

Conversational Voice Controlled App

A

Project Report

Submitted for the partial fulfillment

of B.Tech. Degree

in

COMPUTER SCIENCE & ENGINEERING

by

Anupriya Mishra (1805232011)

Gaurav Singh (1805210020)

Prabhav Garg(1805210033)

Under the supervision of

Prof. Manik Chandra

&

Er. Pratibha Pandey



Department of Computer Science and Engineering

Institute of Engineering and Technology

Dr. A.P.J. Abdul Kalam Technical University, Lucknow, Uttar Pradesh

Contents

DECLARATION	3
CERTIFICATE	4
ACKNOWLEDGEMENT.....	5
ABSTRACT	6
LIST OF TABLES	7
LIST OF FIGURES	8
1.INTRODUCTION	9
2.LITERATURE REVIEW	13
3. METHODOLOGY	14
3.1 PROPOSED MODELS	27
3.1.1 DESIGN STEPS	27
3.1.2 TECH STACK	28
3.2 FRAMEWORKS	28
3.2.1 NODEJS	29
3.2.2 REACTJS	29
3.3 TOOLS	29
3.3.1 ALAN AI	29
3.3.2 REST API	29
4. EXPERIMENTAL RESULTS	30
5. CONCLUSIONS	30
REFERENCES	32

Declaration

We hereby solemnly assert that this submission is a product of own work and that, to the best of our belief and knowledge, it contains no references that had previously been published or submitted by any other entity or material which to a substantial error has been accepted for the award of any degree or diploma of university or other institute of higher learning, except where the acknowledgement has been made in the text. The project has not been submitted by us at any other institute for the requirement of any other degree.

Submitted by: - Date:

(1) Name: Anupriya Mishra

Roll No.: 1805232011



Branch: Computer Science and Engineering

Signature:

(2) Name: Gaurav Singh

Roll No.: 1805210020

Branch: Computer Science and Engineering

Gausean

Signature:

(3) Name: Prabhav Garg

Roll No.: 1805210033

Branch: Computer Science and Engineering

Signature: Prabhav

Certificate

This is to certify that the project report entitled “Conversational Voice Controlled App” presented by Anupriya Mishra, Gaurav Singh and Prabhav Garg in the partial fulfillment for the award of Bachelor of Technology in Computer Science and Engineering, is a record of work carried out by them under my supervision and guidance at the Department of Computer Science and Engineering at Institute of Engineering and Technology, Lucknow.

It is also certified that this project has not been submitted at any other Institute for the award of any other degrees to the best of my knowledge.

(Prof. Manik Chandra & Er. Pratibha Pandey)
Department of Computer Science and Engineering
Institute of Engineering and Technology, Lucknow

Acknowledgement

We are highly indebted to Dr. Manik Chandra and Er. Pratibha Pandey, and we want to thank them for giving us the freedom to operate and experiment with new ideas. We want to make a move to our significant thanks to them for their educational direction and advantage in our task and constant assistance in combination with certainty boosting and driving meetings that showcased extremely productive and were indispensable in injecting confidence and trust inside us. The sustaining and blooming of the current work is primarily because of their significant direction, ideas, adroit judgment, productive analysis, and an eye for flawlessness.

Our mentor consistently addressed a horde of our questions with grinning thoughtfulness and enormous tolerance. They never make us feel like we're on our backsides by constantly listening to our perspectives, respecting and developing them, and allowing us a free hand in our project. It is simply because of their staggering interest and accommodating disposition; the current work has achieved its stage.

Finally, We are grateful to our Institution and colleagues whose constant encouragement served to renew our spirit, refocus our attention and energy, and carry out this work.

Anupriya Mishra (1805232011)
Gaurav Singh (1805210020)
Prabhav Garg (1805210033)

Abstract

“A conversational voice controlled app with authentication written in react and driven by Artificial Intelligence to provide the user with an interactive experience. The app is built in node.js runtime environment, using express for server side utilities and a NoSQL database(MongoDb) for storage and horizontal scaling.

With react we'll be able to provide a responsive user interface that would work across all devices. We'll target building an application that is easily scalable for a large user mass.”

Gaurav

Prabhav

Prabhav

List of Figures

MAIN TOOLS	11
METHODOLOGY	15
COMPLETE ARCHITECTURE	16
SCALAR WEB ARCHITECTURE	22
DATABASE SERVERS	24
LOAD BALANCERS	26
DESIGN STEPS	28
EXPERIMENTAL RESULTS	30

1. Introduction

Automatic voice managed structures have revolutionized the manner human beings engage with a laptop. Voice or speech recognition structures enable a user to make a handsfree request to the pc, which consequently evaluates the request and serves the consumer with suitable responses. After years of research and traits in device mastering and synthetic intelligence, these days voice managed technologies have become more efficient and are broadly applied in many fields to enable and enhance human to human and human to computer interactions. The stateoftheart ecommerce packages with the aid of net technologies produce interactive and consumer pleasant interfaces.

But, there are a few times where human beings, particularly with visual disabilities, are not capable of fully experiencing the serviceability of such packages. A voice controlled gadget embedded in an internet software can decorate user revel in and can provide voice as a means to govern the functionality of ecommerce web sites. In this task, we propose a taxonomy of speech popularity systems (SRS) and give a voice managed commodity buy ecommerce application the usage of Alan AI speech to textual content to illustrate its usability.

Voice reputation is used indistinguishably with speech reputation, but, voice popularity happens to be mainly the project of working out the identification of a speaker rather than the contents of the speaker's speech

Speech reputation is basically the process of getting the sound of phrases or terms spoken through human beings converted into electric alerts to which a meaning is assigned by comparing the alerts with units of phonemic representations for a close health. The phonemic representations are matched towards phrases which are predefined in a phrase vocabulary. The purpose of speech reputation is to permit humans to speak greater clearly and efficiently. This often calls for deep integration with many natural language processing (NLP) components.

Speech reputation may be applied to many domain names and packages. It could eliminate boundaries to human-human interactions by means of helping individuals who communicate one of a kind languages to be able to speak to each other without a human interpreter. Speech popularity generation has made it viable to develop computer based reading coaches that pay attention to college students, verify the performances, and provide instantaneous customized comments.

Conventional data entry methods (keyboard and mouse) do not nearly meet accessibility requirements to support all types of users. Therefore, we need to develop systems and applications that are easier for all users to use.

It is of utmost importance for a corporation or company to design and develop a web application that is not only informative but also interactive. Companies can leverage SRS to attract web audiences, advertise their products and services, educate customers about the features and usability of the products, and provide assistance in troubleshooting as are given through online chat programs which are just a few use cases to name. Today most of the existing applications lack the accessibility characteristics [4, 5], which poses a hurdle especially to the visuallyimpaired users in terms of being accessed efficiently, and use of the information and web services provided by the organizations.

The prototype can cover use cases involving conversion of text data necessitating medical prognostication at the clinics and also offer significant help in services such as kiosks which are generally provided by the government.

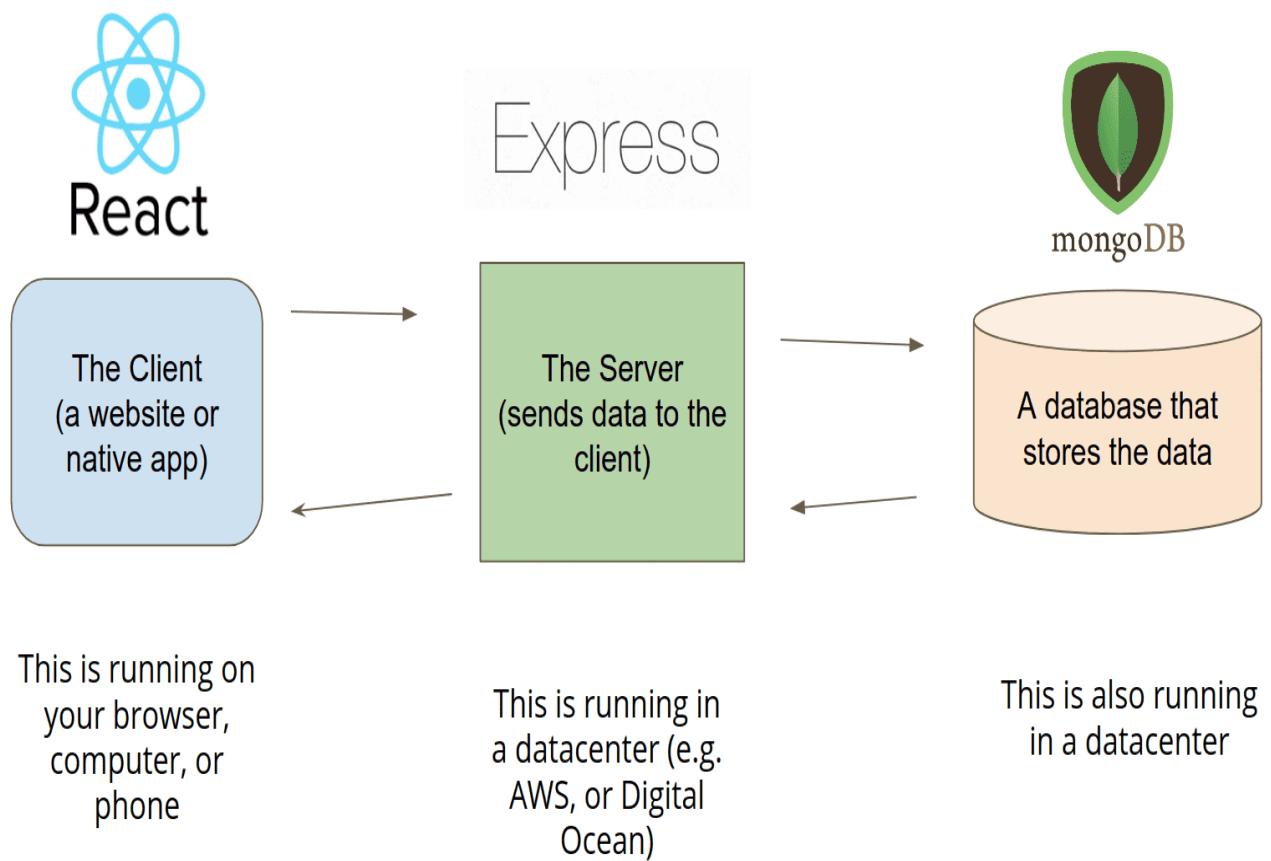
In the following project we aim to develop a complete MERN Stack News Application with REST API, Admin Panel, Mobile applications using NodeJS, Express, MongoDB, ReactJS.

The application that we target to develop, integrates REST API with the following features:

1. Authentication and Authorization with PassportJS and JWT
2. Model Validation with Express Validator
3. Data Modeling and Schema Building with Mongoose
4. Categories, Stories, CRUD Videos and Comments and Approved or Protected Path
5. Google OAuth authentication and image loading
6. Send email, paginate data, rate limit and upload images.

1.1 MAIN TOOLS:

1. AlanAI
2. JavaScript runtime(NodeJs)
3. JavaScript frameworks (React, Express)
4. NoSQL Databases (MongoDB, Redis)



Our plan is to integrate the REST API described above with a responsive **interface** and create a **fully functional** web application **that can scale** to 1 million users.

1.2 Motivation

The intersection of Artificial Intelligence and Scalable Web projects has always been a fascination for the three of us.

For the AI component we propose to implement a real-time talking tool
Live Chat is an inalienable aspect of fulfilling user/customer experience as it facilitates one's seamless connection with your websites' visitors/customers in real-time, directly on your website. These aspects can greatly impact one's economic success as it provides a unique set of features that discerns it from quintessential chat across various contacts.

Apart from this we are also focussing on scalability.

Scalability here can be conceptualized as the ability of a system to function properly in response to increasing demand (such as larger datasets, higher demand rates, size and speed combinations, etc.).

It should work adequately with 10 or a million users and automatically handle occasional traffic spikes. A scalable app only adds and removes resources as needed and consumes only the resources needed to meet demand.

Many companies usually focus on functionality over scalability. Building applications that are both resilient and scalable is an important part of any application architecture and is essential for a great user experience. This project aims to build a highly scalable web application architecture that can scale as the load increases.

2 Literature Review

1.3.1. LipSurf by LipSurf.com

Use audio to type, click, scroll, watch videos, open web pages, create custom language shortcuts, and more.

Use your voice to surf the web while eating, multitasking, or just to rest your hands.

- * Play, pause, fast forward, Youtube and other video sites etc.

- * Duolingo and WaniKani fully answer questions via voice control (hands-free)

- * Reddit voice recognition for votes, access to posts, comments and more

1.3.2. VoxNow

Website Language Recognition

recognizes over 100 languages

Voxpow supports a global user base and recognizes over 100 languages and variants. It uses a machine learning model to transform the results and return the text as soon as the user speaks.

Small JavaScript Speech Recognition Library

Friendly Support

They work closely with their users to provide a high level of support, quick error correction and troubleshooting.

3 METHODOLOGY

1.1 THE AI COMPONENT

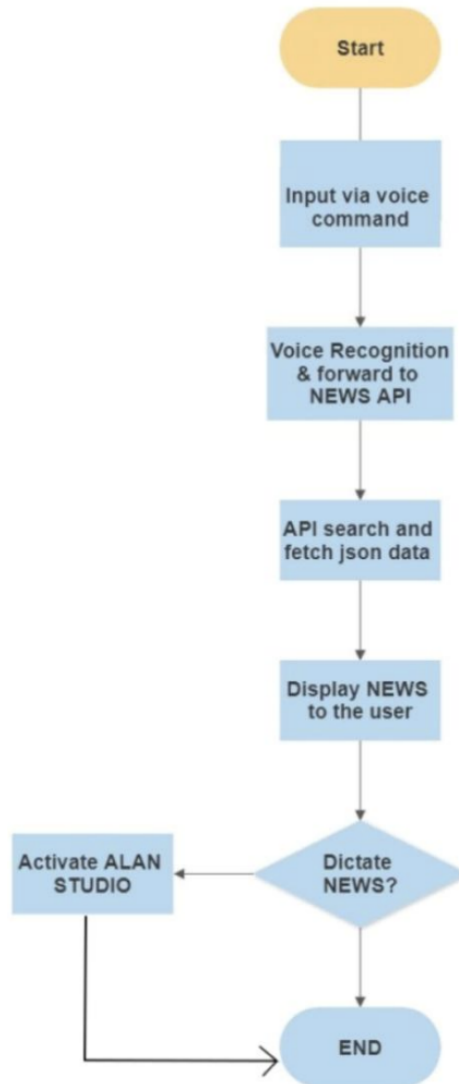
Most of the work we expect on the backend is done by AlanAI. Alan allows a single developer to create and develop a language experience for your app, rather than a team of machine learning and DevOps experts. With

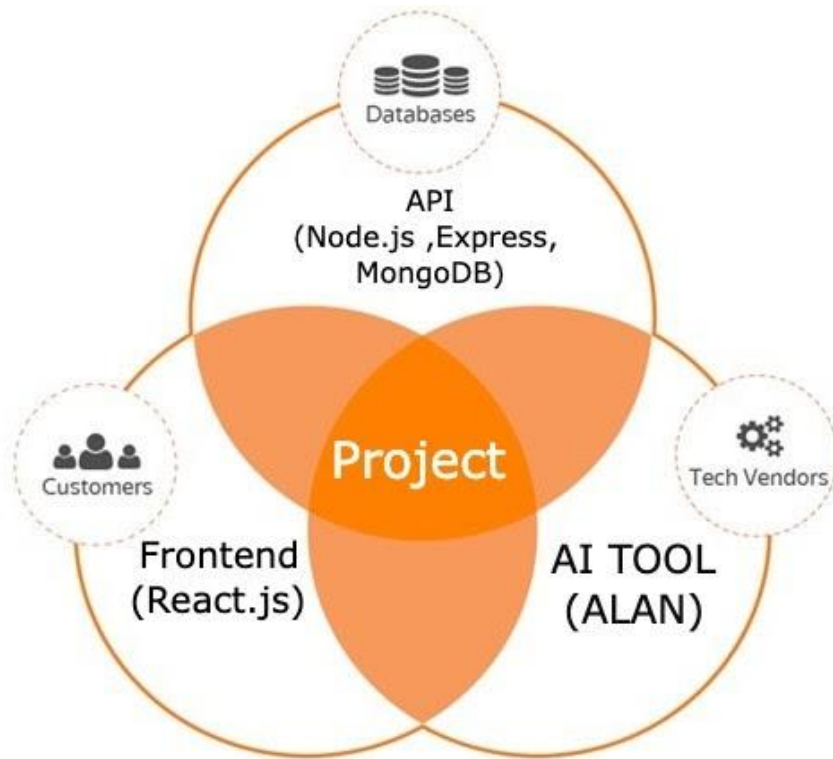
Alan, you can voice-activate complex workflows or features in your app beyond the possibilities of touch and input interfaces. Language scripts are written in JavaScript, so they are customizable and flexible. The voice interface is built once and deployed everywhere. No rebuild is required on any particular platform. Alan provides a lightweight SDK for integration with:

1. Web
2. iOS
3. Android
4. Ionic
5. Apache Cordova
6. Flutter
7. React Native
8. Microsoft Power Apps

If there's a need to change the voice interface, a new change can be submitted without releasing a new version of the application. Thanks to Alan's serverless environment, changes to the voice interface are immediately available to users. This section provides a quick overview of the basics of Alan, what is available, and how Alan's components work together. You will also get pointers to sections in this documentation that will help you decide on where to proceed.

WORKFLOW





1.2 Voice platform

Alan provides a complete AI voice platform. This means you don't have to configure voice components, deploy voice processing infrastructure, or train voice recognition software. All deployment, maintenance, and language processing tasks are offloaded to Alan, which does most of the work.

Alan acts as an AI backend. This allows the app to "understand" human voice and provide the ability to interact with the app via voice. To create a voice interface with Alan, you need to perform the following tasks:

Designing Voice Assistant Dialogs in Alan Studio Designing

Dialogs to Integrate Alan Voices into Your App Using the

Alan SDK

The first thing you need to do is create a dialog (voice script) in Alan Studio. Language scripts describe expected conversation scenarios with users. It covers all topics, questions, and phrases that users can ask or say while interacting with the voice assistant. Voice scripts are written in JavaScript, giving you unlimited flexibility in creating dialogs. Before you start writing a language script, it may be useful to sketch all possible branches that the user can traverse and prototype an alternative path that the user can follow as they move on to the next stage of the dialog. .. It's much easier to write a speech script using the "navigation flow" of such a conversation.

The core components forming the units of the vocal script are:

- Voice command
- Slot
- Context

The user interacts with the voice assistant using voice commands. You can add commands to your voice scripts to perform tasks and actions requested by users and answer user questions. Here are some examples of voice commands:

- Coffee please
- \$ 200 will be transferred to your account
- Tell us about this phone model.

Voice commands are invoked by a specific utterance by the user. In Alain, this call set is known as a pattern. When designing a speech script, you need to consider different ways users can express their requests and create a list of phrases that can be spoken.

For example, the list of phrases for the order coffee command looks like this:

- Please give me a cup of coffee
- Please give me a cup of coffee
- Can I have a cup of coffee?
- etc.

Along with the pattern, the voice command defines the action that should be triggered when this command is called. For example, you can play a response to a user or send a command to a client app to perform an app-side activity.

Alan evaluates each user's utterances and chooses the best command from all the options available in the script. This is called intent matching. If the intent matches, Alan triggers the action defined in the command.

1.3 Slots

Alan allows you to add slots to your voice commands. Slots are basically user-spoken parameters that allow Alain to identify and retrieve important information from user input. Alain can use this data to meet the user's request. For example, you can use slots to get the location of your booking app, or you can get the product categories your shopping app needs from your utterances.

You can add the following types of slots to voice commands.

User-defined slots for user-defined lists of values that the user speaks when issuing a voice command or that can be dynamically loaded from an external source.

1.4 Context

As with real conversations, some user commands in voice scripts can only make sense in context. For example, when working with the voice assistant for weather forecasts, the user can ask: How about tomorrow? To answer this question best, Alain needs to understand the situation the user is referring to, that is, where the user is asking.

Contexts can be used when creating audio scripts for multi-step conversations where some conversation steps can only be performed in certain situations. In this case, when the user says a particular phrase or reaches a particular point in the dialog, you can place a portion of the user's dialog in a context and move to that context to activate it. For example, if you are designing a voice interface for an online store, you can create a context that retrieves address and time details and activate that context only after the checkout process has started.

Built-in JavaScript library You can use the following JavaScript libraries when creating

1. language scripts. Axios for making API calls from
2. language scripts and Moment Timezone and Luxon running at Request
3. Lodash time

1.5 Data sharing

When you create a voice assistant for your app, you may need to exchange data between your voice script and your app, or within the script itself. Alan provides several methods for doing this. You can use the project API and visual state objects to pass data from your app to your language script. You can use the userData and state objects to exchange data between the contexts of language scripts.

For example, if your project has multiple scripts and you want all scripts to have some data available, you can use a project object.

1.6 Integration

Once the voice script is ready, the next step is to integrate Alan voice to your app. Once integrated, you'll see Alan's draggable buttons at the top of the app's interface. Users can interact with the application by clicking this button and providing commands in a script developed by the developer.

For integration details, check out the Integrations section of Alan Studio, the Alan Client SDK documentation, or the Alan sample app on GitHub.

Alan provides generic client API methods for Alan buttons. You can use these methods on the client application side. For example, you can play a greeting when the user starts a chat session and enable or disable Alan's button programmatically as needed.

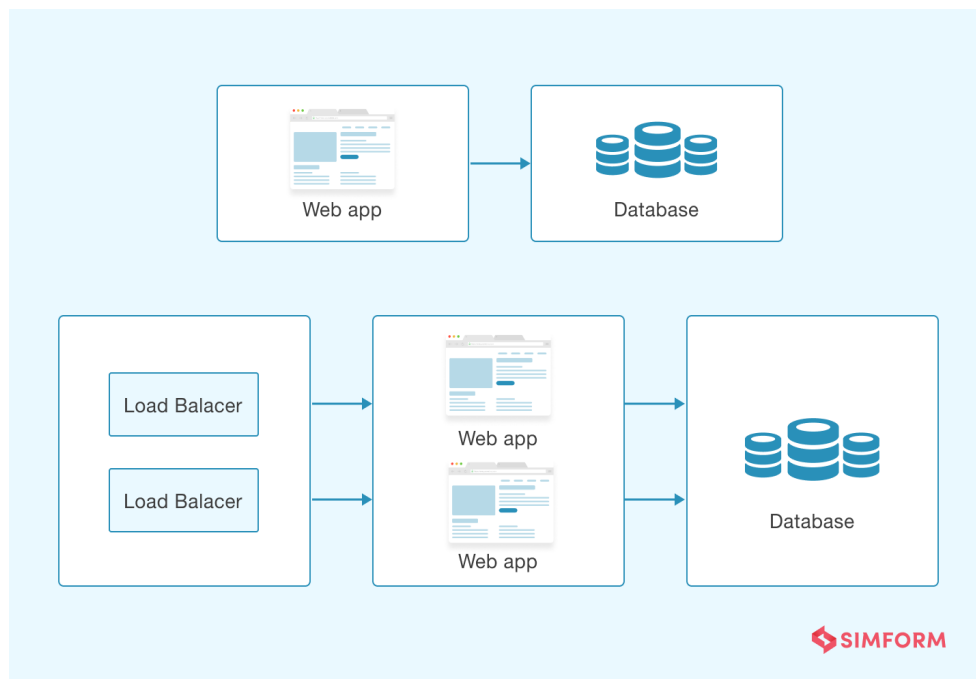
2.1 Scalable Components

Scalability A scalable web architecture like LAMP can provide horizontal scalability. Distributed systems can spontaneously allocate and de allocate resources as and when there is a need.

High availability High availability and reduced downtime are important to any business. Especially for e-commerce businesses, an hour of downtime can lead to significant loss of revenue. A scalable web architecture needs to ensure higher availability by providing critical component redundancy and rapid disaster recovery in the event of a partial system failure.

Fault Tolerant There should be no single point of failure and the system should be able to run efficiently in the event of a component failure.

1. **Share Nothing**-This is an architectural pattern in which each component is independent and can perform a particular function. This can be achieved with various web application tiers, such as sharding or the database tier via sharding.
2. For cache layers. Client-side memcache partitioning. Some of the application data is cached and the rest is managed by a relational database management system (RDBMS).



A typical scalable web architecture will have four key layers,

1. Web servers
2. Database servers
3. Load balancers
4. Shared file Servers

Each of these tiers scale independently, and the database tier is the most difficult to scale. Master slave replication is a great methodology to scale database effectively. Master nodes in this scenario are powerful machines capable of reading and writing data, while slave nodes can only read data. Consequently, load balancer is tasked with ensuring load distribution across master nodes. A layer of caching ensures optimal performance.

In this scenario, the common tasks such as bootstrapping can be adequately addressed, while cache at the application level could resolve large requests.

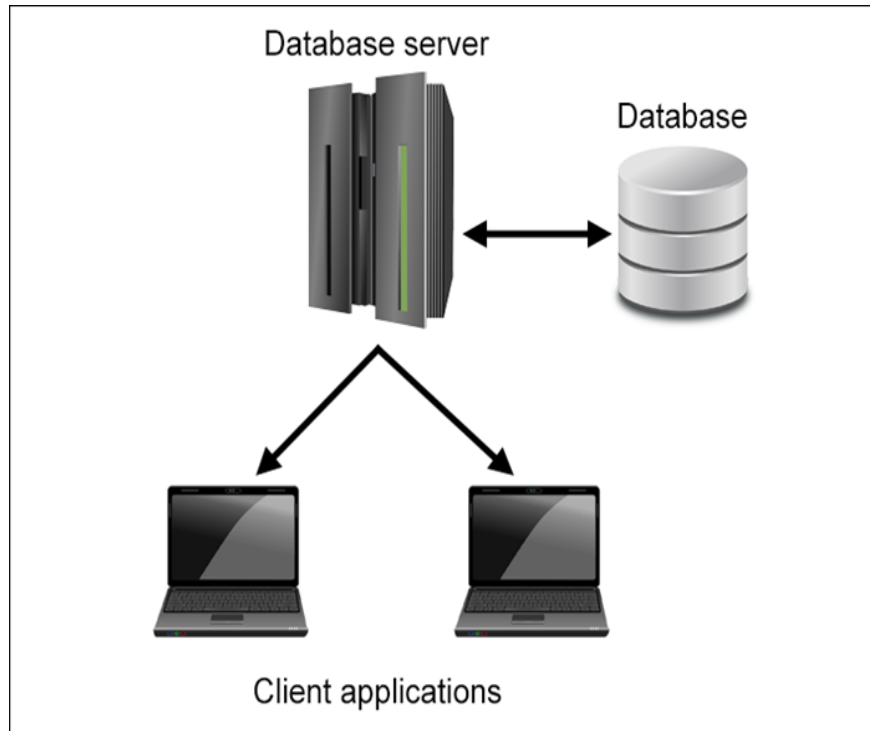
2.2 Web Servers

2.2.1 Definition: A web server could be conceptualized as a machine that is running a website. The programs running at that machine service the web pages as and when required to the. The main task of a web server is to store, process, and deliver web pages to users. This relationship is done using Hypertext Transfer Protocol (HTTP). Most of these web pages are static content, including HTML documents, images, style sheets, tests, and more. In addition to HTTP, the web server also supports Simple Mail Transfer Protocol (SMTP) and File Transfer Protocol (FTP) for sending email and transferring and storing files.

2.2.2 Description: The main task of a web server is to display the content of a website. When a web server is not public and is used internally, it is called an intranet server. When someone requests a website by adding a URL or web address to the address bar of a web browser (Chrome, Firefox, etc.) (eg www.economictimes.com), the browser sends a request to the Internet to view the web. page about it. address. The Domain Name Server (DNS) translates this URL into an IP address that points to the web server (for example, 192.168.216.345). The web server receives a request to display website content in the user's browser. Every website on the Internet has a unique ID in terms of an IP address. This Internet Protocol address is used to communicate between different servers over the Internet. Apache server is the most used web server on the market today. Apache is open source software that powers nearly 70% of all websites available today.

2.3 Databases Servers

The database server is used to store and manage the databases stored on the server and to provide data access to authorized users. This type of server stores data in a central location that can be backed up on a regular basis. In addition, users and applications can centrally access data over the network. Many databases used by your organization can be kept in a server or group of servers specially configured to protect your data and service client needs.



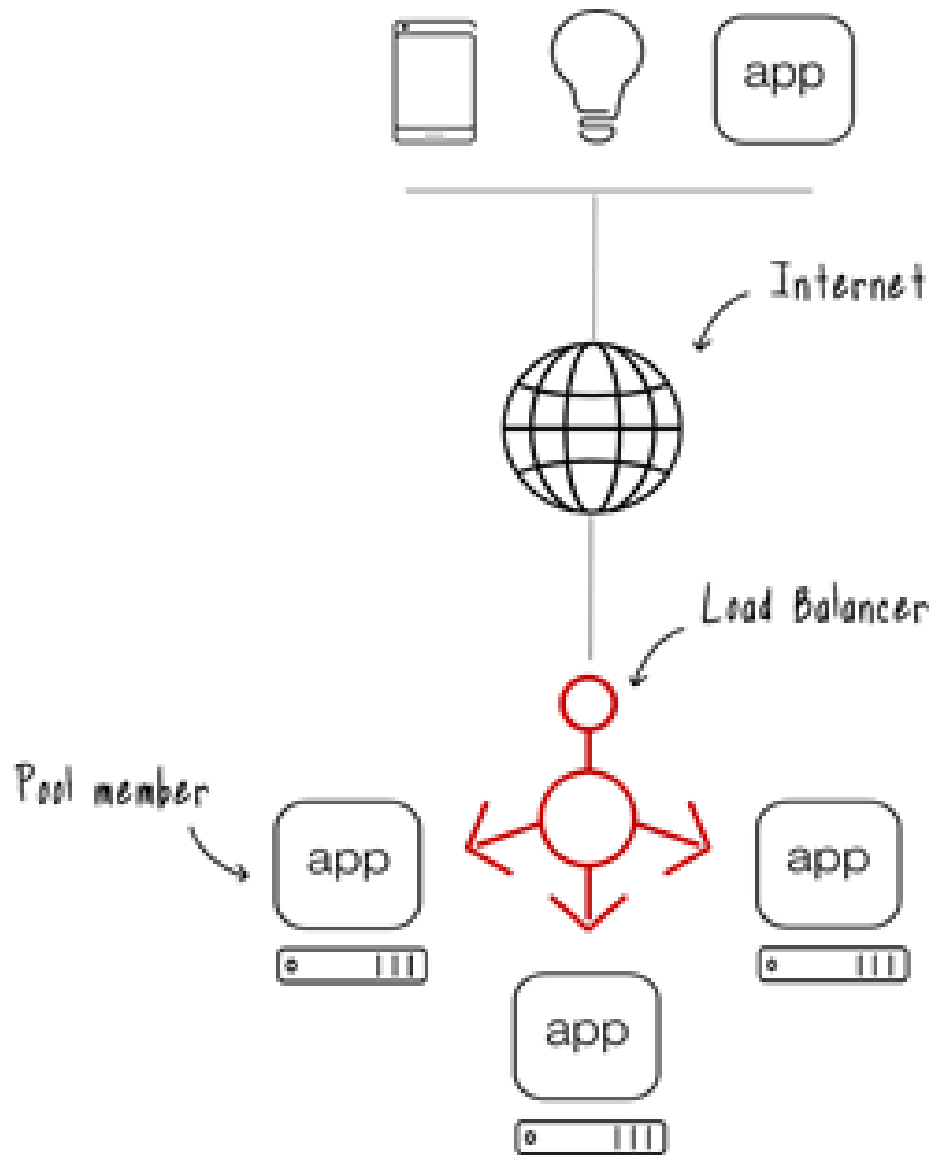
2.4 Load Balancers

A load balancer is a device whose role could be thought of as being the same as that of a reverse proxy and distributes network or application traffic across **multiple** servers. Load balancers main objective is to increase the capacity (concurrent users) and reliability of your application. They offload servers related to maintaining applications

and network sessions as well as managing them and improve the performance of applications in a general manner by performing tasks specific to application.

The requests are received by both types of load balancers and distributed to specific servers based on the algorithm that the load balancer is configured with. Standard Algorithms used across industry are :

- Round robin
- Weighted round robin
- Least connections
- Least response time



Load Balancer Diagram

- Load balancers ensure reliable and available systems by monitoring the "health" of applications and only sending requests to servers and applications that can respond in a timely manner.

2.5 Shared file Servers

2.5.1 What is a file server?

A file server could be conceptualized as a computer tasked with ensuring the storage of data files and the management of those files. This is done so that the machines present on the same network can have restricted or unrestricted access to the files.

File servers are a common target for hackers and ransomware, so **you need to be extra careful to protect them from** attacks.

2.5.2 How do file servers work?

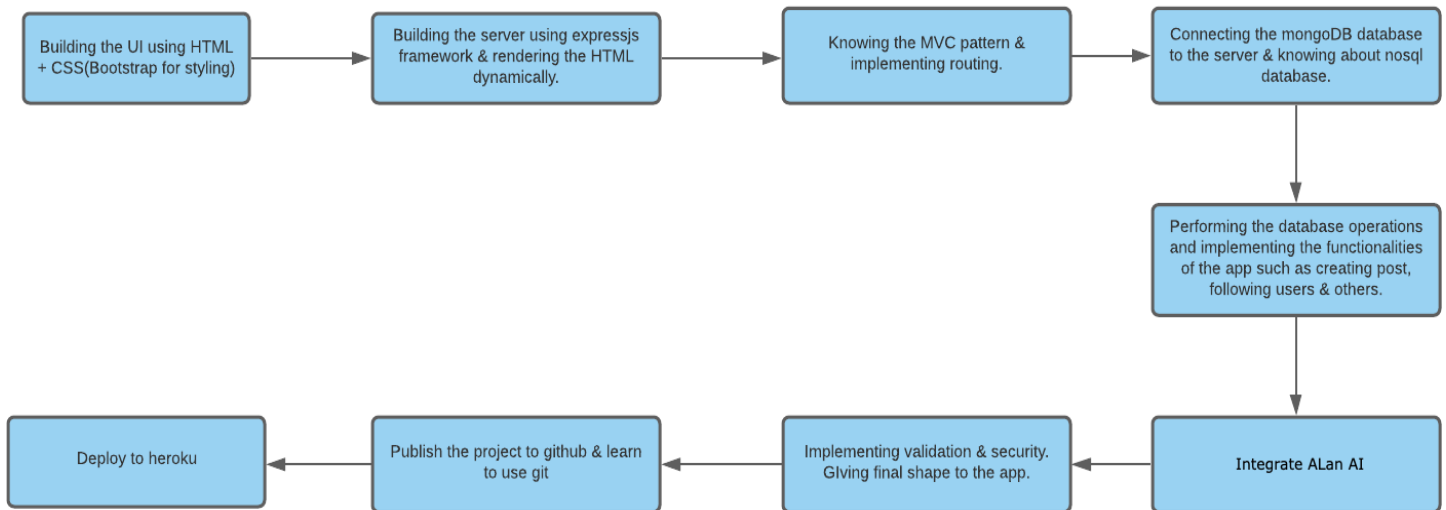
The file server allows clients to use remote file systems.

Stores data, typically in large binary data or file formats. This means that they do not perform any further indexing or processing of the files stored in them. However, there may be additional plugins or server-side features that may provide additional functionality.

A database is not considered a file server because it only handles structured data accessed through queries.

3.1 Proposed Model

3.1.1 DESIGN STEPS



3.1.2 TECH STACK

1. NodeJS

Node.js, an asynchronous event-driven JavaScript runtime. A callback is triggered for each connection, but Node.js sleeps if there is nothing to do.

2. ReactJS

React, made by facebook, makes it painless to create interactive UIs designs. Design simplified views for each state in the website, React will efficiently update and render just the correct components when your data changes.

3. MongoDB

MongoDB is an NoSQL open source database management program. NoSQL is used as an alternative to traditional relational databases. NoSQL databases are quite useful for working with large sets of distributed data. MongoDB is a tool that can manage document-oriented information, store or retrieve information.

4. Redis

Redis is a in-memory data structur, used as a disributed, in-memory key–value database, cache and message broker, with optinal durability. Redis supports different kinds of abstract data structures, such as strings, lists, maps, sets, sorted sets, HyperLogogs, bitmaps, streams, and spatial indices.

5. ExpressJS

Express.js, where js stands for java script, is simply Express, a back end web application framework for Node.js, released as free MIT License, open-source software. It is designed for building web applications and APIs. It has been called the de facto standard server framework for Node.js.

6. AlanAI

Alan is a voice AI platform that offers you to add an audio interface to your app without overhead. Alan is an end-to-end conversational AI platform for building robust and reliable in-app voice assistants and chatbot.

We don't need to create speech language models, train audio recognition software, deploy and host speech components. Most of the development is done by the AlanAI backend.

Alan allows a single developer to create and develop a language experience for your app, rather than a team of machine learning and DevOps experts. With Alan, you can voice-activate complex workflows or features in your app beyond the possibilities of touch and input interfaces. Language scripts are written in JavaScript, so they are highly customizable and flexible.

7. REST API

Representational State Transfer is a software arch style created to analyze and map the design and development of the scalable website architecture. REST API defines a set of requirements on the architecture of distributed media systems on the Internet. The website needs to work.

4. Experimental Results

The proposed project provides a new, faster, more reliable, easier-to-use, and user-friendly experience that helps users connect to and keep up to date with events around the world. The system can distinguish between the various keywords that the user said, so it can return the message according to the query provided by the user.

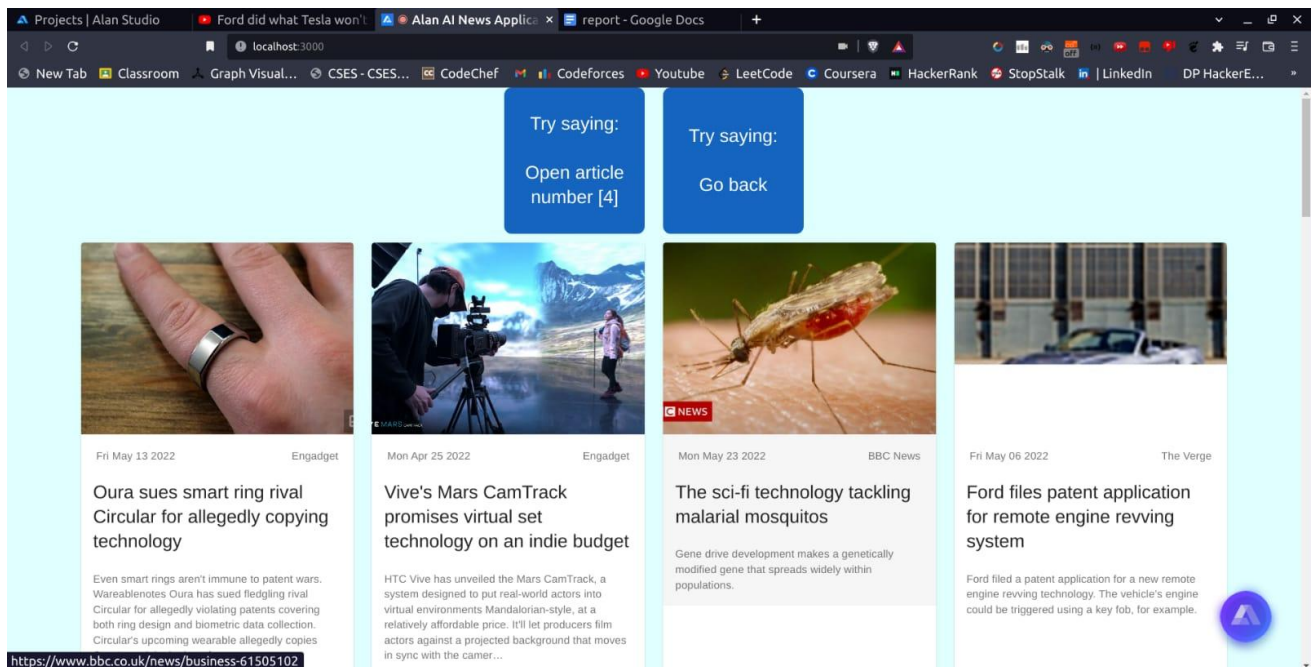
The system distinguishes between different sources, locations, topics, and topics of news that users may be interested in. The above data shows how the system behaves when

search queries of various lengths are entered into the system. The system identifies the set of keywords and displays the results according to the identified keywords. As the number of words increases, so does the number of keywords, and news articles are selected according to the keywords.

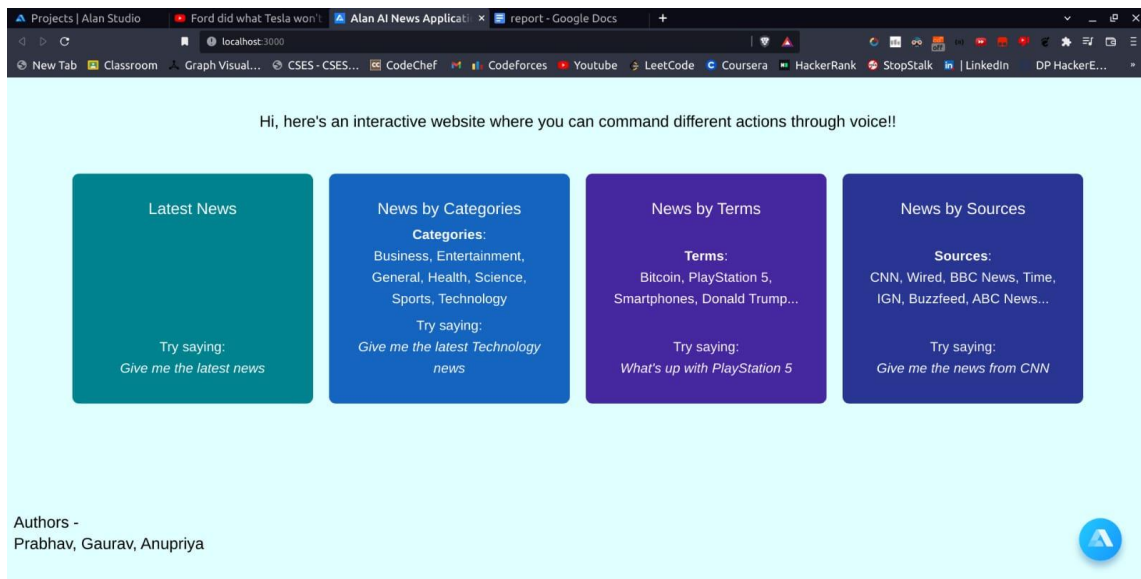
Number of words searched	Number of keywords identified	Related articles fetched	Unrelated articles fetched
1	1	20	0
2	1	20	0
3	1	20	0
4	1	19	1
5	2	18	2
6	2	18	2
7	3	16	4
8	3	16	4
9	3	14	6
10	4	14	6

Thus, a set of news articles that may not be related to the general search term, but which are related to the specific keyword identified by the system, are retrieved. However, the system retrieves more relevant articles compared to the number of unrelated articles retrieved.

Screenshots(Live website)



Screenshot 1



Screenshot 2

5 Conclusions

We would be able to get hands-on experience on frontend development in React and backend technologies like nodejs along with interaction with databases like MongoDB.

It would also strengthen our concepts and we would get a more solid understanding of how cloud computing is instantiated in modern world applications and the level of sophistication involved in making these technologies communicate with each other seamlessly.

This would also give us a first hand experience about AI, its numerous applications and how those technologies could be integrated into an application that serves multiple purposes without us having to reinvent the wheel. We would also get familiarized with the best practices used around Software Development and the rationale behind them, as we try to think our way through the problems that we would face.

5.2.1 Future Works

When speech recognition began to appear in the early 2010s, when Siri, one of the most recognizable devices on the current, was introduced, no one expected that this would be the driving force behind technological innovation in the future. It is estimated that 1/6 of the current US population owns a smart speaker. It gives you an idea of the capabilities of this technology and the vast capabilities of the system.

One of the biggest reviews from speech recognition in 2020 is that people's behavior will change when they search for it. The brand is now experiencing a shift from touchpoints to listening points. No Visual interface with voice assistant. System provides both visual and voice interaction with the user. In this way, users can more actively interact with the systems and technologies as they get all the user-friendly experiences where they can see and hear the content they have requested on the.

The technology is advancing worldwide, and it is expected that the next few years will be the year when speech recognition will grow and generate huge revenues. The system is currently working on most requirements, so features are added to the system. The system will support more languages supported by the system, adding much more comfort and simplicity to users. The CSE 2019 proves that the audio and video displays have been combined into one. The need to combine the visual and audio capabilities has been around for a long time in the market, and the system is a stepping stone toward the by leveraging the audio and display capabilities in a unique and interactive way and path. The system will also become increasingly user friendly, monitoring the most popular news sources and news categories .

References

- [1] <https://nodejs.org/en/docs/>
- [2] <https://expressjs.com/en/advanced/developing-template-engines.html>
- [3] <https://www.algoexpert.io/systems/product>
- [4] <https://alan.app/docs/>
- [5] <https://www.simform.com/blog/building-scalable-application-aws-platform/>
- [6] https://www.w3schools.com/nodejs/nodejs_mongodb.asp
- [7] <https://www.w3schools.com/js/default.asp>
- [8] <https://www.sitepoint.com/using-redis-node-js/>
- [9] Advantages of Voice Recognition Technology Retrieved from <https://www.insidetelecom.com/advantages-anddrawbacks-of-voice-recognition-technology/>
- [10] reading news is beneficial Retrieved from https

- Why: <http://www.paperboy.com/blog/top-5-benefitsimportance-of-reading-news/>
- [11] Alan Studio: <https://alan.app/>
 - [12] React.: <https://reactjs.org/>
 - [13] NEWS API : <https://newsapi.org/docs>
 - [14] Sonali Sen, Shamik Chakrabarty, Raghav Toshniwal, Ankita Bhaumik. Design of an Intelligent Voice Controlled Home Automation System. International Journal of Computer Applications 121(15):39-42, 2015.
 - [15] David F. McCall, Leslie M. Logue, Francis J. Zelina, Matthew V. Sendak, Julie R. Hinson, Ward L. Sanders, Steve Belinski, Brian E. Holtz. Voice controlled surgical suite. Google Patents US6591239B1, 1999-Present.
 - [16] Ian W. Freed, William Folwell Barton, Rohit Prasad. Preventing false wake word detections with a voicecontrolled device. Google Patents US9368105B1, 2014.
 - [17] Stephen S. Burns, Mickey W. Kowitz. Voice controlled wireless communication device system. Google Patents US7957975B2, 2016-Present.
 - [18] K. Christian, B. Kules, B. Shneiderman and A. Youssef, "A Comparison of Voice Controlled and Mouse Controlled Web Browsing," Proceedings of the Fourth International ACM Conference on Assistive Technologies, no. ACM, pp. 72-79, 2000.
 - [19] . B. BAJPEI, M. S. SHAIKH and N. S. RATATE, "VOICE OPERATED WEB BROWSER," International Journal of Soft Computing and Artificial Intelligence, vol. 3, no. 1, pp. 30-32, May-2015.
 - [20] S. Han, G. Jung, B.U. C. Minsoo Ryu and J. Cha, "A Voice-controlled Web Browser to Navigate Hierarchical Hidden Menus of Web Pages in a Smart-tv Environment," Proceedings of the 23rd International Conference on World Wide Web, pp. 587-590, 2014.
 - [21] S. Sagar, V. Awasthi, S. Rastogi, T. Garg, S. Kuzhalvaimozhi, "Web application for voice operated email exchange",
 - [22]<https://clearbridgemobile.com/7-key-predictions-for-thefuture-of-voice-assistants-and-ai/>
 - [23] <https://trends.google.com/trends/explore?cat=16&q=%2Fm%2F052j8s>