A

Project Report

On

‘Text-To-Speech using Node-red, NodeJs’

In partial fulfillment of requirements for the TAE Mini-modelling

In

Electronics & Telecommunication Engineering.

TAE: 03

SUBMITTED BY:

1. **Pranay Rathi.(BECA40)**
2. **Akshay Rangari(BECA39)**
3. **Vivek Patil(BECA37).**

Under the Guidance of

Dr. Vaishali Raut

DEPARTMENT OF ELECTRONICS & TELECOMMUNICATION ENGINEERING

G.H. Raisoni College of Engineering & Management, Pune

Academic Year 2019-20

G.H. Raisoni College of Engineering and Management, Wagholi, Pune

CERTIFICATE

This is certified that Mr. Pranay Rathi has successfully completed Project work entitled Text-To-Speech using Node-red, NodeJs in partial fulfillment of the requirement for the mini modelling.

The matter embodied in this Project Report is a record of his own independent work carried out by him under my supervision and guidance. The matter embodied in this report has not been submitted for any award of any Degree or Diploma.

Dr. Vaishali Raut	Prof. S. K. Waghmare	Dr. S. N. Mali
Project Guide	HOD, E&TC	Director

College of Engineering and Management, Wagholi, Pune

CERTIFICATE

This is certified that Mr. Akshay Rangari has successfully completed Project work entitled Text-To-Speech using Node-red, NodeJs in partial fulfillment of the requirement for the Mini modelling.

The matter embodied in this Project Report is a record of his own independent work carried out by him under my supervision and guidance. The matter embodied in this report has not been submitted for any award of any Degree or Diploma.

Dr. Vaishali Raut	**Prof. S. K. Waghmare**	**Dr. S. N. Mali**
Project Guide	**HOD, E&TC**	**Director**

College of Engineering and Management, Wagholi, Pune

CERTIFICATE

This is certified that Mr. Vivek Patil has successfully completed Project work entitled Text-To-Speech using Node-red, NodeJs in partial fulfillment of the requirement for the mini modelling.

The matter embodied in this Project Report is a record of his own independent work carried out by him under my supervision and guidance. The matter embodied in this report has not been submitted for any award of any Degree or Diploma.

Dr. Vaishali Raut	Prof. S. K. Waghmare	Dr. S. N. Mali
Project Guide	HOD, E&TC	Director

ABSTRACT

Most of the information in world of computer is accessible to a few who can read or understand a particular language. But it could be very much helpful for the common man if the computer talks to him in his language. Text-To-Speech (TTS) is a technology that converts a written text into human understandable voice. A TTS synthesizer is a computer based system that can be able to read any text aloud that is given through standard input devices.

Node-RED is a programming tool for wiring together hardware devices, APIs and online services in new and interesting ways.

It provides a browser-based editor that makes it easy to wire together flows using the wide range of nodes in the palette that can be deployed to its runtime in a single-click. Node-RED is open source and was originally created by the IBM Emerging Technology organization. It is included in IBM's Bluemix (a Platform-as-a-Service or PaaS) IoT starter application package. Node-RED can also be deployed separately using the Node.js application. At present, Node-RED is a JS Foundation project.

CONTENTS

ACKNOWLEDGEMENT

We would like to express our deepest appreciation to all those who provided us the possibility to complete this report. We take this opportunity to express our profound gratitude and deep regard to our guide **Dr. Vaishali Raut** for her exemplary guidance, monitoring and constant encouragement throughout the course of this project.

We would like to express our gratitude towards member of our Department for their kind co-operation and encouragement which help us in completion of this project. We would like to express our special gratitude and thanks to industry persons for giving us such attention and time.

Our thanks and appreciations also go to our colleagues in developing the project and people who have willingly helped us out with their abilities.

INTRODUCTION

Node-RED is a programming tool for wiring together hardware devices, APIs and online services in new and interesting ways.

Internet o Things is defined as a network of physical objects or 'things' embedded with electronics, software, sensors etc. For the various applications of Internet of things, pulling together different hardware devices, APIs and online services in new ways are required. A lot of time is wasted in writing the code to the different systems, such as accessing a serial port. Therefore, tools are required to make it easier for developers at all levels to bring together the different streams of events, both physical and digital. "Node-RED is a visual tool for wiring the Internet of Things" [8], but can also be used for other types of applications to quickly assemble flows of services. It is developed by IBM emerging technology and the open source network. The basic idea behind the development of Node-RED was to build toolbox of reusable codes that people could use to wire together software and hardware systems, that will develop into useful systems. It is an event processing engine, built on Node.js that makes the job of software and hardware hackers easier. It does not take away the requirement of coding altogether, but reduces it to quite an extent. It allows the developers to visually construct flows by connecting simple 'nodes'. The 'nodes' represent reusable pieces of code and logic and make the job of developing an application way simpler. Node-RED is a open source project and the community around Node-RED is constantly evolving and more and more number of Node-RED nodes are being developed. The fact that it can run on low- cost hardware or in the cloud certainly works in the favour of Node-RED.

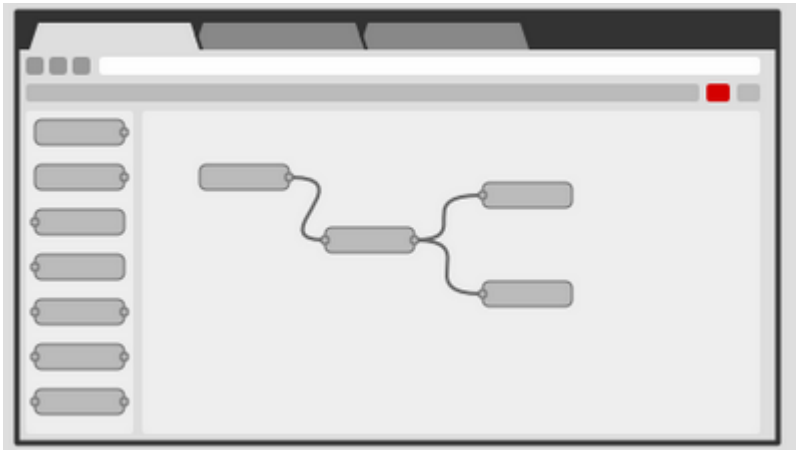
It provides a browser-based editor that makes it easy to wire together flows using the wide range of nodes in the palette that can be deployed to its runtime in a single-click. Node-RED is open source and was originally created by the IBM Emerging Technology organization. It is included in IBM's Bluemix (a Platform-as-a-Service or PaaS) IoT starter application package. Node-RED can also be deployed separately using the Node.js application. At present, Node-RED is a JS Foundation project.

Since early 2014, the mobile has overtaken the personal computer (desktop/laptop) as the leading device used to navigate the Net. Along with the mobile, a number of other portable devices that connect to the Internet have also started proliferating at a very quick rate. Nowadays, most of us carry or possess at least one Internet based device and a mobile. So, the Internet of Things (IoT) now doesn't only mean different 'things', but has evolved into 'intelligent things' which have on-board computation and network connections. Most importantly, they have the capability to sense the environment around us and, accordingly, act intelligently. These devices are now referred to as connected devices, smart objects or the Web of Things.

This adaptation of technology has led to many developments in the visual programming and visual coding arenas. As a result, Node-RED has evolved into a visual programming/coding tool based on the already popular Node.js (a server side Java Scripting platform), mainly targeting the Internet of Things space.

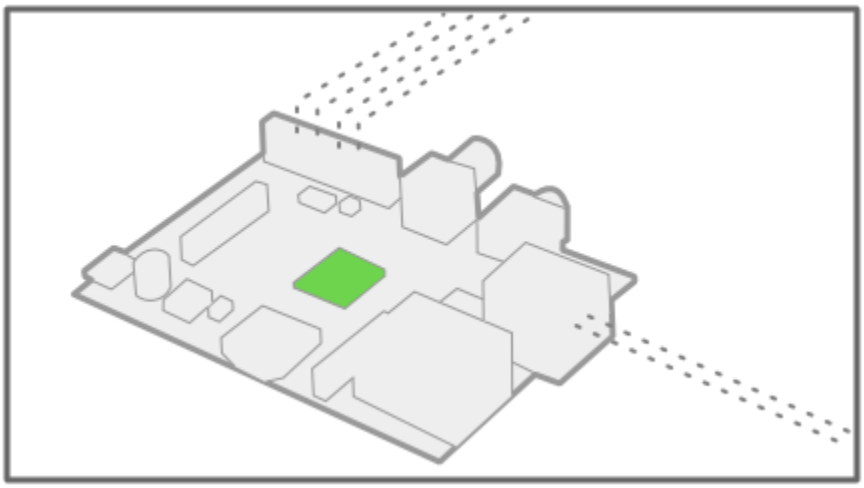
Features of Node-RED

- **Browser-based flow editing**



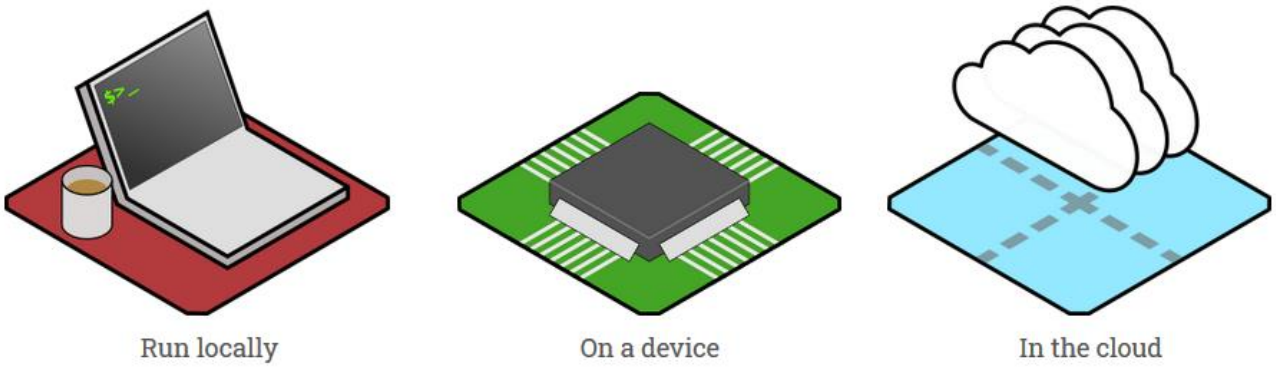
- Node-RED provides a browser-based flow editor that makes it easy to wire together flows using the wide range of nodes in the palette. Flows can be then deployed to the runtime in a single-click.
- JavaScript functions can be created within the editor using a rich text editor.
- A built-in library allows you to save useful functions, templates or flows for re-use.

- **Built on Node.js**
- The light-weight runtime is built on Node.js, taking full advantage of its event-driven, non-blocking model. This makes it ideal to run at the edge of the network on low-cost hardware such as the Raspberry Pi as well as in the cloud.
- With over 225,000 modules in Node's package repository, it is easy to extend the range of palette nodes to add new capabilities.



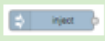
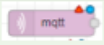
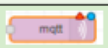
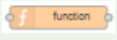
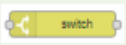
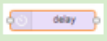
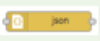
- **Social Development**
- The flows created in Node-RED are stored using JSON which can be easily imported and exported for sharing with others.
- An online flow library allows you to share your best flows with the world.

- **Get Started**



- Node-RED is built on Node.js, taking full advantage of its event-driven, non-blocking model. This makes it ideal to run at the edge of the network on low-cost hardware such as the Raspberry Pi as well as in the cloud.

Commonly used Nodes

Node	Node	Description
Inject		Injects a message into a flow either manually or at regular intervals. The message payload can be a variety of types, including strings, JavaScript objects or the current time.
Mqtt-IN		Connects to a MQTT broker and subscribes to messages from the specified topic
Mqtt-Out		Connects to a MQTT broker and publishes messages.
Function		1. A JavaScript function block to run against the messages being received by the node. 2. The function is expected to return a message object or null 3. The function block can have multiple return and output
Trigger		When triggered, can send a message, and then optionally a second message, unless extended or reset.
Switch		Route messages based on their property values or sequence position
Delay		Delays each message passing through the node or limits the rate at which they can pass.
Json		Converts between a JSON string and its JavaScript object representation, in either direction.

Creating your first flow of Node-red

1. Access the editor

With Node-RED running, open the editor in a web browser.
If you are using a browser on the same computer that is running Node-RED, you can access it with the url: http://localhost:1880.
If you are using a browser on another computer, you will need to use the ip address of the computer running Node-RED: http://<ip-address>:1880.

2. Add an Inject node

The Inject node allows you to inject messages into a flow, either by clicking the button on the node, or setting a time interval between injects.
Drag one onto the workspace from the palette.
Select the newly added Inject node to see information about its properties and a description of what it does in the Information sidebar pane.

3. Add a Debug node

The Debug node causes any message to be displayed in the Debug sidebar. By default, it just displays the payload of the message, but it is possible to display the entire message object.

4. Wire the two together
Connect the Inject and Debug nodes together by dragging between the output port of one to the input port of the other.

5. Deploy

At this point, the nodes only exist in the editor and must be deployed to the server.
Click the Deploy button.
With the Debug sidebar tab selected, click the Inject button. You should see numbers appear in the sidebar. By default, the Inject node uses the number of milliseconds since January 1st, 1970 as its payload.

6. Add a Function node

The Function node allows you to pass each message though a JavaScript function.
Delete the existing wire (select it and press delete on the keyboard).
Wire a Function node in between the Inject and Debug nodes.
Double-click on the Function node to bring up the edit dialog. Copy the following code into the function field:

```
// Create a Date object from the payload
```

```
var date = new Date(msg.payload);
// Change the payload to be a formatted Date string
msg.payload = date.toString();
// Return the message so it can be sent on
return msg;
```

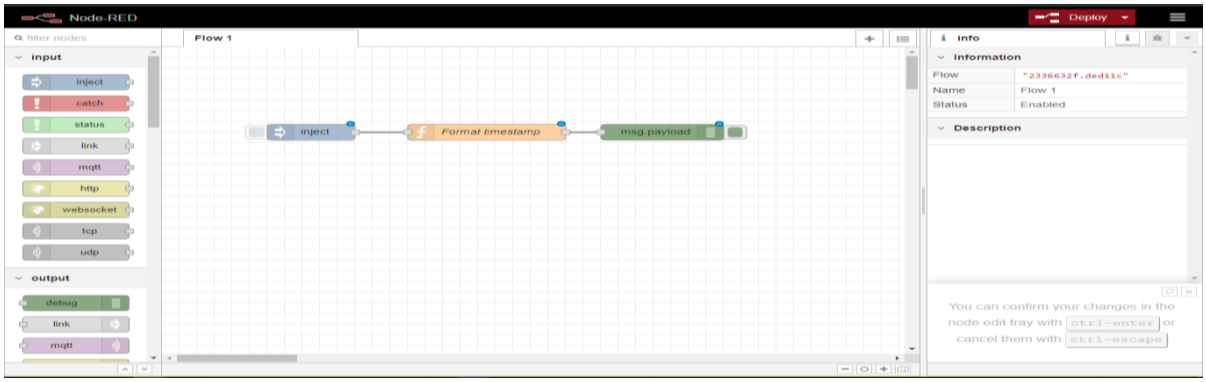
Click Done to close the edit dialog and then click the deploy button.
Now when you click the Inject button, the messages in the sidebar will now be formatted is readable timestamps.

Summary

This flow demonstrates the basic concept of creating a flow. It shows how the Inject node can be used to manually trigger a flow, and how the Debug node displays messages in the sidebar. It also shows how the Function node can be used to write custom JavaScript to run against messages.

Source


```
[{"id":"58ffae9d.a7005","type":"debug","name":"","active":true,"complete":false,"x":640,"y":200,"w
ires":
[]},{"id":"17626462.e89d9c","type":"inject","name":"","topic":"","payload":"","repeat":"","once":f
alse,"x":240,"y":200,"wires":[["2921667d.d6de9a"]]},
{"id":"2921667d.d6de9a","type":"function","name":"Format timestamp","func":"// Create a Date
object from the payload\nvar date = new Date(msg.payload);\n// Change the payload to be a
formatted Date string\nmsg.payload = date.toString();\n// Return the message so it can be sent
on\nreturn msg;","outputs":1,"x":440,"y":200,"wires":[["58ffae9d.a7005"]]}]
```

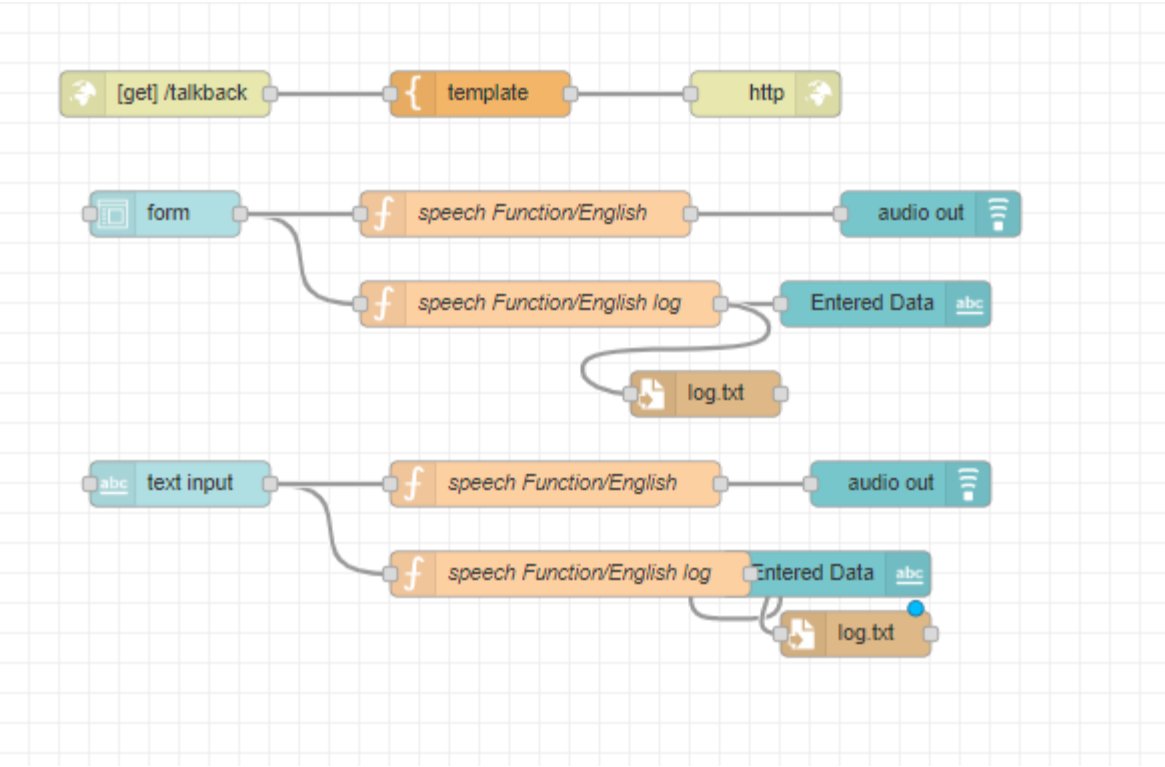
SYSTEM DEVELOPMENT.

I. Algorithm

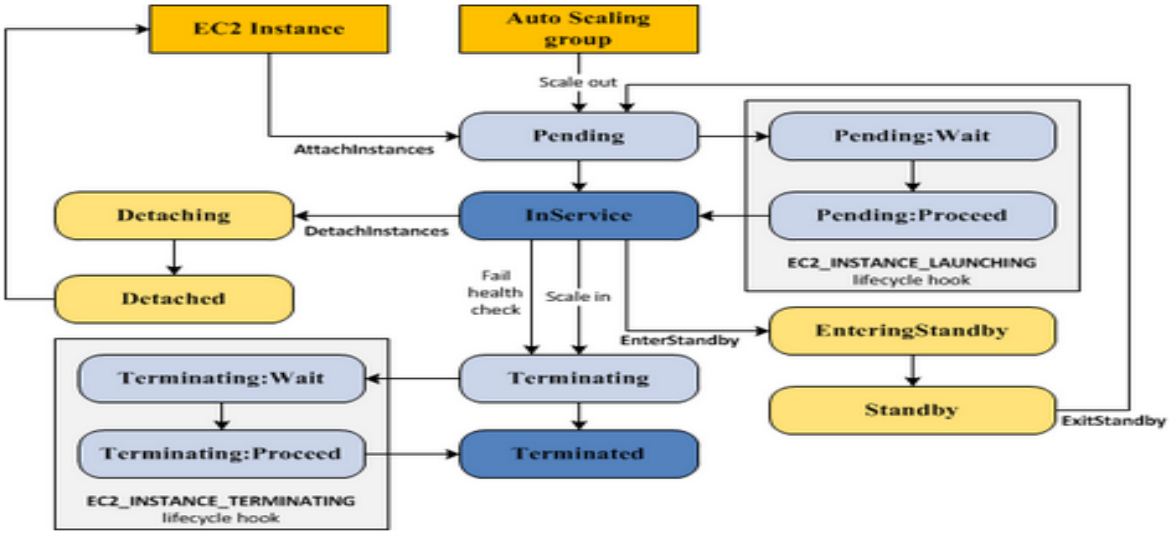
High level architecture design of project.

i. Start
ii. Create front end using Angular
iii. Create data base
iv. Deploy project on nodeJs server
v. End

II. Block diagram



III. Flowchart



iv. Methodology

As a group of students, there are many different ways to deliver projects. Broadly speaking, these ways are our project management models—applying different principles, themes, frameworks, processes and standards to help provide structure to the way we deliver projects. In other words, I don't think a methodology has to be a complete full-stack implementation "system" to be considered a methodology.

One of the more traditional project management methodologies, Waterfall is a linear, sequential design approach where progress flows downwards in one direction—like a waterfall. Originating in the manufacturing and construction industries, its lack of flexibility in design changes in the earlier stages of the development process is due to it becoming exuberantly more expensive because of its structured physical environments.

The methodology was first introduced in an article written in 1970 by Winston W. Royce (although the term 'Waterfall' wasn't used), and emphasizes that you're only able to move onto the next phase of development once the current phase has been completed. The phases are followed in the following order:

1. System and software requirements
2. Analysis
3. Design
4. Coding
5. Testing
6. Operations

Waterfall is a project management methodology that stresses the importance of documentation. The idea is that if a worker was to leave during the development process, their replacement can start where they left off by familiarizing themselves with the information provided on the documents.

Pre-Agile saw the Waterfall methodology being used for software development, but there were many issues due to its non-adaptive design constraints, the lack of customer feedback available during the development process, and a delayed testing period.

Best suited for: Larger projects that require maintaining stringent stages and deadlines, or projects that have been done various times over where chances of surprises during the development process are relatively low.

Hardware & Software used:

Minimum Hardware Requirements of PC:

- Processor : Intel core i3 and Above
- RAM : 4GB

Software Requirements:

- NodeJS
- Npm
- Node-red

The Benefits of Text to Speech (TTS)

Businesses, Organizations, and Publishers that integrate TTS technology benefit from:

- **Enhanced customer experience** – Speech-enabling pre- and after-sales service minimizes human agent workload, provides personalized services, accelerates throughput, and reduces operational costs.
- **Effective branding across touchpoints** – A single TTS voice across multiple contact points supports consistent, emotional branding.
- **Global market penetration** – clear, lifelike, and customizable TTS voices from around the world extend the reach of your business.
- **Optimized development and maintenance** – Robust TTS technology that supports the most widespread platforms, is available both in the cloud and on premises, and is scalable according to actual business needs, saves development and maintenance efforts.
- **More autonomy for the digital content owner** – Many think that text-to-speech software is something to be downloaded manually. It can be, but there are cloud-based forms, or Software as a Service (SaaS). With a few simple lines of code, the audio is generated instantly and for new or updated content, the spoken version is updated automatically.
- **Increased web presence** – Websites with TTS technology attract some of the 774 million people worldwide with literacy issues and the 285 million people with visual impairments. Also, speech enabling web content does not interfere with usability for those without disabilities. It actually aids all other populations, including older users and foreign/non-native speakers.
- **Saved time and money** – With TTS technology that is web- or cloud-based on a SaaS (Software as a Service) platform, online content can quickly and easily be speech enabled, and maintenance is minimal.
- **Easier implementation with Internet of Things (IoT)** – The IoT is becoming a critical factor in digital business transformation. Companies across all verticals have digital marketing strategies in place and focus on engaging customers across various connected channels to optimize how they interact with them. TTS gives connected devices in the IoT a more user-friendly way to communicate with consumers.
- **Word-of-mouth marketing** – Adding an alternative way to consume content online enhances the user experience. Visitors are far more likely to return to and recommend websites where they have had positive experiences. And even in our age of social media marketing, word of mouth is still the most important platform, according to Forbes.
- **Enhanced employee performance with corporate learning programs** – With TTS technology, HR departments and e-learning professionals can make learning modules and employee training much easier for employees to learn anywhere and at any time.

TTS Benefits for End Users

Text to speech has benefits for everyone and here are some points that break down how specific groups see a better user experience:

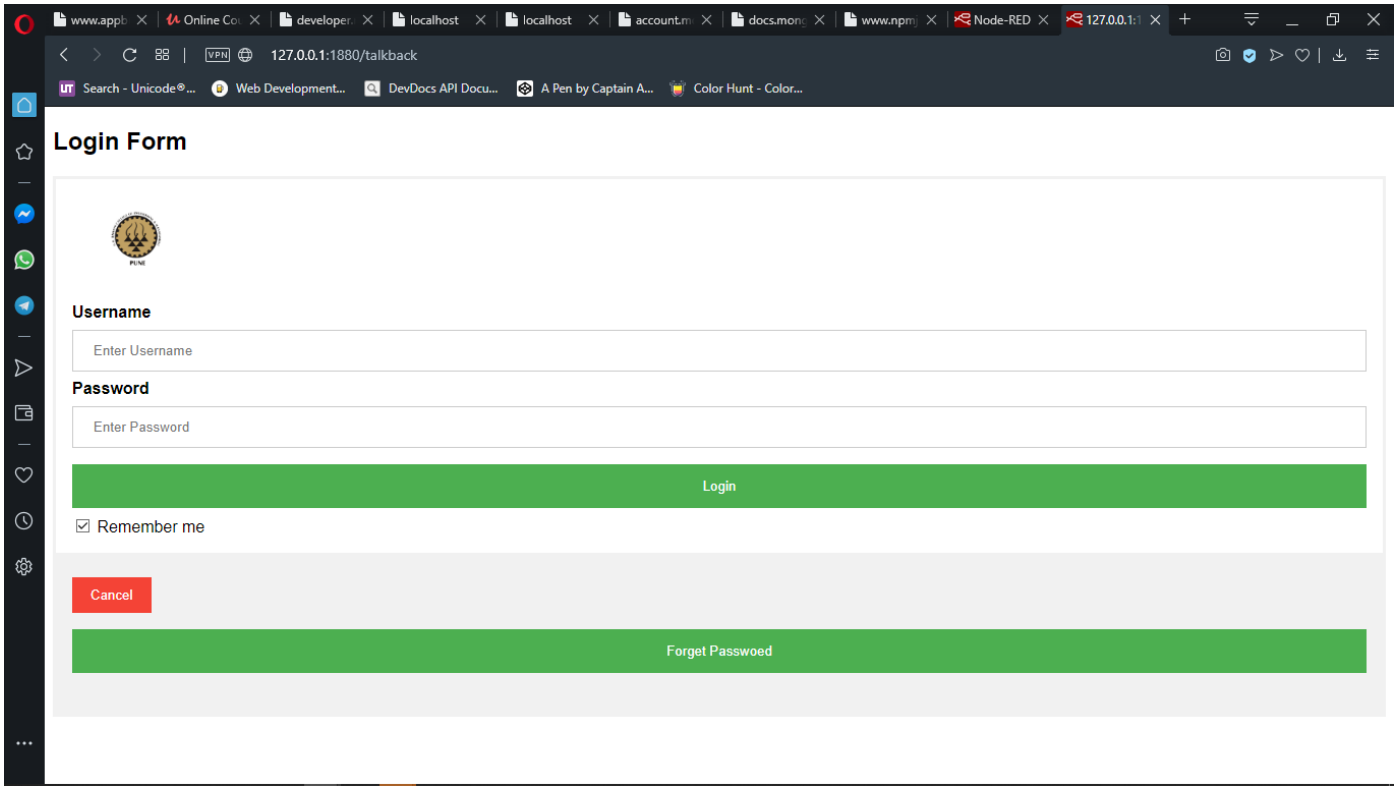
- **Extend the reach of your content** – TTS gives access to your content to a greater population, such as those with literacy difficulties, learning disabilities, reduced vision and those learning a language. It also opens doors to anyone else looking for easier ways to access digital content.
- **Accessibility is relevant** – Did you know that 15-20 percent of the worldwide population has some form of language-based learning disability? Did you know that 14 percent of adults in the US are illiterate and many have only basic reading skills? Making your online content audible helps the online population to better understand the text. The text is read and highlighted simultaneously so that the reader may easily follow along.
- **Populations are evolving** – 244 million people are foreign born across the globe (an increase of 70 million since 2000). Language proficiency and schooling in the host country's language is a very real problem for migrants and their families.
- **A growing elderly population depends on technology** – Between 2015 and 2030, the number of people aged 60 years or over will grow by 56 percent, from 901 million to 1.4 billion (Source: www.un.org). In the US alone, 59% of senior citizens use the Internet daily. Making digital content on the Internet accessible in multiple forms creates an easier user experience.
- **People are increasingly mobile and looking for convenience** – In the US, a growing share of time spent on digital content is on mobile devices and the demand for connected devices continues to rise on a worldwide scale. Text to speech can turn any digital content into a multimedia experience and people can listen to a news or blog article, a PDF document, or an e-book on the go!
- **People with different learning styles** – Some people are auditory learners, some are visual learners, and some are kinesthetic learners – most learn best through a combination of the three. Universal Design for Learning is a plan for teaching which, through the use of technology and adaptable lesson plans, aims to help the maximum number of learners comprehend and retain information by appealing to all learning styles.

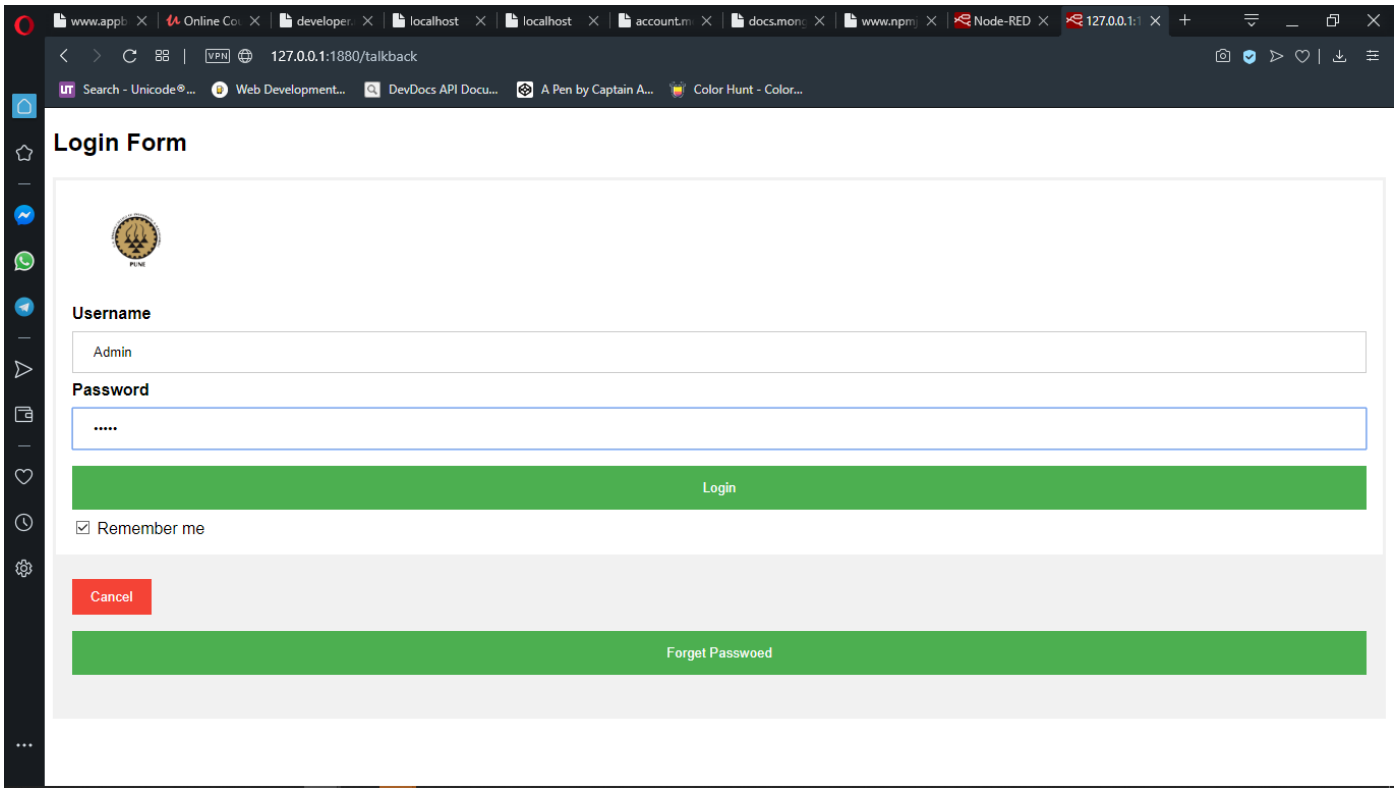
RESULTS AND DISCUSSIONS

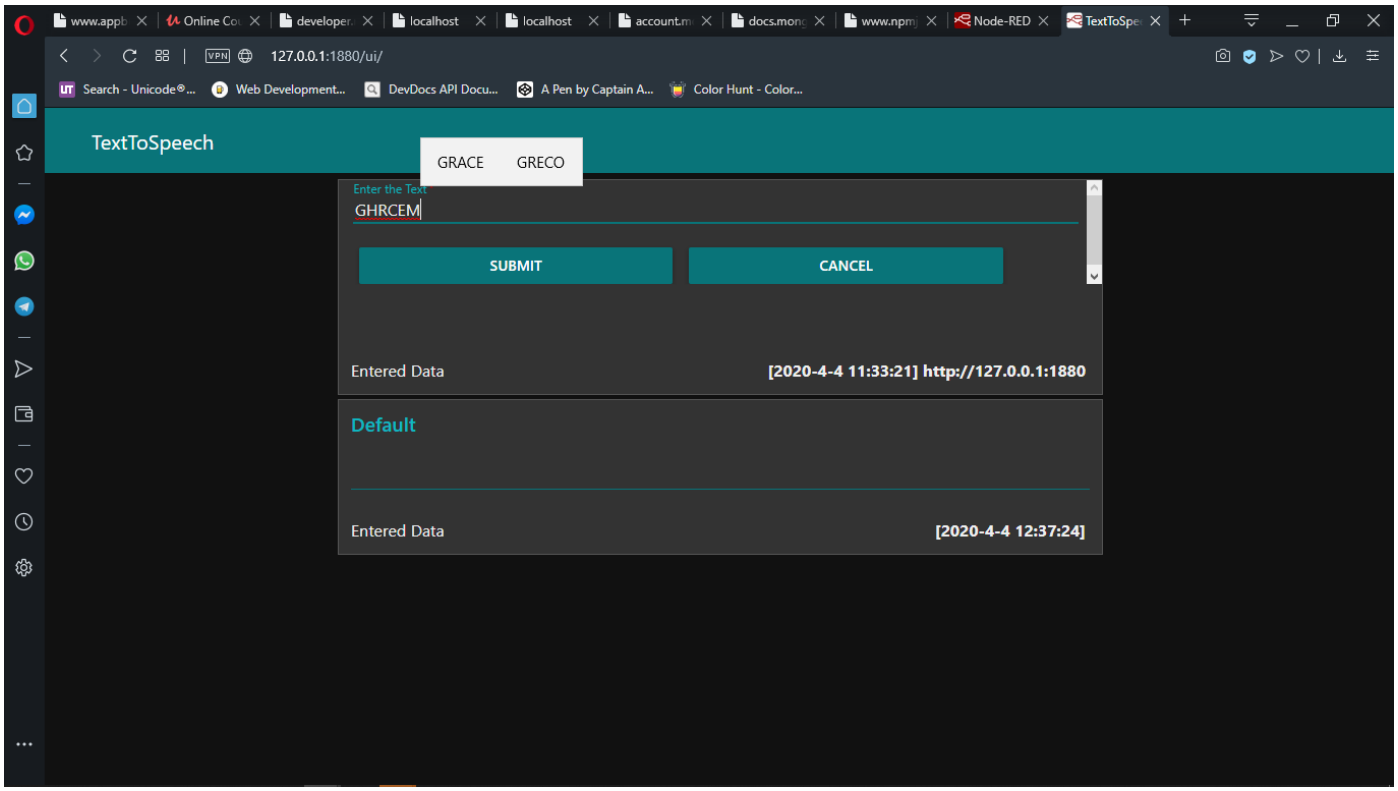
Deeds complete.

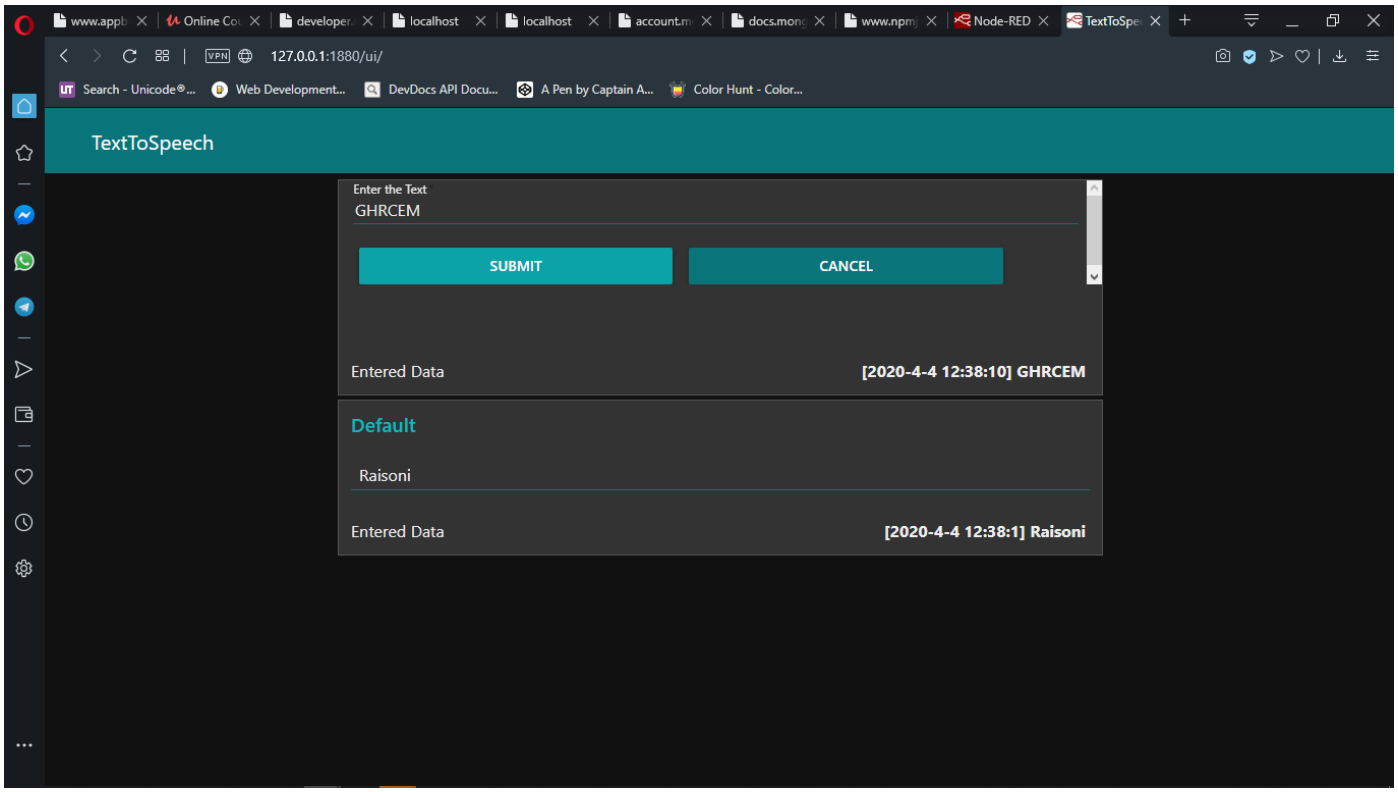
1. Create login page.
2. Create registration page.
3. Dashboard.
4. Log method

Snapshots of work.



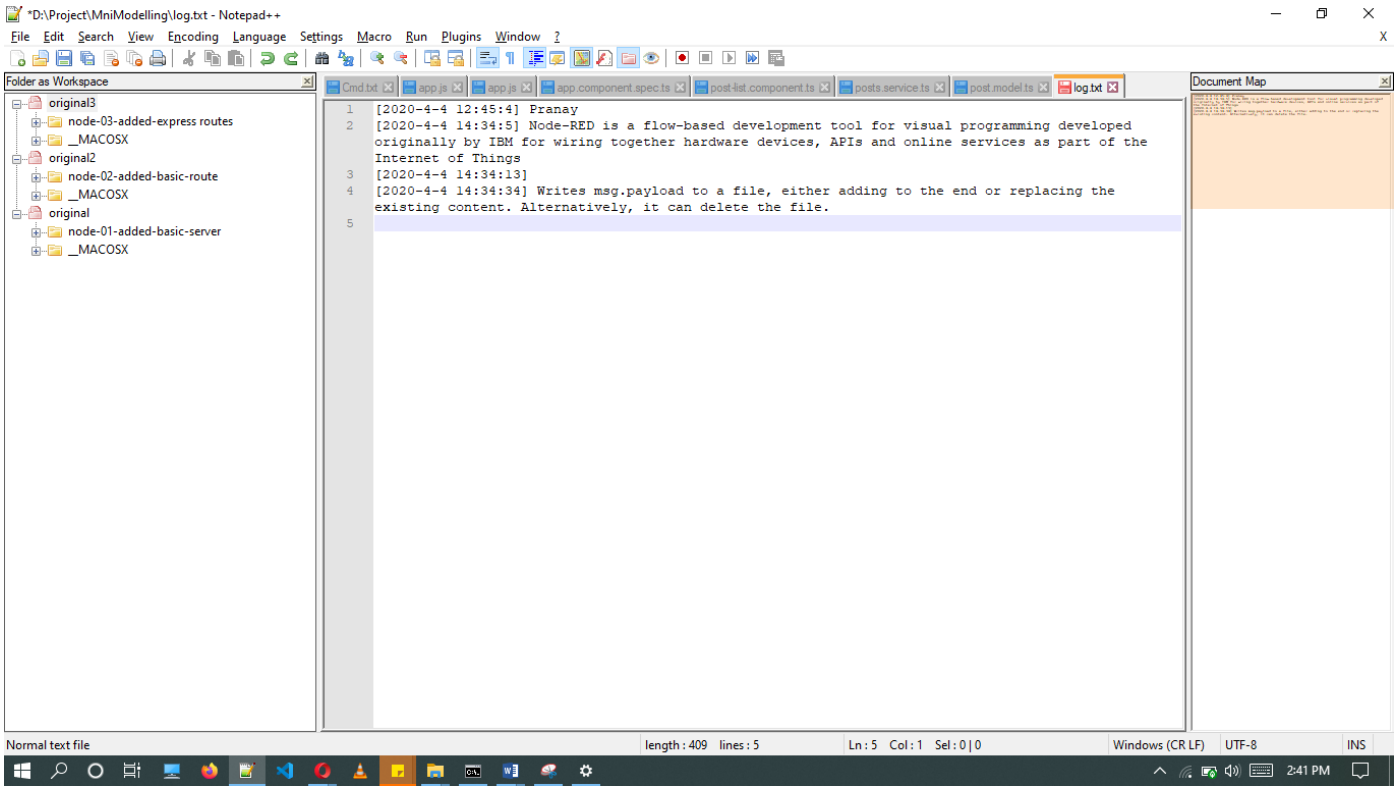

127.0.0.1:1880/talkback
Search - Unicode®...
Web Development...
DevDocs API Docu...
A Pen by Captain A...
Color Hunt - Color...
Login Form
Username
Admin
Password
.....
Login
Remember me
Cancel
Forget Passwoed


127.0.0.1:1880/ui/
Search - Unicode®...
Web Development...
DevDocs API Docu...
A Pen by Captain A...
Color Hunt - Color...
TextToSpeech
GRACE
GRECO
Enter the Text
GHRCEM
SUBMIT
CANCEL
Entered Data
[2020-4-4 11:33:21] http://127.0.0.1:1880
Default
Entered Data
[2020-4-4 12:37:24]

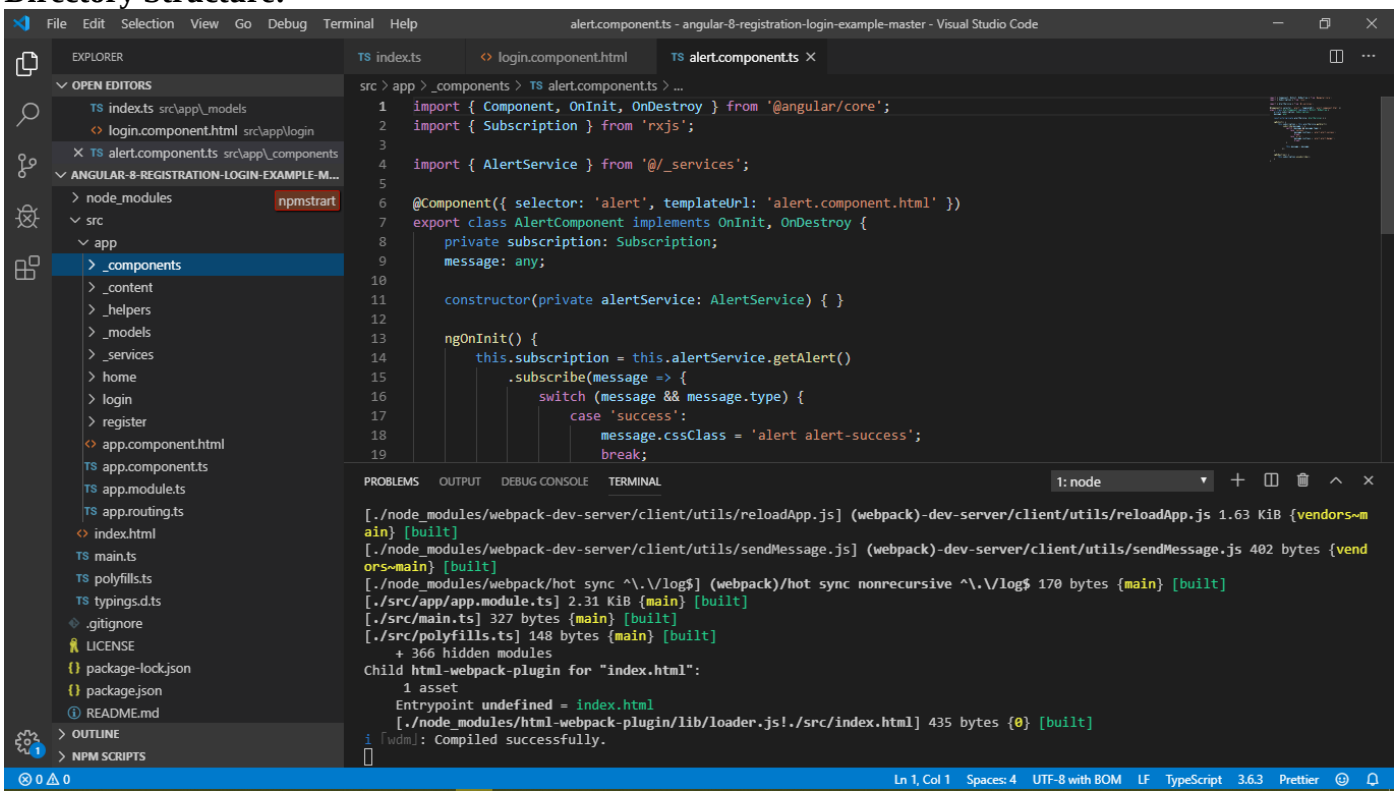


Log file data:

[2020-4-4 12:45:4] Pranay
[2020-4-4 14:34:5] Node-RED is a flow-based development tool for visual programming developed originally by IBM for wiring together hardware devices, APIs and online services as part of the Internet of Things
[2020-4-4 14:34:13]
[2020-4-4 14:34:34] Writes msg.payload to a file, either adding to the end or replacing the existing content. Alternatively, it can delete the file.



Directory Structure:



Discussion:

- **Instant Elasticity & Flexible Capacity: (scaling up and down)** Eliminate guessing on your infrastructure capacity needs.
- **Speed & Agility:** Develop and deploy applications faster Instead of waiting weeks or months for hardware to arrive and get installed.
- **Apps not Ops:** Focus on projects. Lets you shift resources away from data center investments and operations and move them to innovative new projects.
- **Global Reach:** Take your apps global in minutes.
- **Open and Flexible:** You choose the development platform or programming model that makes the most sense for your business.
- **Secure:** Allows your application to take advantage of the multiple layers of operational and physical security in the AWS data centers to ensure the integrity and safety of your data.

CONCLUSION

Modern Management can achieve its true potentials
in the shifting environment of modern times
with the proper blend of science and Judgement skills. *- Patrick J.Robinson*

Because of Text-to-speech, there has been many improvements to society, including those who can't see or speak to now be able to do so. For ages, people with diseases like MND, ALS, PBP, PLS, PMA, and so forth have made it so children and adults are cursed to either never talk or never see. With Text-to-speech, those innocent people now can express themselves through speech, or enjoy the pleasures of a good book. That is why companies like CereProc and Assistive Ware make Text-to-Speech and Speech Synthesis programs. Some children and adults were deprived of the pleasures that we have every day, and they wanted to change that.

REFERENCES

- Inventory management in standard pesticides pvt ltd; Hardik.J.Choksi;2007-2009; MBA; Gujarat technological university
- AngularJS API documentation. https://docs.angularjs.org/, 2016. [Online; accessed October-2016]
- Chandermani Arora. AngularJS performance, 2015. [Online; accessed September-2016].
- Sebastian Fröstl. AngularJS performance tuning for long lists. 2013. [Online; accessed September-2016].
- Footlik, R.B. (2004). The verification of the method the ABC. Problems and judgments. Logistics, 1, p. 22.
- Kharitonov, A.S. (1999). Physical beginnings of the theory of sustainable development of nature and society. "AVANTA +". Moscow.
- Sullivan B. (2016). “Amazon Web Services Public Cloud”, [Online]. Available: http://www.techweekeurope.co.uk/cloud/cloudmanagement/amazon-web-services-public-cloud185687
- "Can AWS Microsoft reach 50 market share cloud", (2015), [Online]. Available: http://marketrealist.com/2015/11/can-aws-microsoftreach-50-market-share-cloud/

Code:

```
[
    {
        "id": "a8f9e09.8028f2",
        "type": "tab",
        "label": "Flow 1",
        "disabled": false,
        "info": ""
    },
    {
        "id": "db5c8809.55a4a8",
        "type": "ui_audio",
        "z": "a8f9e09.8028f2",
        "name": "",
        "group": "b54caba6.18e1e8",
        "voice": "en-US",
        "always": "",
        "x": 740,
        "y": 180,
        "wires": []
    },
    {
        "id": "f63855d2.ec28d8",
        "type": "ui_form",
        "z": "a8f9e09.8028f2",
        "name": "",
        "label": "",
        "group": "b54caba6.18e1e8",
        "order": 1,
        "width": "14",
        "height": "15",
        "options": [
            {
                "label": "Enter the Text",
                "value": "payload",
                "type": "text",
                "required": true,
                "rows": null
            }
        ],
        "formValue": {
            "payload": ""
        },
        "payload": "",
        "submit": "submit",
        "cancel": "cancel",
        "topic": "",
        "x": 230,
        "y": 180,
        "wires": [
            [
                "ecd646fe.cc4408",
                "c73febc6.b6f6f8"
            ]
        ]
    },
    {
        "id": "f89acb77.d533d8",
        "type": "ui_text",
```

```
        "z": "a8f9e09.8028f2",
        "group": "b54caba6.18e1e8",
        "order": 5,
        "width": 0,
        "height": 0,
        "name": "",
        "label": "Entered Data",
        "format": "{{msg.payload}}",
        "layout": "row-spread",
        "x": 710,
        "y": 240,
        "wires": []
    },
    {
        "id": "ecd646fe.cc4408",
        "type": "function",
        "z": "a8f9e09.8028f2",
        "name": "speech Function/English log",
        "func": "let data = msg.payload.payload;\nvar today = new Date();\nvar date = today.getFullYear()+'-'+(today.getMonth()+1)+'-
'+today.getDate();\nvar time = today.getHours() + \":\" + today.getMinutes() + \":\" + today.getSeconds();\nvar dateTime ='['+ date+' '+time +']
';\nvar log = dateTime + data\nreturn {payload : log} ;",
        "outputs": 1,
        "noerr": 0,
        "x": 480,
        "y": 240,
        "wires": [
            [
                "f89acb77.d533d8",
                "fdc15456.0a7168"
            ]
        ]
    },
    {
        "id": "c73febc6.b6f6f8",
        "type": "function",
        "z": "a8f9e09.8028f2",
        "name": "speech Function/English",
        "func": "\nreturn msg.payload;",
        "outputs": 1,
        "noerr": 0,
        "x": 470,
        "y": 180,
        "wires": [
            [
                "db5c8809.55a4a8"
            ]
        ]
    },
    {
        "id": "fdc15456.0a7168",
        "type": "file",
        "z": "a8f9e09.8028f2",
        "name": "",
        "filename": "log.txt",
        "appendNewline": true,
        "createDir": true,
        "overwriteFile": "false",
        "encoding": "none",
        "x": 590,
        "y": 300,
        "wires": [
            []
```

```
    ]
  },
  {
    "id": "a258b0e6.aa83f",
    "type": "http in",
    "z": "a8f9e09.8028f2",
    "name": "",
    "url": "/talkback",
    "method": "get",
    "upload": false,
    "swaggerDoc": "",
    "x": 230,
    "y": 100,
    "wires": [
      [
        "e7d99f73.b33a9"
      ]
    ]
  },
  {
    "id": "e7d99f73.b33a9",
    "type": "template",
    "z": "a8f9e09.8028f2",
    "name": "",
    "field": "payload",
    "fieldType": "msg",
    "format": "html",
    "syntax": "mustache",
    "template": "<!DOCTYPE html>\n<html>\n<head>\n<meta name=\"viewport\" content=\"width=device-width, initial-
scale=1\">\n<style>\nbody {font-family: Arial, Helvetica, sans-serif;}\nform {border: 3px solid #f1f1f1;}\n\ninput[type=text],
input[type=password] {\n  width: 100%;\n  padding: 12px 20px;\n  margin: 8px 0;\n  display: inline-block;\n  border: 1px solid #ccc;\n  box-
sizing: border-box;\n}\n\nbutton {\n  background-color: #4CAF50;\n  color: white;\n  padding: 14px 20px;\n  margin: 8px 0;\n  border: none;\n
cursor: pointer;\n  width: 100%;\n}\n\nbutton:hover {\n  opacity: 0.8;\n}\n\n.cancelbtn {\n  width: auto;\n  padding: 10px 18px;\n  background-
color: #f44336;\n}\n\n.imgcontainer {\n  text-align: center;\n  margin: 24px 0 12px 0;\n}\n\nimg.avatar {\n  width: 40%;\n  border-radius:
50%;\n}\n\n.container {\n  padding: 16px;\n}\n\nspan.psw {\n  float: right;\n  padding-top: 16px;\n}\n\n/* Change styles for span and cancel
button on extra small screens */\n@media screen and (max-width: 300px) {\n  span.psw {\n     display: block;\n     float: none; \n  }\n
.cancelbtn {\n     width: 100%;\n  }\n}\n</style>\n</head>\n<body>\n\n<h2>Login Form</h2>\n\n<form action=\"http://127.0.0.1:1880/ui\"
method=\"get\">\n    <table>\n        <tr>\n            <td>\n                <div class=\"imgcontainer\">\n                    <img
src=\"https://ghrcem.raisoni.net/images/logo.png\" \n                         alt=\"D:\\Project\\MniModelling\\logo.jpg\" class=\"avatar\">\n
</div>\n            </td>\n        </tr>\n    </table>\n  <div class=\"container\">\n    <label for=\"uname\"><b>Username</b></label>\n    <input
type=\"text\" placeholder=\"Enter Username\" name=\"uname\" required>\n\n    <label for=\"psw\"><b>Password</b></label>\n    <input
type=\"password\" placeholder=\"Enter Password\" name=\"psw\" required>\n        \n    <button type=\"submit\">Login</button>\n    <label>\n
<input type=\"checkbox\" checked=\"checked\" name=\"remember\"> Remember me\n    </label>\n  </div>\n\n  <div class=\"container\"
style=\"background-color:#f1f1f1\">\n    <button type=\"button\" class=\"cancelbtn\">Cancel</button>\n \n\n    <button
onclick=\"myFunction()\">Forget Passwoed</button>\n\n        <p id=\"asd\"></p>\n        \n        <script>\n
document.getElementById(\"asd\").innerHTML = \"Password : Admin\";\n        } \n    </script>  \n  </div>\n</form>\n</body>\n</html>\n",
    "output": "str",
    "x": 440,
    "y": 100,
    "wires": [
      [
        "e0b7c330.fccbb8"
      ]
    ]
  },
  {
    "id": "e0b7c330.fccbb8",
    "type": "http response",
    "z": "a8f9e09.8028f2",
    "name": "",
    "statusCode": "",
    "headers": {},
```

```
    "x": 630,
    "y": 100,
    "wires": []
  },
  {
    "id": "2db582cb.42426e",
    "type": "ui_text_input",
    "z": "a8f9e09.8028f2",
    "name": "",
    "label": "",
    "tooltip": "",
    "group": "3e6cefa2.b6cf2",
    "order": 1,
    "width": 0,
    "height": 0,
    "passthru": true,
    "mode": "text",
    "delay": "0",
    "topic": "",
    "x": 240,
    "y": 360,
    "wires": [
      [
        "7501a823.757e78",
        "a31acfff.7b278"
      ]
    ]
  },
  {
    "id": "34dc87b4.6504e8",
    "type": "ui_audio",
    "z": "a8f9e09.8028f2",
    "name": "",
    "group": "3e6cefa2.b6cf2",
    "voice": "en-US",
    "always": "",
    "x": 720,
    "y": 360,
    "wires": []
  },
  {
    "id": "299468a2.272e08",
    "type": "ui_text",
    "z": "a8f9e09.8028f2",
    "group": "3e6cefa2.b6cf2",
    "order": 4,
    "width": 0,
    "height": 0,
    "name": "",
    "label": "Entered Data",
    "format": "{{msg.payload}}",
    "layout": "row-spread",
    "x": 670,
    "y": 420,
    "wires": []
  },
  {
    "id": "7501a823.757e78",
    "type": "function",
    "z": "a8f9e09.8028f2",
    "name": "speech Function/English log",
    "func": "let data = msg.payload;\nvar today = new Date();\nvar date = today.getFullYear()+'-'+(today.getMonth()+1)+'-
```

'+today.getDate();\nvar time = today.getHours() + \":\" + today.getMinutes() + \":\" + today.getSeconds();\nvar dateTime ='['+ date+' '+time +']
';\nvar log = dateTime + data\nreturn {payload : log} ;",
        "outputs": 1,
        "noerr": 0,
        "x": 500,
        "y": 420,
        "wires": [
          [
            "299468a2.272e08",
            "de27ca10.9b6608"
          ]
        ]
    },
    {
        "id": "a31acfff.7b278",
        "type": "function",
        "z": "a8f9e09.8028f2",
        "name": "speech Function/English",
        "func": "\nreturn msg;",
        "outputs": 1,
        "noerr": 0,
        "x": 490,
        "y": 360,
        "wires": [
          [
            "34dc87b4.6504e8"
          ]
        ]
    },
    {
        "id": "de27ca10.9b6608",
        "type": "file",
        "z": "a8f9e09.8028f2",
        "name": "",
        "filename": "log.txt",
        "appendNewline": true,
        "createDir": true,
        "overwriteFile": "false",
        "encoding": "none",
        "x": 690,
        "y": 460,
        "wires": [
          []
        ]
    },
    {
        "id": "b54caba6.18e1e8",
        "type": "ui_group",
        "z": "",
        "name": "Demo",
        "tab": "eb9a025f.91c9e",
        "disp": false,
        "width": "14",
        "collapse": false
    },
    {
        "id": "3e6cefa2.b6cf2",
        "type": "ui_group",
        "z": "",
        "name": "Default",
        "tab": "eb9a025f.91c9e",
        "disp": true,

```
        "width": "14",
        "collapse": false
    },
    {
        "id": "eb9a025f.91c9e",
        "type": "ui_tab",
        "z": "",
        "name": "Home",
        "icon": "dashboard",
        "disabled": false,
        "hidden": false
    }
]
```