted to the Web. The idea behind Hypertext is that instead of reading text in rigid linear structure, we can easily jump from one point to another point. We can navigate through the information based on our interest and preference. A markup language is simply a series of elements, each delimited with special characters that define how text or other items enclosed within the elements should be displayed. Hyperlinks are underlined or emphasized works that load to other documents or some portions of the same document.
* HTML can be used to display any type of document on the host computer, which can be geographically at a different location. It is a versatile language and can be used on any platform or desktop.
* HTML provides tags (special codes) to make the document look attractive. HTML tags are not case-sensitive. Using graphics, fonts, different sizes, color, etc., can enhance the presentation of the document. Anything that is not a tag is part of the document itself.
* **Basic HTML Tags:**
* **<!-- -->** Specifies comments
* <A>……….</A>Creates hypertext links
* <B>……….</B>Formats text as bold
* <BIG>……….</BIG>Formats text in large font.
* <BODY>…</BODY>Contains all tags and text in the HTML document <CENTER>...</CENTER>Creates text
* <DD>…</DD>Definition of a term
* <DL>...</DL>Creates definition list
* <FONT>…</FONT>Formats text with a particular font
* <FORM>...</FORM>Encloses a fill-out form
* <FRAME>...</FRAME>Defines a particular frame in a set of frames
* <H#>…</H#>Creates headings of different levels <HEAD>...</HEAD>Contains tags that specify information about a document <HR>...</HR>Creates a horizontal rule
* <HTML>…</HTML>Contains all other HTML tags
* <META>...</META>Provides meta-information about a document <SCRIPT>…</SCRIPT>Contains client-side or server-side script <TABLE>…</TABLE>Creates a table
* <TD>…</TD>Indicates table data in a table
* <TR>…</TR>Designates a table row
* <TH>…</TH>Creates a heading in a table

**CHAPTER-6**

**IMPLEMENTATION**

**6.IMPLEMENTATION**

**6.1 Modules:**

There are five modules that are implementing the Voice based Email System.

They are as follows:

* User Registration
* User Login
* Home
* Compose Mail
* Inbox Mails

**6.2 Module Description**

**User Registration Module:**

This module operation is to add new users to the database so that the users can utilize the services provided by the Voice based Email System . If a new user login to this System he/she must haven an account to access system. This module helps to create an account by taking some important information like email, password, mobile etc. through commands. It will ask some questions to the registering user and the user respond according to it.

Commands used In this module are:

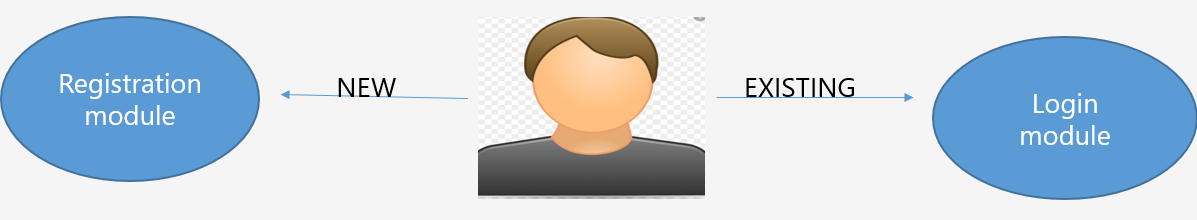
Register user has to fill the details by saying the filed name followed by details

Command submit is used to submit to register to the application.

**User Login Module:**

This module operation is to validate the login user. If he/she is registered user then it allow to home page otherwise it doesn’t allow until he/she register.

It prompts to create an account for new user . command “Yes” to create a new user account and “No” is used to login for user account that have an account in it.



**Home module**

This module operation is to showcase the services provided by the application like compose , inbox , logout .

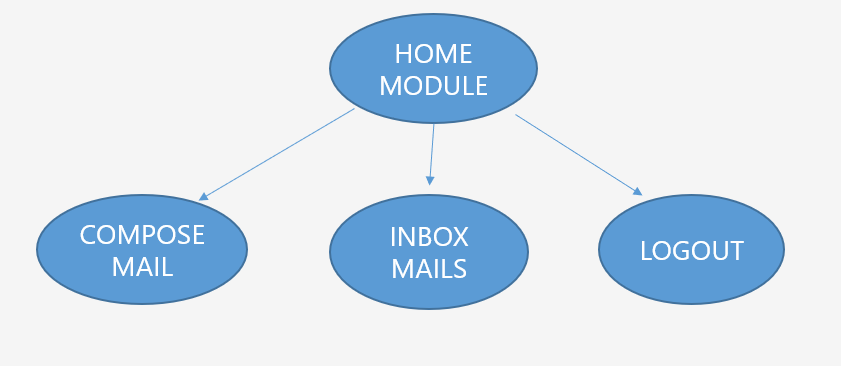
It will prompt the user to say the commands

Inbox : To go to inbox messages

Compose Mail: To write a mail for someone

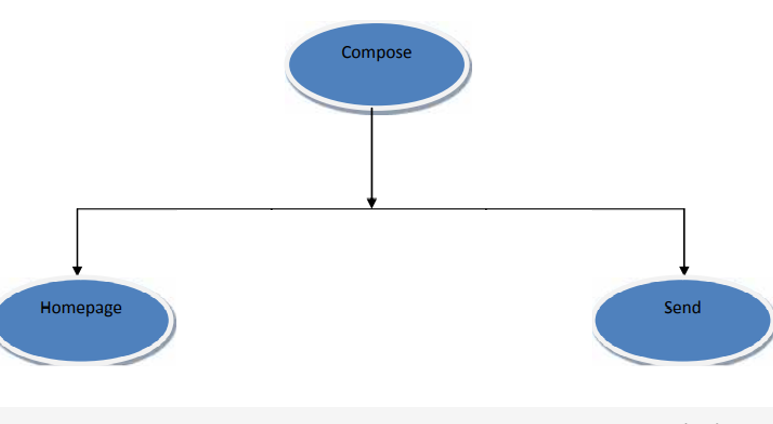
Logout : To logout from account

The user can choose any of the command so that it redirect to that service.



**Compose Module**

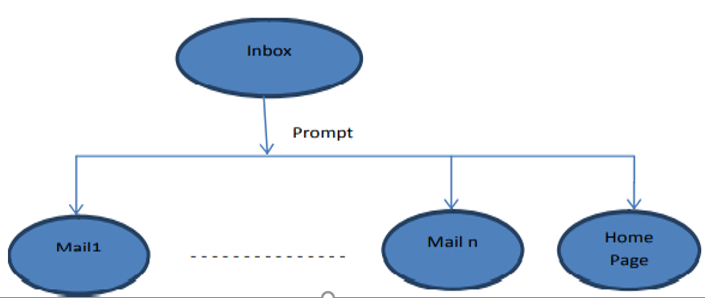
This module operation is used to compose a mail .It prompts require fields And the user respond according to it and finally send command is used to send the composed mail after successful completion it redirect to Home page other wise it remains in the same page and he can go to home page at any moment by saying go to home.



**Inbox Module**

This module operation is to read and display the inbox mails.

Command read Inbox is used to read the mails in the inbox and go to home command to go to home page



**CHAPTER-7**

**TESTING**

**7.1 INTRODUCTION TO TESTING**

The Purpose of Testing is to discover Errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies or a finished product. It is the process of exercising software with the intent of ensuring that the software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of testing. Each test type addresses a specific testing requirement.

**7.2 TESTING STRATEGIES**

**7.2.1 Unit Testing**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results. Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

**Test strategy and approach**

Field testing will be performed manually and functional tests will be written in detail.

**Test objectives**

* All field entries must work properly.
* Pages must be activated from the identified link.
* The entry screen, messages and responses must not be delayed.

Features to be tested

Verify that the entries are of the correct format

No duplicate entries should be allowed

All links should take the user to the correct page.

Features to be tested

Verify that the entries are of the correct format

No duplicate entries should be allowed

All links should take the user to the correct page

**7.2.2 Integration Testing**

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error. Integration tests are designed to test integrated software components to determine if they actually run as one program.

Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

**Functional test:**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centred on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

**7.2.3 System Testing**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

**7.2.4 White Box Testing**

White Box Testing is a testing in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

**7.2.5 Black Box Testing**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot ―see‖ into it. The test provides inputs and responds to outputs without considering how the software works.

**Test Results:**

All the test cases mentioned above passed successfully. No defects encountered.

**7.3 TESTCASE DESIGN**

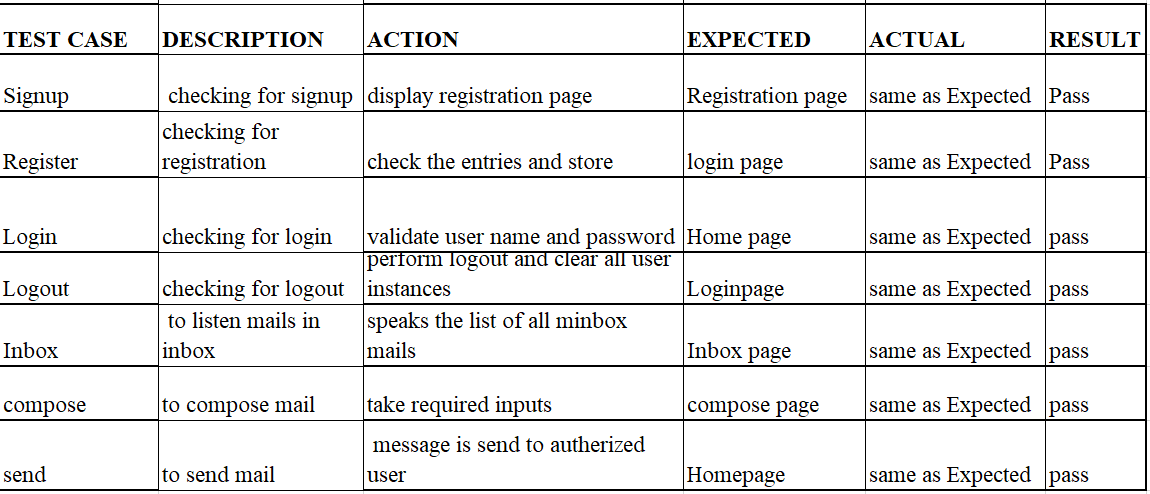
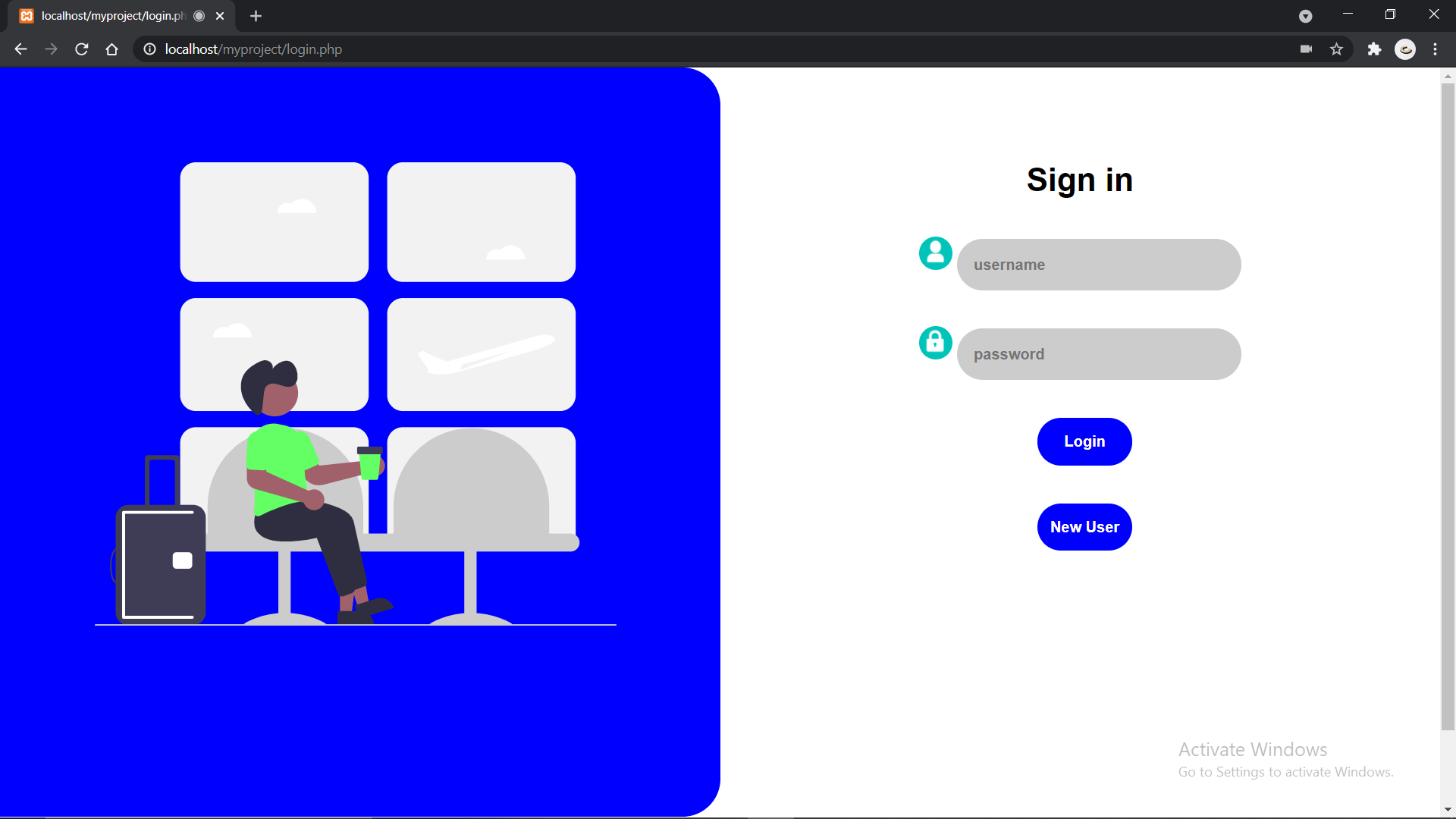
****

Table 7.3.1

**CHAPTER-8**

**RESULTS**

Login page

  
 Fig 8.1 login page

Registration page

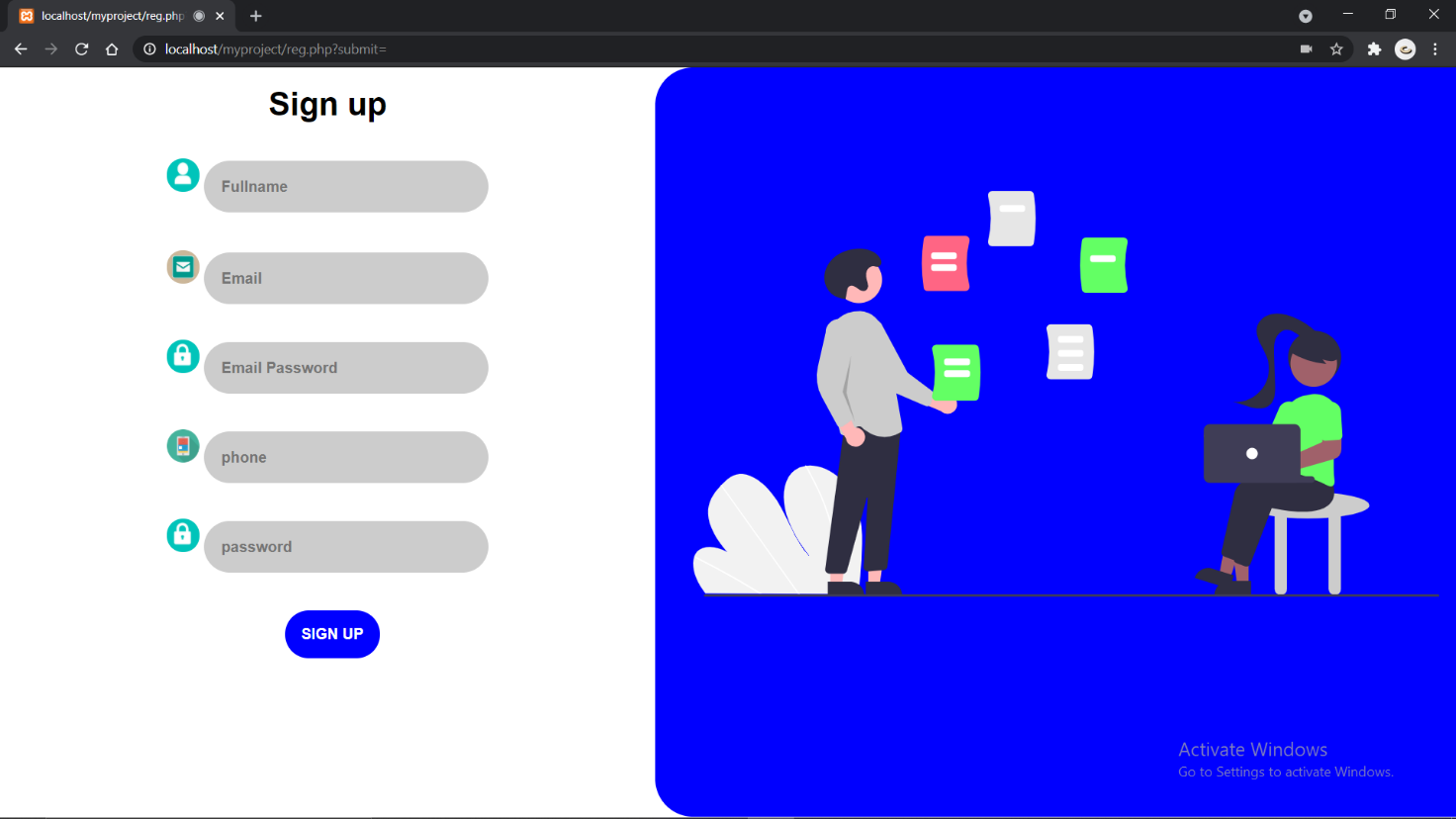


Fig 8.2 Registration page

Home page

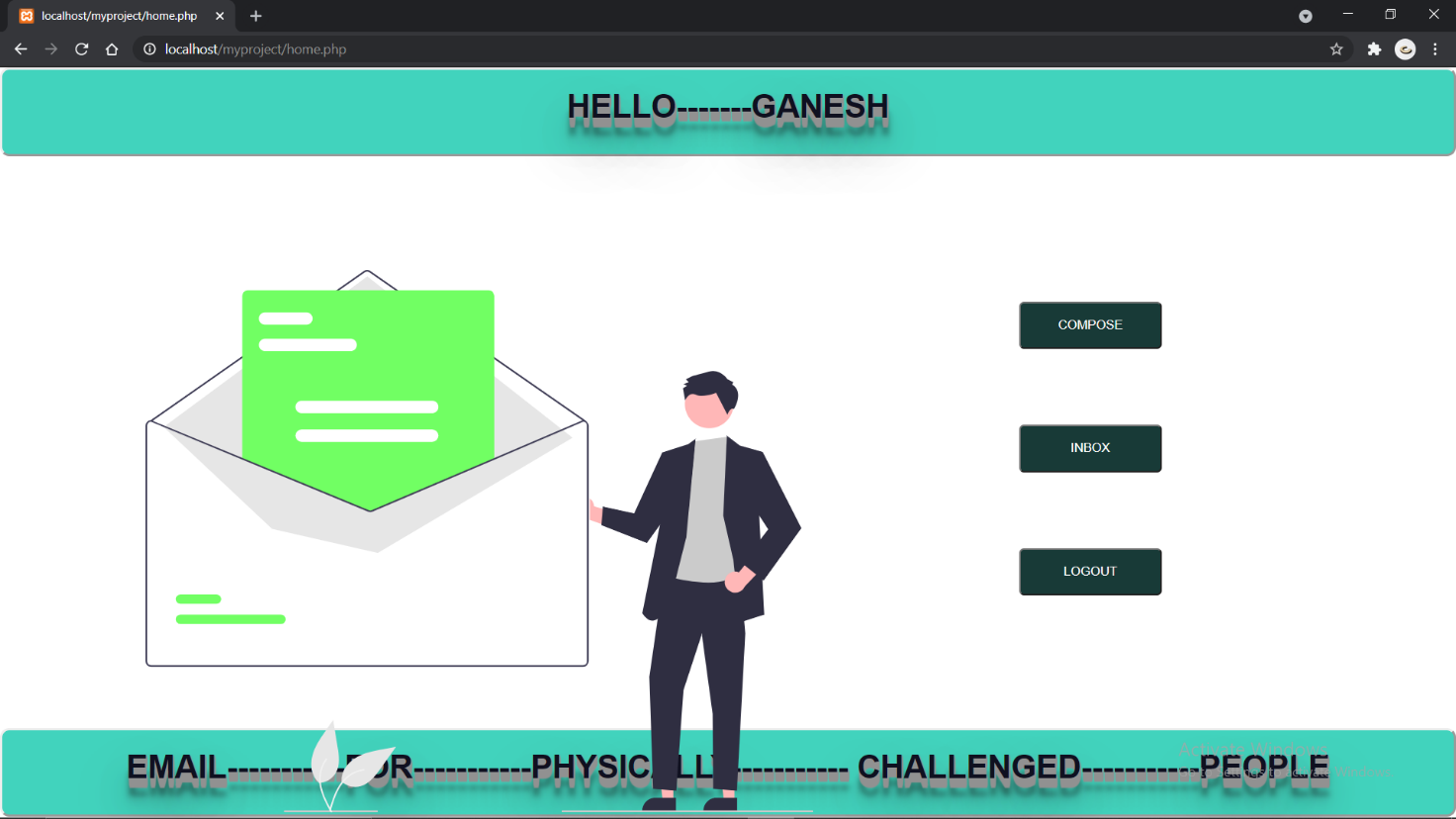


Fig 8.3 Home page

Compose page

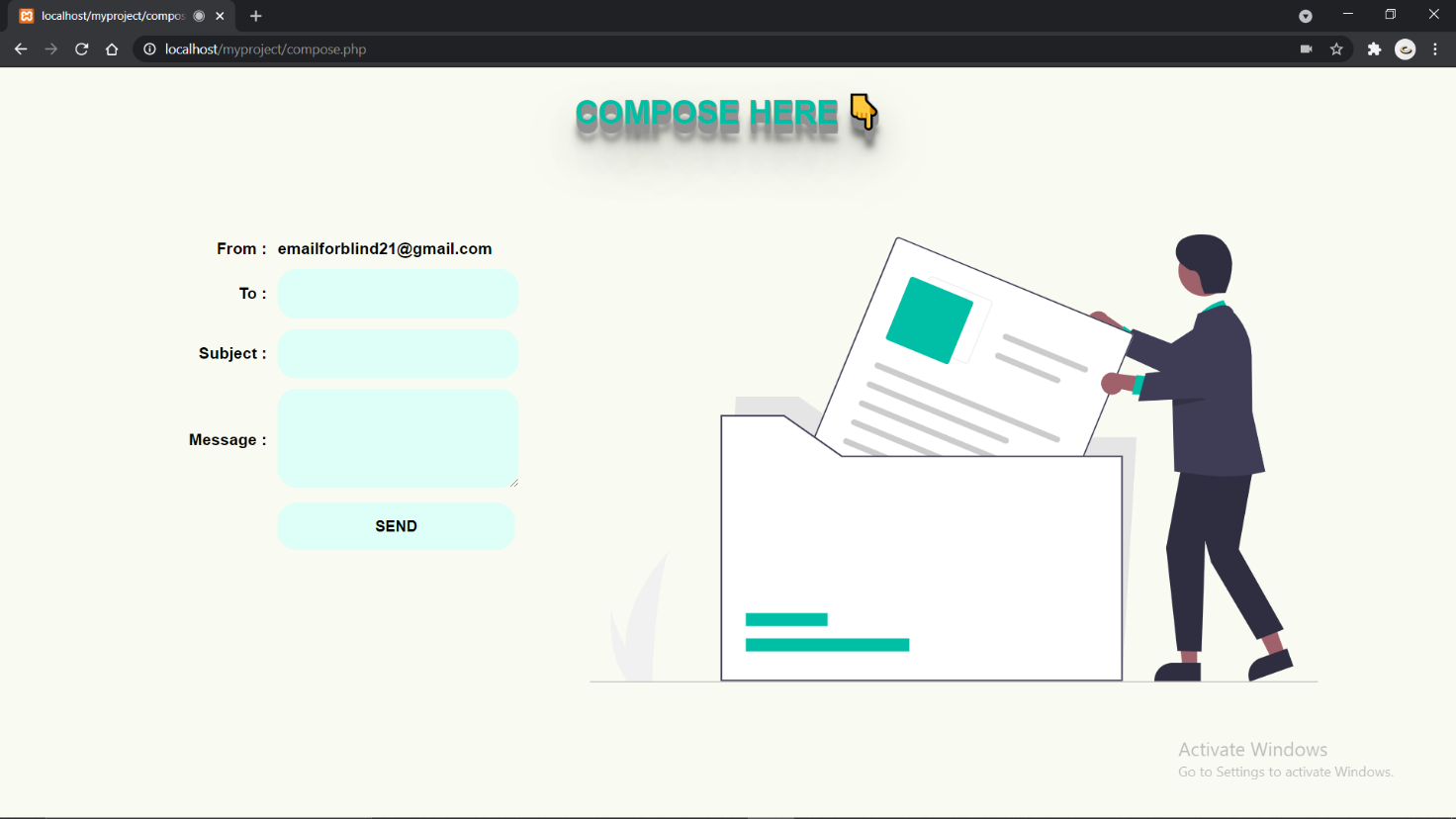


Fig 8.4 Compose page

Inbox page

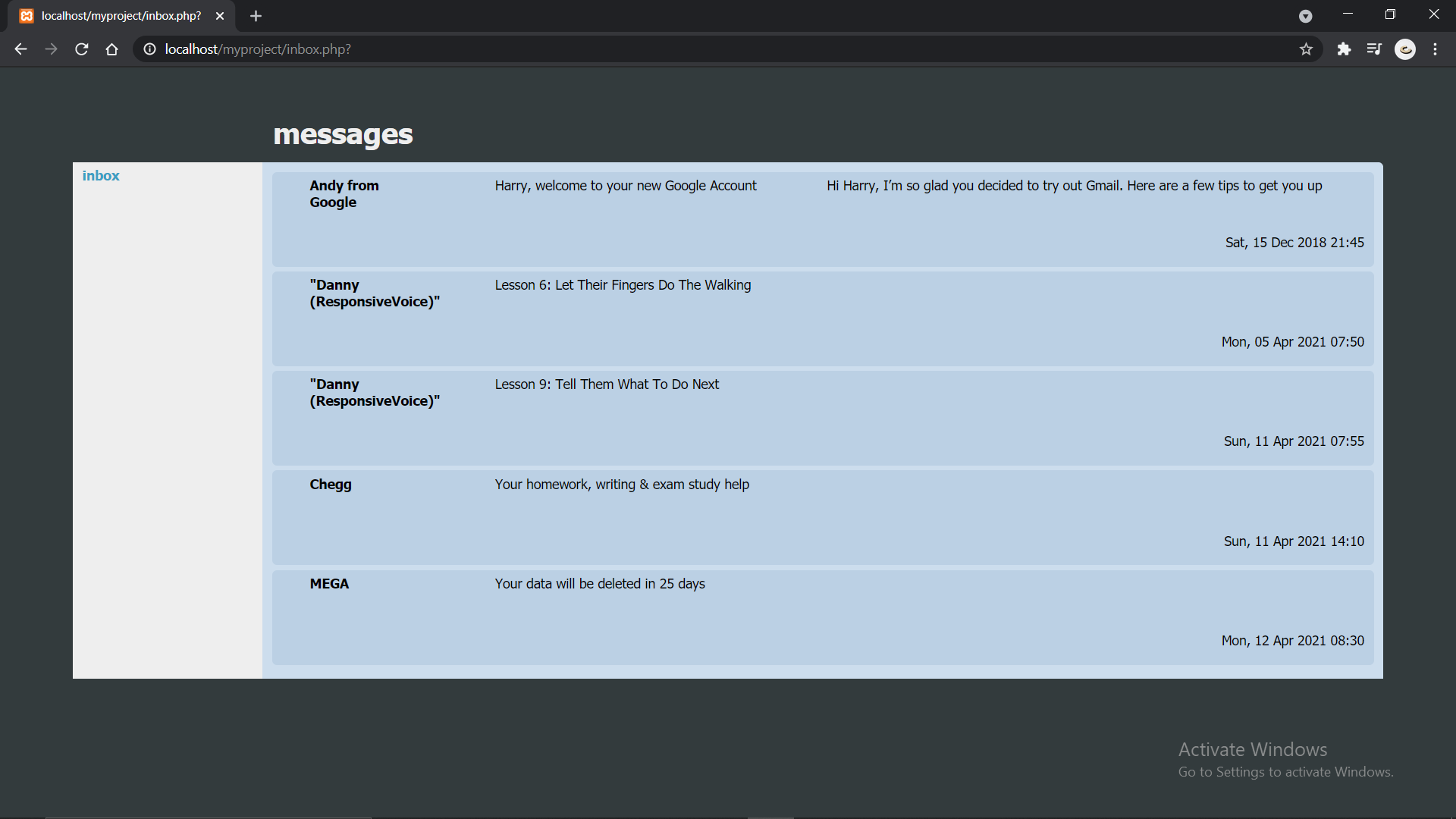


Fig 8.5 Inbox page

**CHAPTER-9**

**CONCLUSION**

**CONCLUSION**

Our project help blinds and handicapped people to access mails easily. It provides a voice-based mailing service where the visually impaired person could read and send mail on their own. It builds confidence and the user gets independent as they do not need help of others. System has eliminated all the concepts and overcome all difficulties that were in traditional methods that were faced by the visually impaired people. Unlike current system, this system focuses more on user friendliness of all types of people including normal people and visually impaired people .

**FUTURE SCOPE**

We will add face authentication in future for strong security and other email related features(like deleting mail , searching mail )

**CHAPTER-10**

**REFERENCES**

**REFERENCES**

[1] Jagtap Nilesh, Pawan Alai, Chavhan Swapnil and BendreM.R.. “Voice Based System in Desktop and Mobile Devices for Blind People”. In International Journal of Emerging Technology and Advanced Engineering (IJETAE), 2014 on Pages 404-407 (Volume 4, issue 2).

[2] UmmuhanysifaU. ,NizarBanu P K , “Voice Based Search Engine and Web page Reader”. In Internationa Journal of Computational Engineering Research (IJCER). Pages 1-5.

[3] G. Shoba, G. Anusha, V. Jeevitha, R. Shanmathi. “AN Interactive Email for Visually Impaired”. In International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE), 2014 on Pages 5089-5092.(Volume 3, Issue 1).

[4] T. Dasgupta and A. Basu. A speech enabled Indian language text to braille transliteration system. In Information and Communication Technologies and Development (ICTD), 2009 International Conference

[5] R. Ghose, T. Dasgupta, and A. Basu. Architecture of a web browser for visually handicapped people. In Students’ Technology Symposium (TechSym), 2010 IEEE, pages 325 –329, april 2010

[6] T. Lauwers, D. Dewey, N. Kalra, T. Stepleton, and M.B. Dias. Iterative design of a braille writing tutor to combat illiteracy. In Information and Communication Technologies and Development, 2007. ICTD 2007. International Conference on, pages 1–8. IEEE,2007

[7] Mamatha, A., Jade, V., Saravana, J., Purshotham, A., & Suhas, A. V. (2020). Voice Based E-mail System for Visually Impaired. International Journal of Research in Engineering, Science and Management, 3(8), 51- 54.

[8] Belekar, A., Sunka, S., Bhawar, N., & Bagade, S. Voice Based E-mail For The Visually Impaired. International Journal of Computer Applications, 975, 8887.

[9] Khan, R., Sharma, P. K., Raj, S., Verma, S. K., & Katiyar, S. Voice Based E-Mail System using Artificial Intelligence. [4 ] Pathan, N., Bhoyar, N., Lakra, U., & Lilhare, D. (2019). V-Mail (Voice Based E-Mail Application).

[10] Sawant, S., Wani, A., Sagar, S., Vanjari, R., & Dhage, M. R. (2018). Speech Based E-mail System for Blind and Illiterate People. International Research Journal of Engineering and Technology (IRJET) e-ISSN, 2395-

<https://responsivevoice.org/>

https://www.talater.com/annyang/