"AURAL"

A project report submitted in partial fulfillment Of requirement for the award of the degree of

BACHELOR OF COMPUTER APPLICATIONS



RANI CHANNAMMA UNIVERSITY, BELAGAVI.

Submitted by:

Mr. Mallikarjun Rayar

Under the guidance of **Prof. Vaishali Kale**



Karnatak Law Society's

GOGTE COLLEGE OF COMMERCE

Tilakwadi, Belgaum-590006

2020 - 2021

Karnatak Law Society's **GOGTE COLLEGE OF COMMERCE**

Tilakwadi, Belgaum. **Bachelor of Computer Applications**



CERTIFICATE

This is to certify that

Mr. Mallikarjun Rayar Reg No. M1810447

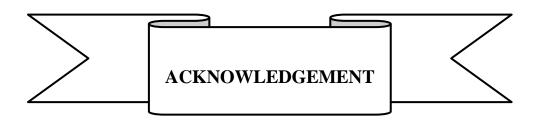
Has satisfactorily completed the project work entitled

"AURAL"

FOR THE FULFILLMENT OF BACHELOR OF COMPUTER APPLICATIONS OF RANI CHANNAMMA UNIVERSITY, BELAGAVI, **FOR THE YEAR 2020-2021**

Guide Prof. Vaishali Kale	Coordinator Prof. V. S. Jalihal	Dr. S. G.Sugur	
EXAMINERS			
1		2	

DEDICATED TO TEACHERS AND FRIENDS



"A helping hand, kind soul these are the ingredients that help to make a success out of any effort."

We take this opportunity to acknowledge the contribution of each individual who has in some way or the other helped me in completing this project successfully. We express our gratitude to our institute, Karnatak Law Society's Gogte College of Commerce (BCA Department), Belagavi and our Director Dr.S.G.Sugur, BCA section for being the source of encouragement.

We also enhance our gratitude to our beloved **Prof.Venugopal Jalihal**, Co-Ordinator BCA Dept for his constant inspiration and necessary resources and working environment in the college.

With pleasure, we use this occasion to concede our heartfelt thanks to **Prof. Vaishali Kale** our internal guide who have guided us with their valuable suggestions & guidance.

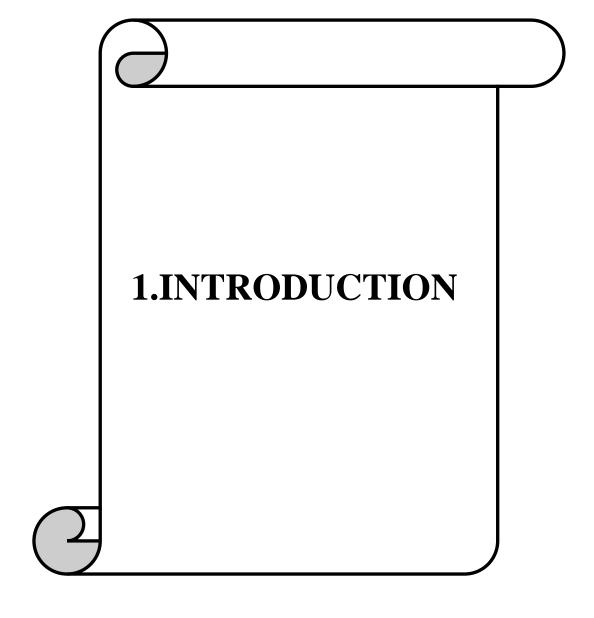
Our expressions extend unbounded to thank our most beloved parents & family members who have always been a moral support & strong pillars at every stage of our life with cheer enthusiasm. We dedicate our work to them. Last but not least, we are thankful to the Almighty for giving us moral support, which helped us during the successful completion of the project.

With heartfelt thanks to One & All, *Mallikarjun Rayar*

INDEX

SL.NO	TOPICS	PAGE NO.		
1.	Introduction 1.1 Introduction 1.2 Objectives of Project 1.3 Scopes of Project	1		
2.	System Analysis and Plannig 2.1 Problem Statement 2.2 System Requirements 2.3 Functional Requirements 2.4 Non-Functional Requirements 2.5 Methadology 2.6 Pert Chart 2.7 Gantt Chart	3		
3	System Design 3.1 System Specification 3.2 Data Flow Diagram 3.3 Use Case Diagram	12		
4.	System Implementation 4.1 System Specification 4.2 Technologies used 4.3 Coding	18		
5.	System Evaluation 5.1 Software Testing 5.2 Test Cases 5.3 Input/Output Scenes	40		
6.	Conclusion 6.1 Conclusion	52		

	6.2 System Requirements	
7.	References	54
	7.1 Sites	
	7.2 Books	
	7.3 Final Application	



1.1 Introduction.

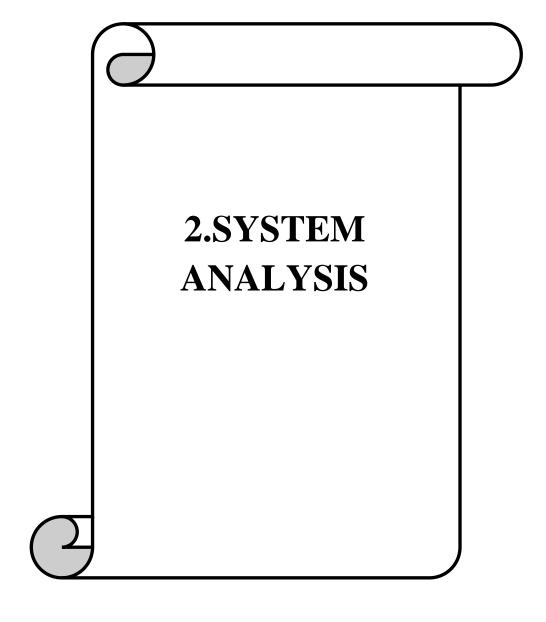
- My project aims at converting PDF file to Audio in a most simplest way, where in each and every user can easily listen to it anywhere anytime.
- The purpose of this project is converting the PDF file to listenable voice. It is made up of two applications, which runs on the user's Pc and server application, which runs on any Pc on the network.
- It's a major advantage is that it plays a very important role for visually impaired users who can simply just hear to it. One such great example is of a girl student who could successfully crack the 'IAS Examination' just by using this technology.

1.2 Objectives

- The main objective of Aural application is to convert the PDF files to listenable audio format.
- It speaks out all the information which is inside a PDF file.

1.3 Scope of the Project

- Instant Conversion of PDF to audio.
- Authentication of the user.
- Real Time feel to the users.
- Small in size.
- Easy to use.



2.1 Problem Statement

This project is to create an application with a server and users to enable the user to convert the PDF file and listen on the go.To develop an instant converting solution to enable users to seamlessly listen and understand the PDF file information. The project should very easy to use enabling even a novice person to use it.

2.2 System Requirements

Hardware Requirements

• System: i5 Processor and above.

• Hard Disk: 500 GB and above.

• RAM: 8 GB and above.

Software Requirements

• Operating system: Windows

• Browser: Chrome

• Coding language: Android SDK, JavaScript

• IDE: Android Studio.

Code-Editor: Visual Studio CodeBackend Service: Firebase.Github.

Runtime Environment

• Device : Smartphone / Tablet.

• OS: Android.

• Minimum Version : 4.4 Android Kitkat.

2.3 Functional Requirements

• User Sign-up/Sign-in:

User must be able to Signup and Signin for the application through a email id.At initial phase the user need to Signup,if the user already have an account then the user can proced with the Signin.

• Uploading PDF file:

The application should require PDF file to convert that file to audio format.

• Selecting Page:

User should mention page number of the PDF file which he would like to listen to.So that it is convenient to the user to listen particular information of the file.

Set Speed Rate:

User should be able to set the audio speed rate at which user want to listen.

• Listen / Start:

Now user can be able to start listening to the PDF file, by clicking on the Listen/Start button, and application start's reading the the information inside the PDF file.

• Stop:

User must be able to stop listening to the PDF file information by clicking on the Stop button.

• Sign-out:

The user can Signout from his account.

2.4 Non-functional Requirements

Scalability

App should be able to provide instant conversion service.

Privacy

Users account should be encrypted to maintain privacy.

Robustness

In case user's device crashes, a backup of their account details must be stored on remote database servers to enable recoverability.

• Performance

Application must be lightweight and must convert PDF files to audio instantly.

2.5 Methodology Adopted

Spiral model:

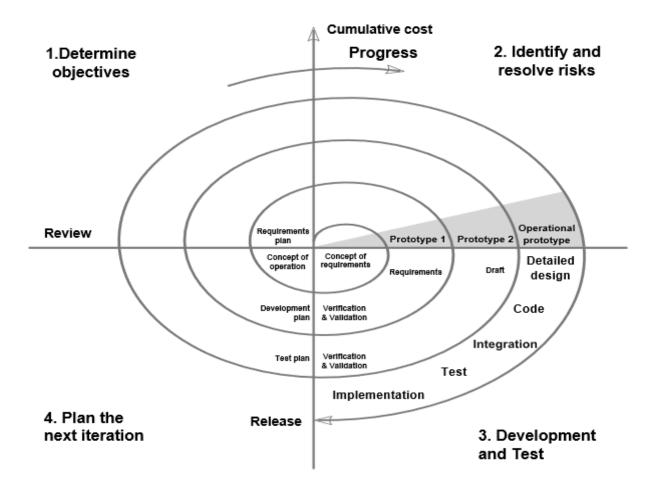


Figure 2.1

Spiral model is one of the most important Software Development Life Cycle models, which provides support for **Risk Handling**. In its diagrammatic representation, it looks like a spiral with many loops. The exact number of loops of the spiral is unknown and can vary from project to project. **Each loop of the spiral is called a Phase of the software development process.** The exact number of phases needed to develop the product can be varied by the project manager depending upon the project risks.

As the project manager dynamically determines the number of phases, so the project manager has an important role to develop a product using spiral model. The Radius of the spiral at any point represents the expenses(cost) of the project so far, and the angular dimension represents the progress made so far in the current phase.

Each phase of Spiral Model is divided into four quadrants as shown in the above figure. The functions of these four quadrants are discussed below-

1. Objectives determination and identify alternative solutions

Requirements are gathered from the customers and the objectives are identified, elaborated and analyzed at the start of every phase. Then alternative solutions possible for the phase are proposed in this quadrant.

2. Identify and resolve Risks

During the second quadrant all the possible solutions are evaluated to select the best possible solution. Then the risks associated with that solution is identified and the risks are resolved using the best possible strategy. At the end of this quadrant, Prototype is built for the best possible solution.

3. Development and test

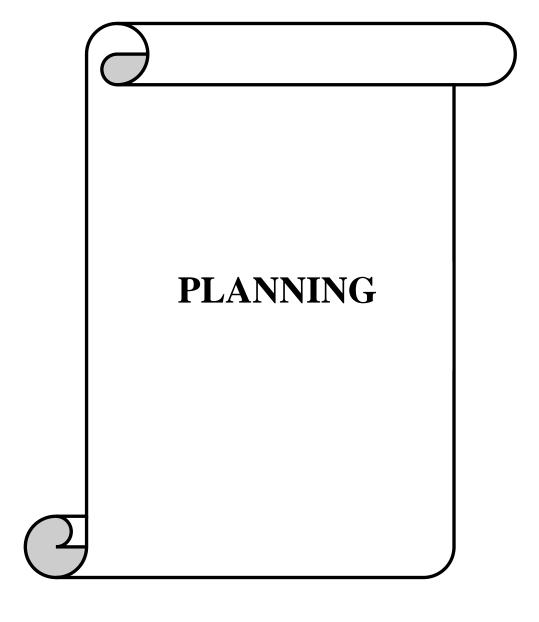
During the third quadrant, the identified features are developed and verified through testing. At the end of the third quadrant, the next version of the software is available.

4. Review and plan for the next Phase

In the fourth quadrant, the Customers evaluate the so far developed version of the software. In the end, planning for the next phase is started.

Risk Handling in Spiral Model

A risk is any adverse situation that might affect the successful completion of a software project. The most important feature of the spiral model is handling these unknown risks after the project has started. Such risk resolutions are easier done by developing a prototype. The spiral model supports coping up with risks by providing the scope to build a prototype at every phase of the software development.

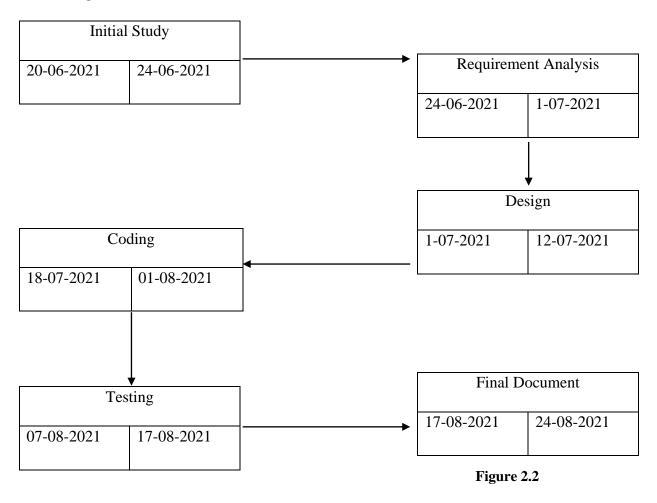


2.6 PERT CHART

A PERT chart is a graphic representation of a project's schedule, showing the sequence of tasks, which tasks can be performed simultaneously, and the critical path of tasks that must be completed on time in order for the project to meet its completion deadline. A PERT chart can document an entire project or a key phase of a project.

The chart allows a team to avoid unrealistic timetables and schedule expectations, to help identify and shorten tasks that are bottlenecks, and to focus attention on most critical tasks. Because it is primarily a project-management tools, a PERT chart is most useful for planning and tracking entire projects or for scheduling and tracking the implementation phase of a planning or improvement effort.

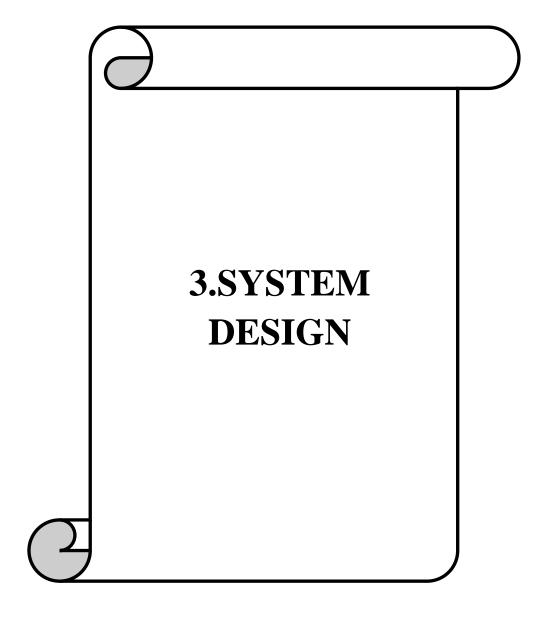
PERT CHART



2.7 GANTT CHART

		Weeks													
	Tasks	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	Initial Study				1	•	•	•	•	•	1	•	1	1	•
2	Requirement Analysis														
3	Design														
4	Coding														
5	Testing and Implementation														
6	Final Document														

Figure 2.3



Aural

3.1 System Specification

Hardware requirements:

Hard Disk: 500 GB and above.

• RAM: 8 GB and above.

• System: i5 Processor and above.

3.2 Data Flow Diagrams

Data Flow Diagrams (DFD)

A Data Flow Diagram is graphical aid for defining systems input, process and outputs. It represents increasing information flow and function details.

A DFD shows the flow of data through a system. A system may be an organization, a manual procedure, software system, mechanical system, a hardware system or any combinations of these. A DFD shows the movement of data through different processes in the system DFD's are made of a number of symbols, which represents system components like process, data flow and external entities.

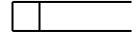
a) Process

Process show that systems do each process can have one or more inputs or outputs. A process in DFD is represented as hollow cylinder. And each process has unique name and number.



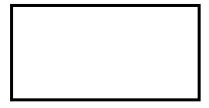
b)Data Store

A file or data store is repository of data. Each data store is represented by an open sided rectangle and each store and has unique name. The symbol of data store is.



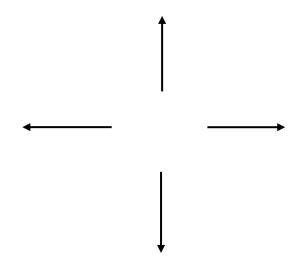
c) External Entities

External entities are outside the system but they either supply input into system or use the system output. They may be an organization's customer or others which system interacts. External entities, which supply data to the system, are sometimes called as 'sinks'. These are represented by a rectangle in the DFD



d) Data Flow

Data flow indicates the passage of data in the system, from where the data flows. It is indicated by an arrow which indicates the direction of flow. The arrow is labeled by the name of the data flow.



Flow of data in the system could be any one of the following:

- From a data store to a process.
- From source to process.
- From a process to a sink.

Context Level DFD of the proposed system:

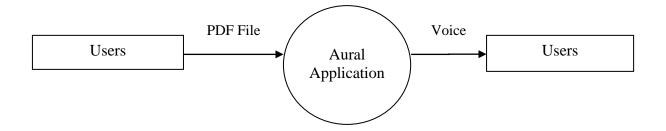


Figure 3.1

Top Level DFD:

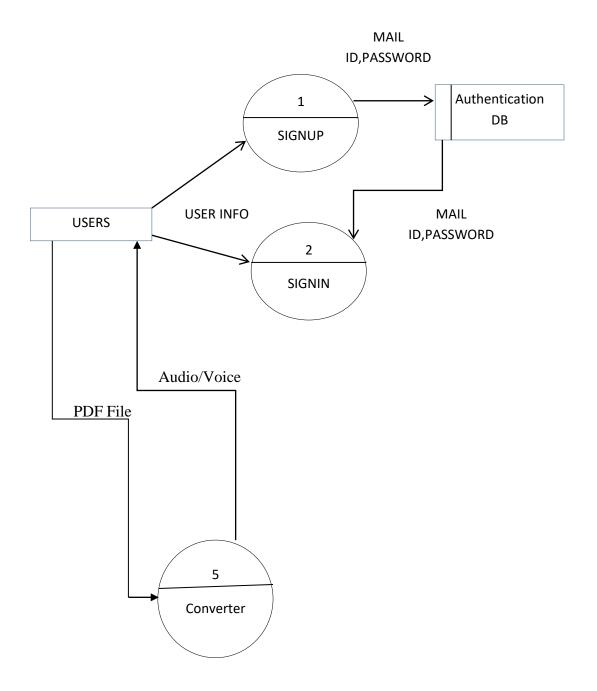


Figure 3.2

3.3 Use Case Diagram

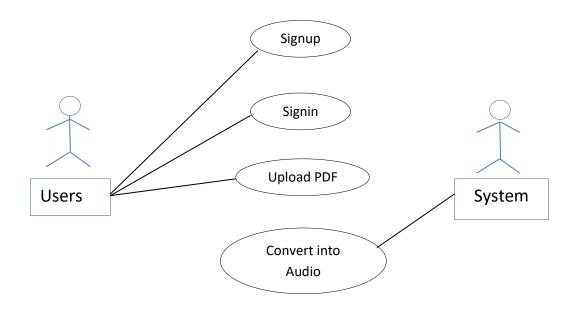
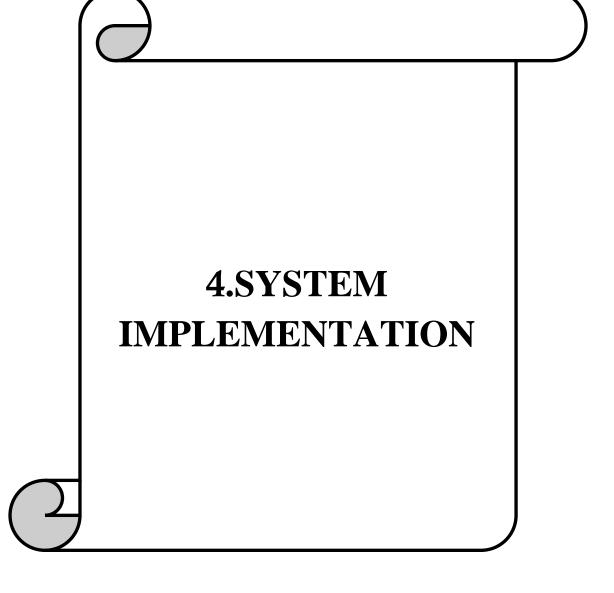


Figure 3.3



4.1 System Specification

Hardware Requirements

- System: i5 Processor and above.
- Hard Disk: 500 GB and above.
- RAM: 8 GB and above.

Software Requirements:

- Operating system: Windows
- Coding language: Android SDK, JavaScript
- IDE:Android Studio.
- Code-Editor: Visual Studio Code
- Backend Service: Firebase, Github.

Runtime Environment:

- Device : Smartphone / Tablet.
- OS: Android.
- Minimum Version: 4.4 Android Kitkat.

Tools and Languages:.

- IDE
 - 1.. Android Studio
- Front End
 - 1.React Native
 - 2.HTML, CSS, JavaScript
- Middleware
 - 1.JavaScript
 - 2.Android SDK
 - 3. Node is

Backend Services

- 1.Firebase
- 2.GitHub Pages

4.2 TECHNOLOGY USED

1. React Native:-

React Native is a JavaScript framework for writing real, natively rendering mobile applications for iOS and Android. It's based on React, Facebook's JavaScript library for building user interfaces, but instead of targeting the browser, it targets mobile platforms. In other words: web developers can now write mobile applications that look and feel truly "native," all from the comfort of a JavaScript library that we already know and love. Plus, because most of the code you write can be shared between platforms, React Native makes it easy to simultaneously develop for both Android and iOS —

- React has wider audience You don't have to decide which audience to target,
 i.e. iOS or Android users, as cross platform software runs on both, which gives you access to wider user base.
- **Platform consistency** There are some navigation and design differences between iOS and Android, which in cross-platform development are dealt with by default, thanks to the shared codebase, This helps with creating a consistent app brand identity on both platform with less effort than if build on native.

Similar to React for the Web, React Native applications are written using a mixture of JavaScript and XML-esque markup, known as JSX. Then, under the hood, the React Native

"bridge" invokes the native rendering APIs in Objective-C (for iOS) or Java (for Android). Thus, your application will render using real mobile UI components, *not* webviews, and will look and feel like any other mobile application. React Native also exposes JavaScript interfaces for platform APIs, so your React Native apps can access platform features like the phone camera, or the user's location.

2. Java:-

Java is a general-purpose computer **programming language** that is <u>concurrent</u>, class-based, <u>object-oriented</u>, and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation.

For example, you can write and compile a Java program on UNIX and run it on Microsoft Windows, Macintosh, or UNIX machine without any modifications to the source code. WORA is achieved by compiling a Java program into an intermediate language called **byte code**. The format of byte code is *platform-independent*. A virtual machine, called the <u>Java Virtual Machine (JVM)</u>, is used to run the byte code on each platform.

3. Javascript:-

JavaScript, often abbreviated as **JS**, is a programming language that conforms to the ECMAScript specification. JavaScript is high-level, often just-in-time compiled, and multi-paradigm. It has curly-bracket syntax, dynamic typing, prototype-based object-orientation, and first-class functions.

Alongside HTML and CSS, JavaScript is one of the core technologies of the World Wide Web, Over 97% of websites use it client-side for web page behavior, often incorporating

third-party libraries. Most web browsers have a dedicated JavaScript engine to execute the code on the user's device.

As a multi-paradigm language, JavaScript supports event-driven, functional, and imperative programming styles. It has application programming interfaces (APIs) for working with text, dates, regular expressions, standard data structures, and the Document Object Model (DOM).

4. Node js:-

Node.js is an open-source, cross-platform, JavaScript runtime environment that executes JavaScript code outside of a web browser. Node.js lets developers use JavaScript to write command line tools and for server-side scripting—running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser. Consequently, Node.js represents a "JavaScript everywhere" paradigm, unifying web-application development around a single programming language, rather than different languages for server- and client-side scripts.

Though .js is the standard filename extension for JavaScript code, the name "Node.js" doesn't refer to a particular file in this context and is merely the name of the product. Node.js has an event-driven architecture capable of asynchronous I/O. These design choices aim to optimize throughput and scalability in web applications with many input/output operations, as well as for real-time Web applications (e.g., real-time communication programs and browser games). The Node.js distributed development project was previously governed by the Node.js Foundation, and has now merged with the JS Foundation to form the OpenJS Foundation, which is facilitated by the Linux Foundation's Collaborative Projects program.

5. HTML:-

The **HyperText Markup Language**, or **HTML** is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items.

6. Firebase:

Firebase is a Backend-as-a-Service (Baas). It provides developers with a variety of tools and services to help them develop quality apps, grow their user base, and earn profit. It is built on Google's infrastructure. Firebase is categorized as a NoSQL database program, which stores data in JSON-like documents.

In Firebase, a document is a set of key-value pairs defined by a schema. A group of documents makes up a collection.

Key Features

a) Authentication:

It supports authentication using passwords, phone numbers, Google, Facebook, Twitter, and more. The Firebase Authentication (SDK) can be used to manually integrate one or more sign-in methods into an app.

b) Real-time database:

Data is synced across all clients in real-time and remains available even when an app goes offline. It uses JSON tree to save the data.

c) Cloud Firestore:

Cloud Firestore is a cloud-hosted, NoSQL database that your ios, android, and web app can access directly via native SDKs. It uses document/collection model to save the data

d) Hosting:

Firebase Hosting provides fast hosting for a web app; content is cached into content delivery networks worldwide.

e) Test lab:

The application is tested on virtual and physical devices located in Google's data centers.

f) Cloud Messaging:

Firebase Cloud Messaging, formerly known as Google Cloud Messaging, is a crossplatform cloud solution for messages and notification for Android, iOS, and Web application, which currently can be used at no cost.

7. Android SDK:

Every time Google releases a new version of Android, a corresponding SDK is also released. To be able to write programs with the latest features, developers must download and install each version's SDK for the particular phone.

The development platforms that are compatible with SDK include operating systems like Windows (XP or later), Linux (any recent Linux distribution) and Mac OS X (10.4.9 or later). The components of Android SDK can be downloaded separately. Third party addons are also available for download.

Although the SDK can be used to write Android programs in the command prompt, the most common method is by using an integrated development environment (IDE). The

recommended IDE is Eclipse with the Android Development Tools (ADT) plug-in. However, other IDEs, such as NetBeans or IntelliJ, will also work. Most of these IDEs provide a graphical interface enabling developers to perform development tasks faster. Since Android applications are written in Java code, a user should have the Java Development Kit (JDK) installed.

4.3 Code

App.js:

```
import React from 'react';
import {
 StyleSheet,
} from 'react-native';
import Signup from './Screens/Signup';
import Signin from './Screens/Signin';
import Loading from './Screens/Loading';
import { NavigationContainer } from '@react-navigation/native';
import { createNativeStackNavigator } from '@react-navigation/native-stack';
import Home from './Screens/Home';
const Stack = createNativeStackNavigator();
const App= () => {
 return (
      <>
   <NavigationContainer>
    <Stack.Navigator screenOptions={{headerShown: false}}>
       <Stack.Screen name="Loading" component={Loading} />
       <Stack.Screen name="Home" component={Home} />
       <Stack.Screen name="Signin" component={Signin} />
       <Stack.Screen name="Signup" component={Signup} />
```

```
</Stack.Navigator>
   </NavigationContainer>
  </>
 );
};
const styles = StyleSheet.create({
container:{
 flex:1,
 alignItems:"center",
 justifyContent:"center"
}
});
export default App;
Signup.js:
import React, { useState } from 'react';
import {TextInput, Button} from 'react-native-paper';
import {StyleSheet, View, Text, TouchableOpacity, Alert, Linking} from 'react-native'
import firebase from '../config';
const Signup=(props)=>{
  const [email,setEmail]=useState("");
  const [password,setPassword]=useState("");
  const [cpassword,setCPassword]=useState("");
  const signUp=(props)=>{
     console.log(email,password,cpassword)
    if(password===cpassword)
       firebase.auth().createUserWithEmailAndPassword(email,password)\\
       .then(user=>{
          props.navigation.replace("Home")
       })
       .catch(err=>{
```

```
console.log(err)
         if(! err.message.includes("undefined is not an object")){
            Alert.alert(err.message)
       })
    }else{
       Alert.alert("Password not matching")
  }
  const googleSignin=()=>{
    var provider = new firebase.auth.GoogleAuthProvider();
    firebase.auth().signInWithPopup(provider)
    .then(result=>{
       console.log(result)
    })
    .catch(error=>{
       console.log(error)
    })
  }
  return(
    <View style={styles.container}>
       <Text style={{textAlign:'center',fontSize:66,marginBottom:10,fontFamily:"CinzelDec</pre>
orative-Bold",color:"#f5a105",textDecorationLine:"underline"}}>Aural</Text>
    <View style={styles.content}>
       <TextInput
         autoCapitalize="none"
         autoCompleteType="email"
         label="Email"
         mode="outlined"
         value={email}
         onChangeText={text => setEmail(text)}
         />
       <TextInput
         style={{marginTop:15}}
```

```
label="Password"
         mode="outlined"
         secureTextEntry
         value={password}
         onChangeText={text => setPassword(text)}
         />
       <TextInput
         style={{marginTop:15}}
         label="Confirm Password"
         mode="outlined"
         secureTextEntry
         value={cpassword}
         onChangeText={text => setCPassword(text)}
         />
       <Button onPress={()=>signUp()} style={{marginTop:15}} mode="contained">Sign-
Up</Button>
       <TouchableOpacity style={{marginTop:15}}>
         <Text onPress={()=>props.navigation.replace("Signin")} style={{textAlign:"center
",color:"white",fontFamily:"JuliusSansOne-Regular"}}>Already have an account ?</Text>
       </TouchableOpacity>
    </View>
       <Text onPress={()=>Linking.openURL("https://mallikarjunrayar.co.vu")} style={{col
or: "white", font Family: "Libre Barcode 39 Text-
Regular",marginTop:30,fontSize:24}}>Mallikarjun</Text>
    </View>
  )
}
const styles = StyleSheet.create({
  content:{
   width:"90%",
  },
  container:{
    backgroundColor: "rgb(42, 43, 46)",
    flex:1,
    alignItems:"center",
```

```
justifyContent:"center"
   },
   googleButton:{
      maxWidth:"50%",
      alignSelf:"center",
      marginTop:10
   }
  });
export default Signup;
Signin.js:
import React, { useState } from 'react';
import {TextInput, Button} from 'react-native-paper';
import {StyleSheet, View, TouchableOpacity, Text, Alert, Linking} from 'react-native'
import firebase from '../config';
const Signin=(props)=>{
  const [email,setEmail]=useState("")
  const [password,setPassword]=useState("")
  const userSignIn=()=>{
    firebase.auth().signInWithEmailAndPassword(email,password)
     .then(user=>{
       props.navigation.replace("Home")
     })
     .catch(err=>{
       console.log(err)
       Alert.alert(err.message);
     })
  }
  return(
```

```
<View style={styles.container}>
       <Text style={{textAlign:'center',fontSize:66,marginBottom:10,fontFamily:"CinzelDec
orative-Bold",color:"#f5a105",textDecorationLine:"underline"}}>Aural</Text>
    <View style={styles.content}>
       <TextInput
         autoCapitalize="none"
         autoCompleteType="email"
         label="Email"
         mode="outlined"
         value={email}
         onChangeText={text => setEmail(text)}
         />
       <TextInput
         style={{marginTop:15}}
         label="Password"
         mode="outlined"
         value={password}
         secureTextEntry
         onChangeText={text => setPassword(text)}
         />
       <Button onPress={()=>{userSignIn()}} style={{marginTop:15}} mode="contained"
> Sign-In</Button>
       <TouchableOpacity style={{marginTop:15}}>
         <Text onPress={()=>props.navigation.replace("Signup")} style={{textAlign:"center
",color:"white",fontFamily:"JuliusSansOne-
Regular" \}>Don't have account? Create here</Text>
       </TouchableOpacity>
    </View>
```

```
<Text onPress={()=>Linking.openURL("https://mallikarjunrayar.co.vu")} style={{col
or: "white", fontFamily: "LibreBarcode39Text-
Regular",marginTop:30,fontSize:24}}>Mallikarjun</Text>
     </View>
  )
}
const styles = StyleSheet.create({
  content:{
   width:"90%"
  },
  container:{
    backgroundColor: "rgb(42, 43, 46)",
    flex:1,
    alignItems:"center",
    justifyContent:"center"
   }
  });
export default Signin;
Loading.js:
import React, { useEffect } from 'react';
import {
 StyleSheet,
 ActivityIndicator
} from 'react-native';
import firebase from '../config'
const Loading= (props) => {
 useEffect(()=>{
  firebase.auth().onAuthStateChanged(user=>{
   if(user){
    props.navigation.replace("Home");
   }else{
```

```
props.navigation.replace("Signup");
   }
  })
 },[])
 return (
  <>
   <ActivityIndicator style={styles.loading} size='large' color='blue' />
  </>
 );
};
const styles = StyleSheet.create({
  loading:{
     flex:1,
    justifyContent:"center",
    alignItems:'center'
  }
});
export default Loading;
Home.js:
import React from 'react';
import { Alert, Linking, StyleSheet, View } from 'react-native';
import { Button, Text } from 'react-native-paper';
import firebase from '../config';
const Home=(props)=>{
 const redirect=()=>{
    try{
       Linking.openURL("https://manojrayar.github.io/converter")
     }
     catch(e){
```

```
Alert.alert("Something went wrong",e)
     }
  }
  const signOut=()=>{
     firebase.auth().signOut().then(()=>{
       Alert.alert("Signned out")
       props.navigation.replace("Signin")
     })
     .catch(err=>{
       console.log(err)
       Alert.alert(err.message)
     })
  }
  return(
     <View style={styles.container}>
       <Text style={{textAlign:'center',fontSize:66,marginBottom:10,fontFamily:"CinzelDec</pre>
orative-Bold",color:"#f5a105",textDecorationLine:"underline"}}>Aural</Text>
       <Button mode="contained" style={{marginBottom:30}} onPress={()=>{redirect()}}
> <Text style={ fontSize:24,color: "white",fontFamily: "JuliusSansOne-
Regular"}}> Click here to listen PDF's </Text> </Button>
       <Button mode="contained" style={{opacity:1}} color="#b00020" onPress={()=>{sig
nOut()}}>Sign-out</Button>
     </View>
  )
}
const styles = StyleSheet.create({
  container:{
    backgroundColor: "rgb(42, 43, 46)",
    flex:1,
    alignItems:"center",
    justifyContent:"center",
  }
})
```

export default Home;

Package.json:

```
"name": "aural",
"version": "0.0.1",
"private": true,
"scripts": {
 "android": "react-native run-android",
 "ios": "react-native run-ios",
 "start": "react-native start",
 "test": "jest",
 "lint": "eslint ."
},
"dependencies": {
 "@react-navigation/native": "^6.0.2",
 "@react-navigation/native-stack": "^6.0.5",
 "firebase": "^8.9.1",
 "react": "17.0.1",
 "react-google-button": "^0.7.2",
 "react-native": "0.64.2",
 "react-native-paper": "^4.9.2",
 "react-native-safe-area-context": "^3.3.0",
 "react-native-screens": "^3.5.0",
 "react-native-webview": "^11.13.0"
},
"devDependencies": {
 "@babel/core": "^7.12.9",
 "@babel/runtime": "^7.12.5",
 "@react-native-community/eslint-config": "^2.0.0",
 "babel-jest": "^26.6.3",
```

```
"eslint": "7.14.0",
"jest": "^26.6.3",
"metro-react-native-babel-preset": "^0.64.0",
"react-test-renderer": "17.0.1"
},
"jest": {
   "preset": "react-native"
}
```

index.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  k rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/css/mat
erialize.min.css">
  <link rel="stylesheet" href="./css/style.css">
  <title>Document</title>
</head>
<body>
  <div class="container">
      <form action="#">
       <div class="file-field input-field">
        <div class="btn">
         <span>PDF-File</span>
         <input type="file" accept="application/pdf" id="input" />
        </div>
        <div class="file-path-wrapper">
         <input class="file-path validate" type="text">
```

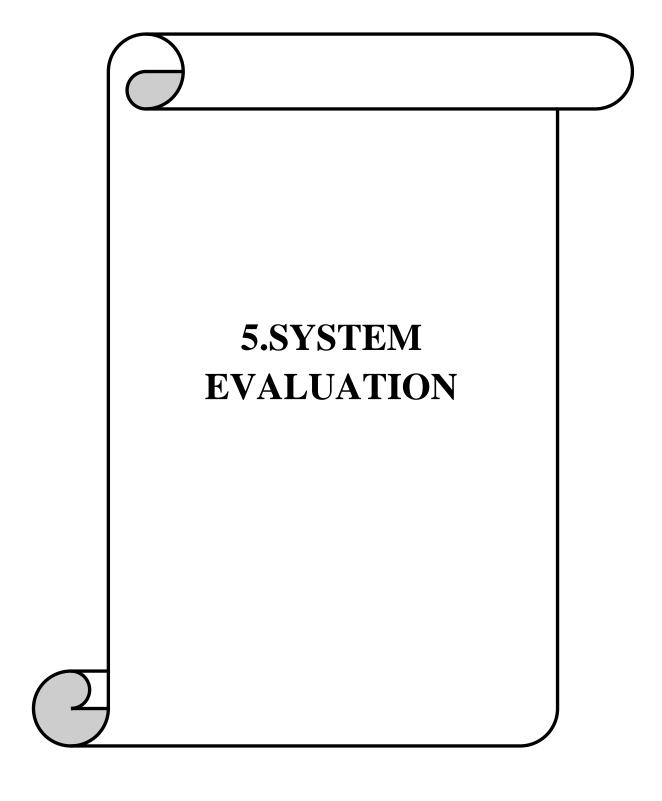
```
</div>
       </div>
      </form>
      <div class="row">
        <div class="input-field col s6">
         <input placeholder="eg: 1" id="pageno" type="number" class="validate" value="1"</pre>
>
         <label for="first_name">Page Number</label>
        </div>
        <div class="col s6">
         <form action="#">
           <label for="rate">Rate</label>
             <input type="range" min="0.1" max="10" value="1" id="rate" step="0.1" />
             <span id="rate-label" class="ms-2">1</span>
           </form>
        </div>
      </div>
      <div class="row">
       <button class="btn waves-effect" id="speak">Listen</button>
       <button class="btn waves-effect" id="cancel">Stop</button>
      </div>
  </div>
<script src="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/js/materialize.min.js">
</script>
  <script src="https://cdn.jsdelivr.net/npm/pdfjs-dist@2.9.359/build/pdf.min.js"></script>
  <script src="./js/app.js"></script>
</body>
</html>
```

converter.js:

```
let speech = new SpeechSynthesisUtterance();
speech.lang = "en";
let voices = [];
window.speechSynthesis.onvoiceschanged = () = > \{
 voices = window.speechSynthesis.getVoices();
 speech.voice = voices[0];
 var elems = document.querySelectorAll('select');
 var instances = M.FormSelect.init(elems);
};
document.querySelector("#rate").addEventListener("input", () => {
  const rate = document.querySelector("#rate").value;
  speech.rate = rate;
  document.querySelector("#rate-label").innerHTML = rate;
 });
var PDF_URL = './resume.pdf';
const inputElement = document.getElementById("input");
inputElement.addEventListener("change", handleFiles, false);
function handleFiles() {
   const fileList = this.files;
   console.log(fileList[0].name)
   PDF_URL=fileList[0].name
   _OBJECT_URL = URL.createObjectURL(fileList[0])
}
function getPageText(pageNum, PDFDocumentInstance) {
return new Promise(function (resolve, reject) {
     PDFDocumentInstance.getPage(pageNum).then(function (pdfPage) {
       pdfPage.getTextContent().then(function (textContent) {
          var textItems = textContent.items;
          var finalString = "";
```

```
for (var i = 0; i < \text{textItems.length}; i++) {
            var item = textItems[i];
            finalString += item.str + " ";
          }
          resolve(finalString);
        });
     });
   });
}
document.getElementById('speak').addEventListener('click', function () {
   var loadingTask = pdfjsLib.getDocument({url:_OBJECT_URL})
   loadingTask.promise.then(PDFDocumentInstance => {
     var totalPages = PDFDocumentInstance.numPages;
     document.getElementById('pageno').setAttribute('max',totalPages)
     var pageNumber =parseInt(document.getElementById('pageno').value);
     console.log( typeof(pageNumber) )
     getPageText(pageNumber, PDFDocumentInstance).then(function (textPage) {
       speech.text = textPage.toLowerCase();
       window.speechSynthesis.speak(speech);
       console.log(textPage.toLowerCase());
     });
   })
})
document.querySelector("#cancel").addEventListener("click", () => {
   console.log("Stoped")
  window.speechSynthesis.cancel();
 });
Style.css:
*{
  color: white;
  }
```

```
body{
   background-color:#2a2b2e;
}
.btn{
   background-color: #661fff8a;
}
```



5.1 Software Testing:

The primary and larger objective of testing is to deliver quality software. Quality software is one that is devoid of error and meets with customer's stated requirements.

If errors are found, then the software must be debugged to locate these errors in the various parts of the program. Corrections are then made. The program/system must be tested once again after corrections have been implemented – this time with additional objective of finding out whether or not corrections in one part of the system have introduced any new errors elsewhere in the system.

Once all errors are found, then another objective must be accomplished that is to check whether or not the system is doing what it is supposed to do. So another aspect of testing is that it must also ensure that the system meets with user requirements.

Here are some of the testing techniques.

- **1.** Unit Testing
- **2.** Integration Testing
- **3.** Module Testing
- **4.** Subsystem Testing
- **5.** Black Box Testing
- **6.** White Box Testing
- **7.** Equivalence Partitioning
- **8.** Ad-hoc Testing
- **9.** Boundary Value Analysis

The Testing Technique used in this project is as follows.

1. Unit Testing

Unit testing is performed by the developers before the setup is handed over to the testing team to formally execute the test cases. Unit testing is performed by the respective developers on the individual units of sources code assigned areas. The developers use test data of the quality assurance team. Unit testing is performed on each of the modules like top trending tweets, retweets, user information etc.

The goal of unit testing is to isolate each part of the program and show that individual parts are correct in terms of requirements and functionality.

2. Integration Testing

The testing of combined parts of an application to determine if they function correctly together is Integration testing. There are two methods of doing Integration Testing bottom-up Integration testing and Top down Integration testing.

Bottom-up

a. This testing begins with unit testing, followed by tests of progressively higher-level combinations of units called modules or builds.

Top-Down

b. This testing, the highest-level modules are tested first and progressively lower-level modules are tested after that.

In a comprehensive software development environment, bottom-up testing is usually done first, followed by top-down testing. The process concludes with multiple tests in the complete application, preferably in scenarios designed to mimic those it will encounter in customer's computers, systems and network. Integration testing is performed on user information module to check whether each of the components work in proper manner when they are combined.

3. Module Testing

Module testing is the testing of complete code objects as produced by the compiler when built from source. It is a collection of independent components such as an object class, an abstract data-type or some loser collection of procedures and functions. A module encapsulates related components so can be tested without other modules. Module testing is done on pie chart, histogram etc.

4. Sub System Testing

This phase of testing involves collecting of modules, which have been integrated into subsystems. Subsystems may be independently designed and implemented. The most common problems, which arise in the large software systems, are sub system interface mismatches. The subsystem test process should therefore concentrate on the detection of interface errors by rigorously exercising these interfaces.

5. System testing

System testing is the testing of behaviour of a complete and fully integrated software product based on the software requirements specification (SRS) document. In main focus of this testing is to evaluate functional end user requirements.

5.2 Test Cases

POSITIVE TESTING OF LOGIN FUNCTIONALITY:

Testing for Login Functionality with valid email and password.

Test Case ID	Test Title	Description	Input Data	TestCase Steps	Expected Result	Actual Result	Status
1	User Signin	Testing Signin functionality of User successful Sigin	Valid Email Valid Password	1.Open the app and enter 2.Type valid email id in email section 3.Enter Valid Password 4.Click on Signin	Home screen	Siggne d into app, Home Screen visible	Test Case Passed

Testing under **Signin** functionality with Valid Email and Incorrect Password.

Test Case ID	Test Title	Description	Input Data	TestCase Steps	Expected Result	Actual Result	Status
2	User	Testing Signin functionality of User for unsuccessful Sigin	Valid Email Invalid Password	1.Open the app 2.Type valid email id in email section 3.Enter Invalid Password 4.Click Signin	Display Error Invalid Password	Display Error	Test Case Passed

NEGATIVE TESTING OF LOGIN FUNCTIONALITY

Testing under **Signin** functionality with Invalid Email and Password.

Test Case ID	Test Title	Description	Input Data	Test Case Steps	Expecte d Result	Actual Result	Status
3	User Signin	Testing Signin functionality of User for unsuccessful signin with blank Email and blank Password	Blank Email Blank Password	1.Open the app 2.Type blank email id in email section	Display Error Enter Valid Id	Display Error	Test Case Passed

POSITIVE TESTING OF REGISTRATION FUNCTIONALITY:

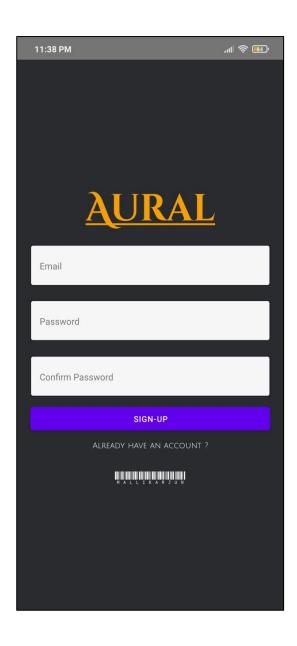
Testing under Registration/Signup functionality with following details in Aural app.

Test Case ID	Test Title	Description	Input Data	TestCase Steps	Expect ed Result	Actual Result	Status
4	User registration /Signup	Testing Register functionality of User for successful register	Valid Email, Usernam, Password	1.Open the app. 2.On the Signin Page click on the "don't have an account create here" 3.Type Email, Password.	Users email Id and verified, Redirec t to Home screen	User registered /Signup inside the app	Test Case Passe d

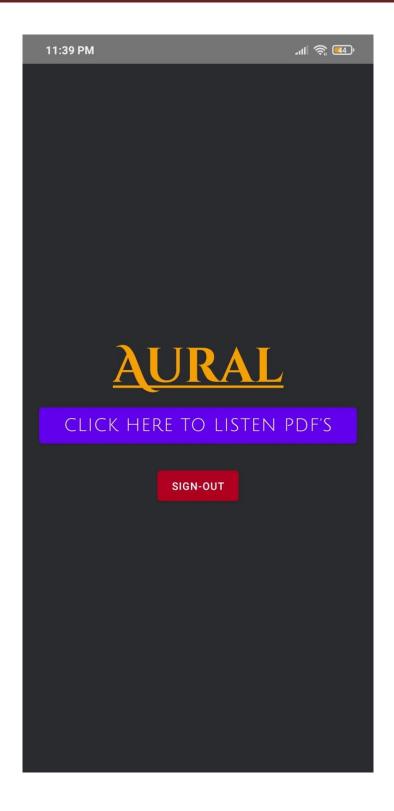
Positive Testing for Aural App

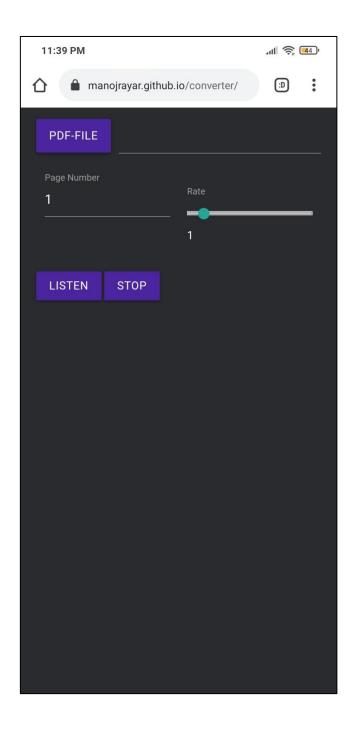
ID	Module	Sub Module	Input Data	Expected Result	Actual Result	Statu s
1.	PDF Module	Selecting File	Valid PDF file	Display selected file name	PDF file name displayed	Pass
2.	Register/Signu p Module	User register/ Signup	Email, Password	After click on "Signup" Redirect to Home	User is registered	Pass
3.	Speed Rate	Audio speed	Select the value in range	Audio speed increased.	Speed increased	Pass
4.	Page Module	Page Number	Enter the page number	Application speak out the selected page.	Speaks the selected page	Pass
5	Listen Module	Start speaking	Select the PDF, set the rate, enter the page no.	Application reads the PDF file	Able to listen the audio	Pass
6	Stop Module	Stop speaking	Click on the Stop button	Audio stops	Application stopped reading	Pass
7	Sign-out Module	Sign-out from App	Click on the Signout button	Signout from the current account and directed at login page	Signout is Successful	Pass

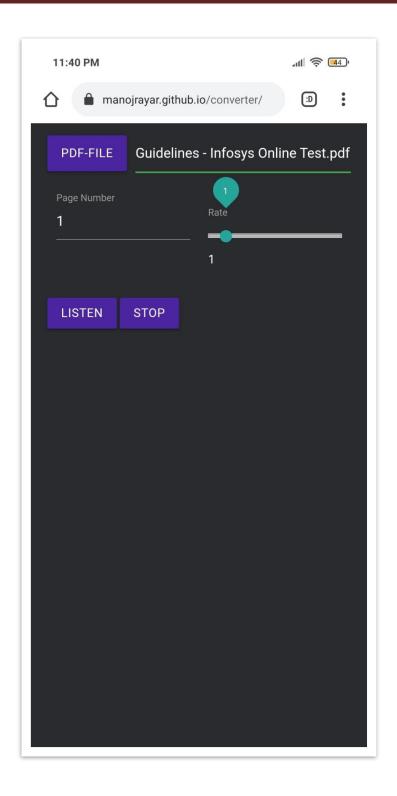
5.3 Input And Output Screens

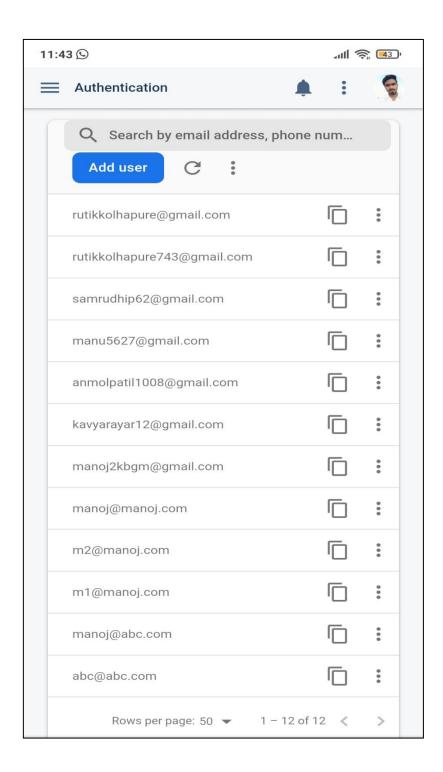


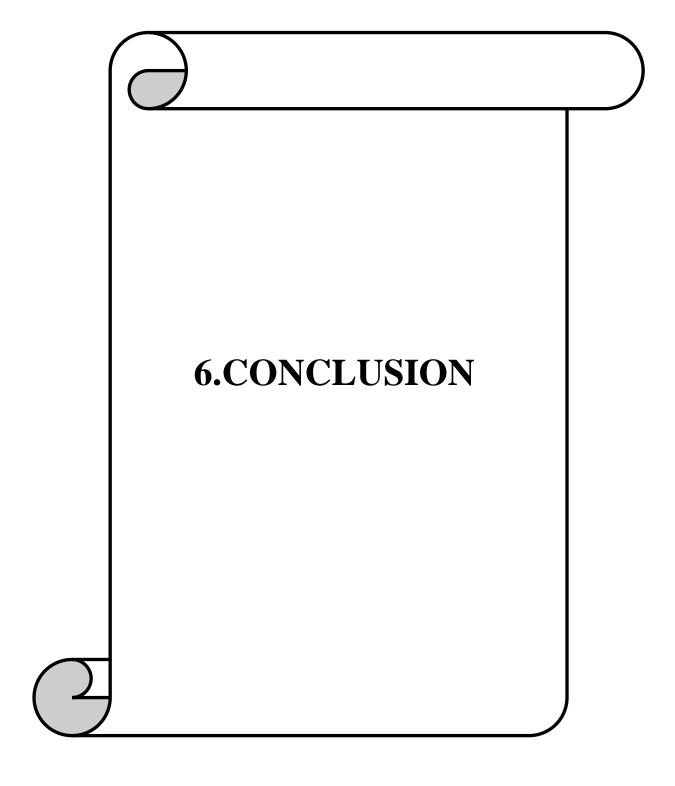










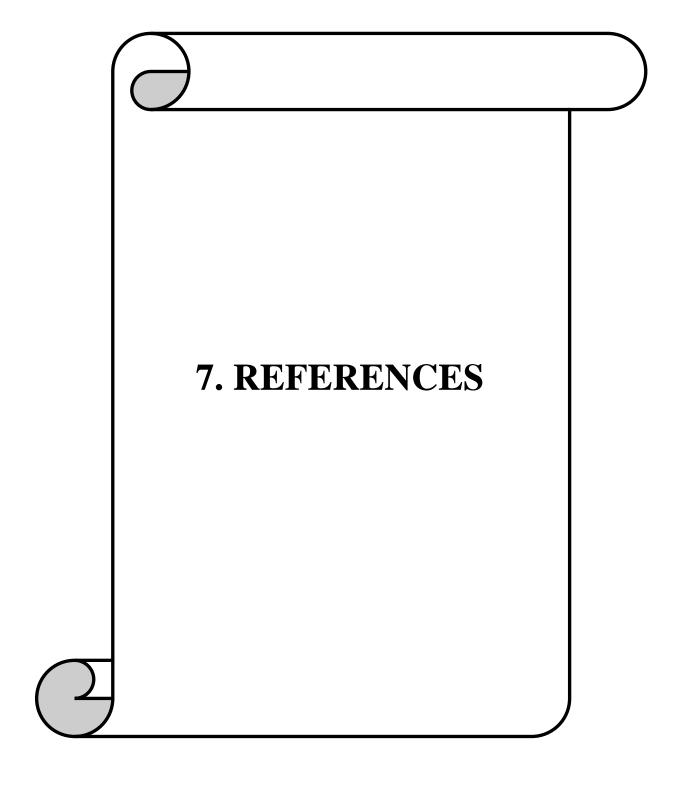


6.1 Conclusion

- We studied the efficient use of developing aural app an android application by making the great use of it.
- We focused on the authentication processs and privacy of user and the feasible way of converting PDF to audio.
- We use these results to determine the usage of this application is very easy and everyone can make use of it by easy and feasible ways of conversion.
- Main objective to develop the app was to provide conversion of PDF files to audio format.

6.2Future Enhancements

- In our project we can make a way for the user to Signin with Google, Facebook.
- Text to Audio conversion module can be added to users.
- We can allow users to download audio file into local storage.
- Storing converted audio file on the cloud.
- We can even develop our applications for IOS and Windows Operating System or Devices.



7.1 SITES:

- Firebase reference:- https://firebase.google.com/docs/android
- SpeechSynthesis:- https://developer.mozilla.org/en-US/docs/Web/API/SpeechSynthesis
- Node js reference :- https://nodejs.org/en/docs/
- React-Native :- https://reactnative.dev/docs/getting-started
- React-Native with Firebase tutorials: https://www.codersneverquit.in/
- pdf js :- https://mozilla.github.io/pdf.js/

7.2 BOOKS:

- Android Studio Development Essential By Neil Smith
- Software engineering seventh edition by Ian Sommerville.

7.3 Final Application:

• https://github.com/manojrayar/converter