

TITLE OF THE PROJECT

A PROJECT REPORT

Submitted for

CSE3009 – NoSQL Databases

by

Name-1 (17BCD7061)

Name-2 (17BCD7061)



**SCHOOL OF COMPUTER SCIENCE ENGINEERING
VIT-AP UNIVERSITY
AMARAVATI- 522237**

May 2021

ABSTRACT

This project focuses on creating a web application to provide effective assistance to medical personnels to generate a simple medical report from an audio file containing the details and medical conditions of the patient.

The main aim of this project is to help the Emergency Medical Technicians(EMTs) in the ambulance quickly generate a report of the patient's condition in the ambulance inorder to prepare in advance and start the treatment upon the patient's arrival without further delay. The application requires the EMT to enter an email id and select the audio file that he has recorded. When the medical report has been generated, it is automatically mailed to the given email id. This not only helps the doctor prepare for the patient's treatment in advance but also to instruct the EMT about first aids during emergencies.

TABLE OF CONTENTS

Chapter	Title	Page No.
	Abstract	3
1	Introduction	7
1.1	Objectives	7
1.2	Background and Literature Survey	8
1.3	Organization of the Report	8
2	Medical Assistance using AWS	9
2.1	Proposed System	9
2.2	Working Methodology	10
2.3	Software Details	13
2.3.1	Web Page	13
2.3.2	AWS Cloud Services	14
3	Database Analysis	22
3.1	Dataset used and attributes	22
4	Results and Discussion	23
5	Conclusion & Future Works	24
6	Appendix	25
7	References	36

CHAPTER 1

INTRODUCTION

The medical sector in India has been rapidly growing with the use of technology and the invention of new techniques for treatment. The technology has increased the provider capabilities resulting in improved quality of life. This growth cannot change the importance of time during emergencies.

The rate of deaths due to accidents has been rapidly increasing. Unfortunately, about 30% of these deaths occur in the ambulance due to either a delay in reaching the hospital or an emergency. An ambulance comes with an EMT who is not a doctor but is qualified to provide the patient with first aid. But sometimes, the absence of a doctor for immediate assistance can prove to be fatal. During emergencies, the need for a doctor's consultation in an ambulance has become important.

1.1 Objectives

The following are the objectives of this project:

- creating an application to provide effective assistance to medical personnels
- generate a simple medical report from an audio file containing the details and medical conditions of the patient by utilising the time taken to reach hospital more efficiently
- allow the doctor and/or hospital staff to prepare for the patient's treatment in advance
- facilitate the doctor to instruct the EMT to assist the patient in case of critical emergencies

1.2 Background and Literature Survey

On an average, it takes about 8-10 minutes for an ambulance to carry the patient to the hospital. During an emergency, these 8-10 minutes can make a big difference. An ambulance comes with an emergency medical technician(EMT), a health professional who provides emergency medical services. The idea of this project is to make the doctor at the hospital aware of the patient's condition within these 8-10 minutes.

The basis of this project comes for the two major problems. Firstly, in India, about 20% of emergency patients' deaths are caused during the travel time. Secondly, there are a number of general transcription softwares available on the internet. But they do not come with a feature to identify and categorise medical terminology. A majority of softwares that have the medical transcription feature, support only Windows and Android. (Ex: The FTW Transcriber, Express Scribe)

1.3 Organization of the Report

The remaining chapters of the project report are described as follows:

- Chapter 2 contains the proposed system, methodology and software details.
- Chapter 3 gives the cost involved in the implementation of the project.
- Chapter 4 discusses the results obtained after the project was implemented.
- Chapter 5 concludes the report.
- Chapter 6 consists of codes.
- Chapter 7 gives references.

CHAPTER 2

MEDICAL ASSISTANCE USING AWS

This Chapter describes the proposed system, working methodology and software details.

2.1 Proposed System

The following block diagram shows the system architecture of this project.

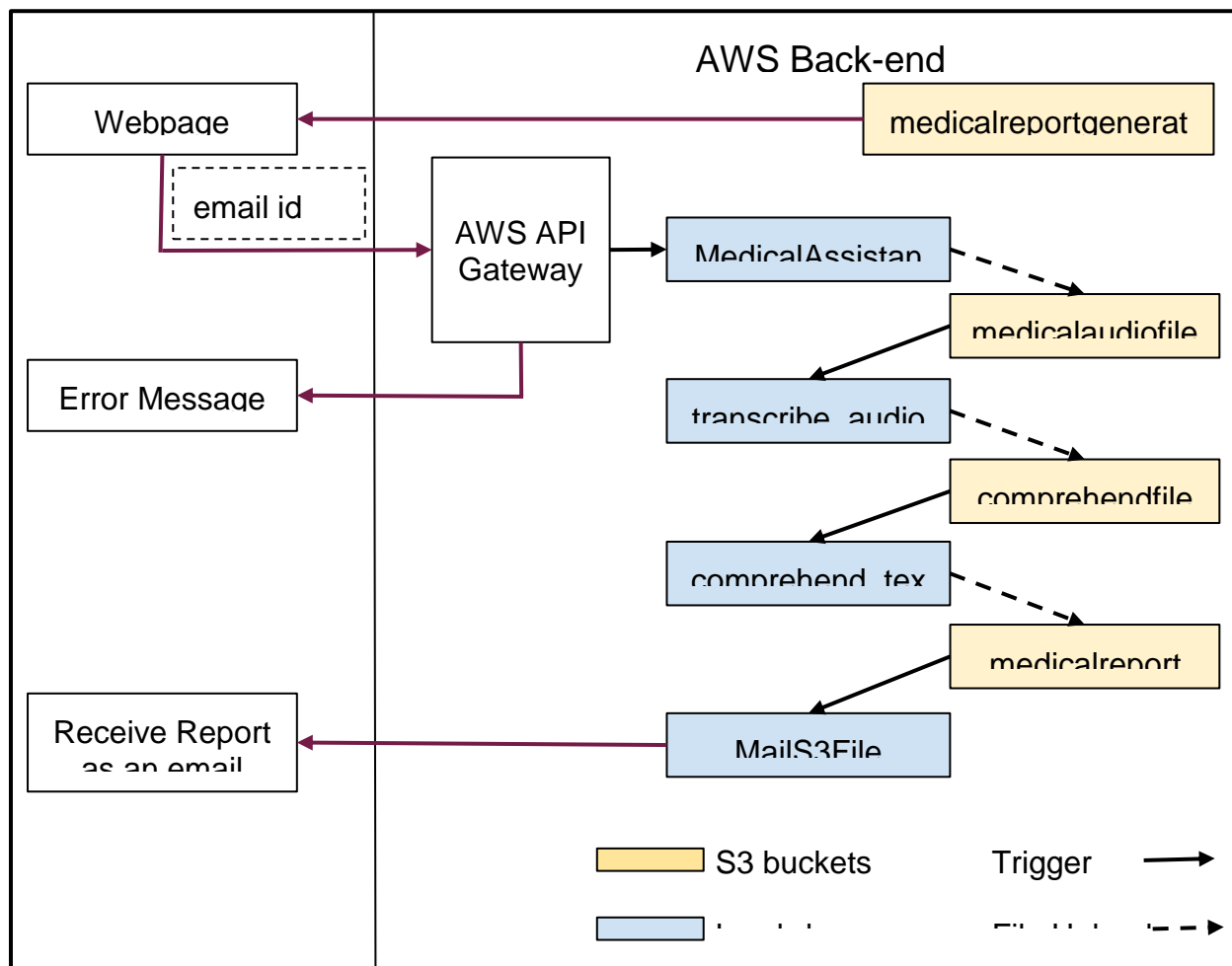


Figure 1. System Architecture Block Diagram

2.2 Working Methodology

The project is completely based on software with two sections, front-end and back-end. The front-end consists of an HTML(HyperText Markup Language) file and a CSS(Cascading Style Sheets) file. Being a static website, it has been hosted on the AWS cloud using the S3 bucket 'medicalreportgenerator'. The back-end is entirely on AWS built using various AWS services. The functions are automatically called in a fixed sequence with each function being triggered due to an upload action in the corresponding S3 bucket.

AWS Transcribe Medical is an extension of AWS Transcribe solely dedicated to medical data. It is a cost effective automatic speech recognition(ASR) service that adds accurate medical speech-to-text capabilities to the application. Using Machine Learning, it transcribes medical terminologies such as medicine names, procedures, conditions and diseases. The service being stateless prioritises patient data security and privacy, which means that it neither stores the audio file (input) nor the json file (output) containing the text.

AWS Comprehend Medical is an extension of AWS Comprehend which is a natural language processing (NLP) service that extracts complex medical information from unstructured text. It identifies medical information, such as medical conditions, medications, dosages, tests, treatments, procedures, and protected health information. It automates and lowers the cost of analysing unstructured data.

Simple Email Service(SES) is a flexible, scalable and cost-effective email service that enables the developer to send transactional, marketing and bulk emails from within the application. It makes it easy to configure the email and eliminates the challenges that are faced during building the infrastructure, configuring the network, warming up the IP addresses and protecting the sender reputation.

CHAPTER 3

DATABASE ANALYSIS

3.1 Datasets and attributes

AWS offers a free tier for 12 months with a monthly limit on few services. The individual costs of the various AWS Services used in this project are given below in Table 3.1. The total cost depends on the usage and varies for each service.

CHAPTER 4

RESULTS AND DISCUSSIONS

The application consists of a single web page allowing the user to fill a form with a valid email address and an audio file. When an invalid email address and/or an invalid file is submitted, an error message is displayed as shown in Figures 5(a) and 5(b).

The project abides by the objectives discussed and completes the entire process within two minutes which is clear from the 'Last modified' column in the Figures 6, 7 and 8 and the email received time from Figure 9.

CHAPTER 5

CONCLUSION AND FUTURE WORK

This web application builds a path to utilise every second to its maximum. It allows the doctor and the hospital to make the necessary arrangements for the patient's treatment far in advance before their arrival at the hospital saving a lot of time during a crucial situation.

The medical field and the technology, both are open to improvements with huge scopes. Lot can be done in this area. This project can be extended

- to support translation using AWS Translate.
- to share the current location of the ambulance to provide a heads-up on the patient's time of arrival.
- to support the storage and retrieval of reports by hospital staff as well as the patients.

CHAPTER 6

APPENDIX (ONLY NOSQL CODE)

Front-end (HTML) - fileUpload.html

```
<html>
  <head>
    <meta http-equiv="refresh" content="30">

    <link rel="stylesheet" href="style.css">

    <title>MedApp</title>
  </head>
  <body>
    <div class='center div1'>
      <h1>Medical Report Generator</h1>
    </div>
    <hr>
    <div class='center div2'>
      <h2>Upload an audio file containing the patient's details and the report will be mailed to
the email id entered.</h2>
    </div>
    <div class='center div3'>
      <form method="post" enctype="multipart/form-data"
action="https://7tplmesiaa.execute-api.us-east-2.amazonaws.com/audio/upload">
        <label><b>Email ID</b></label><br>
        <input type='text' id="mailid" name="mailid" required>
        <br><br>
        <label><b>Select an Audio File</b></label><br>
        <input type="file" id="myFile" name="file" required>
        <br>
        <button type="submit"><b>Upload</b></button>
      </form>
    </div>
    <div class="footer">
      <h5>This page will automatically refresh every 30 seconds</h5>
    </div>
  </body>
</html>
```

Front-end (CSS) - style.css

```
input {
  width: 95%;
  padding: 12px 20px;
  margin: 8px 0;
  font-size: 15px;
}
label {
  font-size: 22px;
}
.center {
  display: flex;
  justify-content: center;
  align-items: center;
}
form {
  padding: 40px;
  border-style: solid;
  border-width: 0.5px;
}
button {
  background-color: rgb(59, 236, 59);
  padding: 5px;
  width: 95%;
  padding: 12px 20px;
  font-size: x-large;
}
button:hover {
  background-color: rgb(6, 175, 6);
  color: white;
}
.div1 {
  font-size: xx-large;
}
body {
  background-image: url('bg.jpg');
  background-repeat: no-repeat;
  background-attachment: fixed;
  background-size: cover;
}
```

```
.footer {  
    position: fixed;  
    left: 0;  
    bottom: 0;  
    width: 100%;  
    color: black;  
    text-align: justify;  
}
```

Lambda Function 1 - MedicalAssistance.py

```
import json  
import boto3  
import re  
import time  
import base64  
import email  
from pprint import pprint  
from datetime import datetime  
  
s3 = boto3.client('s3')  
ses = boto3.client("ses")  
  
#Error messages have been added at the end of this function for better readability  
errorMsg = 'xyz' #invalid email id  
errorAudio = 'xyz' #uploaded file is not an audio file  
success = 'xyz' #successful execution  
  
def handler(event, context):  
    data = base64.b64decode(event['body'])  
    content_type = event["headers"]['content-type']  
    ct = "Content-Type: "+content_type+"\n"  
  
    # parsing message from bytes  
    msg = email.message_from_bytes(ct.encode()+data)  
  
    # if message is multipart  
    if msg.is_multipart():  
        multipart_content = { }  
        # retrieving form-data  
        for part in msg.get_payload():
```

```

        if part.get_filename():
            file_name = part.get_filename()
            multipart_content[part.get_param('name', header='content-disposition')] =
part.get_payload(decode=True)

audio = multipart_content['file']
key = datetime.now().strftime("%m%d%Y%H%M%S")
ext = file_name.split('.')[1]
fileName = key+'.'+ext

emailid = multipart_content['mailid'].decode("utf-8")
regex = '^[a-z0-9]+[\.\_]?[a-z0-9]+[@]\w+[\.\_]\w{2,3}$'
if(not re.search(regex,emailid)):
    return {
        'statusCode': 200,
        'body': errorMsg,
        'headers': {
            'Content-Type': 'text/html'
        }
    }

if(ext!='mp3' and ext!='mp4' and ext!='wav' and ext!='flac' and ext!='ogg' and ext!='amr' and
ext!='webm'):
    return {
        'statusCode': 200,
        'body': errorAudio,
        'headers': {
            'Content-Type': 'text/html'
        }
    }
data = s3.put_object(
    Bucket="medicalaudiofiles",
    Key=fileName,
    Body=audio,
    Metadata={'email':emailid}
)
time.sleep(20)

return {
    'statusCode': 200,

```

```
'body': success,
'headers': {
    'Content-Type': 'text/html'
}
}
```

Error Messages

```
<html>
  <head>
    <title>Error</title>
    <style>
      body {
        background-image: url('https://medicalreportgenerator.s3.us-east-
2.amazonaws.com/bg.jpg');
        background-repeat: no-repeat;
        background-attachment: fixed;
        background-size: cover;
      }
      .center {
        display: flex;
        justify-content: center;
        align-items: center;
      }
      .div1 {
        font-size: xx-large;
      } </style>
    </head>
    <body>
      <div class='center div1'>
        <h1>Medical Report Generator</h1>
      </div>
      <hr>
      <div class='center'>
        <h2>Invalid Email ID</h2> # errorMsg
        <h2>Upload Audio Files Only</h2> # errorAudio
        <h2>File is being processed. Mail will be sent soon.</h2> # Success
      </div>
    </body>
  </html>
```

CHAPTER 7

REFERENCES

1. <https://aws.amazon.com/blogs/machine-learning/performing-medical-transcription-analysis-with-amazon-transcribe-medical-and-amazon-comprehend-medical/>
2. <https://docs.aws.amazon.com/AmazonS3/latest/userguide/create-bucket-overview.html>
3. <https://docs.aws.amazon.com/AmazonS3/latest/dev/WebsiteHosting.html>
4. <https://docs.aws.amazon.com/apigateway/latest/developerguide/http-api.html>
5. <https://docs.aws.amazon.com/lambda/latest/dg/getting-started-create-function.html>
6. <https://docs.aws.amazon.com/lambda/latest/dg/services-apigateway.html>
7. <https://docs.aws.amazon.com/lambda/latest/dg/with-s3.html>
8. <https://docs.aws.amazon.com/transcribe/latest/dg/transcribe-medical.html>
9. <https://docs.aws.amazon.com/comprehend/latest/dg/comprehend-med.html>
10. <https://docs.aws.amazon.com/ses/latest/DeveloperGuide/send-email.html>