iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 11 **Issue:** IV **Month of publication:** April 2023

DOI: https://doi.org/10.22214/ijraset.2023.50507

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue IV Apr 2023- Available at www.ijraset.com

Cloudbook Website

Aanchal Padmavat[1], Gaurav Agarwal[2], Nandini Agrawal[3], Madhura Dhongade[4]
[1, 2, 3, 4]*Department of Computer Science, Shrimati Kashibai Navale College of Engineering, Pune - 411041*

Abstract: ***CloudBook is a notes-taking platform that allows users to create and organize notes in separate files with headings, subheadings, and content. The platform automatically updates the changes made when users save and sign out. While phone notepads are commonly used to write down grocery lists, to-do lists, or notes from presentations, these notes often get lost or become disorganized.***
Additionally, transferring them to a computer can be challenging. To address this issue, CloudBook was created to provide an organized and accessible format for note-taking.
The platform includes features such as checklists, emoticons, and the ability to save external links for further reading. Users can access their notes from any device by logging into their accounts.
Keywords: MERN, Notes, Cloud-Based, Flexibility, Organize, Website, Management

I. INTRODUCTION

CloudBook is a website that allows you to create notes with headings, subheadings, and content. Any changes made are automatically saved to your account when you sign in and out. While many of us use phone notepads to jot down grocery lists or work-related notes, these can easily get lost or become disorganized. Additionally, transferring this information to a laptop or desktop for official use can be difficult. CloudBook provides a solution to these issues.

II. PROPOSED USAGE

1) *Personal Organization:* Individuals can use cloud-based note-taking websites to keep track of personal to-do lists, appointments, and important information.
2) *Remote Work:* Remote workers can use cloud-based note-taking websites to collaborate with team members and keep track of project information.
3) *Student Notes:* Students can use cloud-based note-taking websites to take notes during lectures, organize their study materials, and collaborate with classmates on group projects.
4) *Meeting Notes:* Teams can use cloud-based note-taking websites to take notes during meetings and share them with all members.
5) *Project Management:* Project managers can use cloud-based note-taking websites to keep track of project information, assign tasks, and collaborate with team members.
6) *Research:* Researchers can use cloud-based note-taking websites to organize their findings, keep track of sources, and collaborate with other researchers.
7) *Travel Planning:* Travellers can use cloud-based note-taking websites to keep track of travel itineraries, make packing lists, and organize information about destinations.

III. PROPOSED OBSERVATIONS

A. *Parameters*

1) *Scalability:* The ability of the database model to handle increasing amounts of data and users over time. This includes query response time, data storage capacity, and concurrent user capacity.
2) *Performance:* The speed and efficiency of database operations, such as data retrieval and modification. This includes metrics such as query execution time and data retrieval speed.
3) *Data Consistency:* The ability of the database model to maintain accurate and consistent data, even in the presence of failures or concurrent access. This includes factors such as transaction isolation, data integrity, and consistency checks.
4) *Security:* The ability of the database model to protect sensitive information and prevent unauthorized access. This includes factors such as encryption, access controls, and data audit trails.

©IJRASET: All Rights are Reserved

5) *Availability:* The ability of the database model to remain operational and accessible, even in the presence of failures or maintenance operations. This includes factors such as failover capabilities, load balancing, and backup and recovery mechanisms.
6) *Flexibility:* The ability of the database model to adapt to changing requirements and accommodate new data structures and types. This includes factors such as the ability to add or modify columns, tables, and data types, and the ability to integrate with other systems and technologies.
7) *Cost:* The financial cost of implementing and maintaining the database model, including hardware, software, and licensing costs, as well as staffing and maintenance costs.

B. *Findings*

1) Why use a non-relational database like MongoDB instead of a simpler relational database like MySQL?

To a certain extent of the number of records (mostly less), both MySQL and MongoDB function with similar efficiency. But MongoDB scores over MySQL when the website/application is scaled and thus must store more entries.

2) What are the benefits of Rich Internet Applications?

RIA technologies tap into numerous advantages such as boosting the usability of a function-rich interface enabling users to model several queries of different formats including but not limited to keyword, audio similarity, image similarity, and video similarity, to improve queries, to receive suggested queries and content, and much more.

3) How did choosing the MERN stack as the underlying architecture of CloudBook help in creating a responsive site?

MERN stack is a robust confluence of 4 modern-day technologies- MongoDB (NoSQL DB), ExpressJS(Web Application Framework), NodeJS(Server Environment), and ReactJS(Framework for developing frontend). Benefits of each of them are:

a) *Mongo DB:* Does not need a scheme, Suitable for large scale applications.
b) *ExpressJS:* Asynchronous, Speedy, Not limited to single page web apps, Robust, Pluggable .
c) *ReactJS:* Enhances user experience by improving speed and performance of applications, Availability of nested components.
d) *NodeJS:* Cross-platform, Offers Node Package manager (NPM), Asynchronous.

4) Why use ReactJS for the front end?

ReactJS along with the Redux framework can help in achieving quality requirements such as uncluttered UI, responsiveness to both mobile and web browsers, no shuttering while scrolling, and many more.

C. *Challenges*

Not enough server power is there for extraordinary performance and response time. Many bugs and errors occurred to us while programming the code as the project is innovative and many new features are being added. Exception handling for instances like the wrong API calls/ No data sent to API was hard to implement.

IV. LITERATURE SURVEY

SR.NO	AUTHOR	TITLE	DATE	GAP	FINDINGS
1	-Hanen Abbes -Faiez Gargouri	Big Data Integration: a MongoDB Database and Modular Ontologies based Approach	5 Sept 2016	Using a NoSQL database to store data	How to use NoSQL database (MongoDB) to store data
2	-Piero Fraternali -Sara Comai -Alessandro Bozzon -Giovanni Toffetti	Engineering Rich Internet Applications with a Model-Driven Approach	April 2010	How to design platforms that improve user experience	RIA technologies are employed to optimize the usability of a function-rich interface.
3	Sanchit Aggarwal	Modern Web-Development using ReactJS	March 2018	User Friendly and interactive frontend	ReactJS renders reusable UI components using features like lightweight DOM, JSX

4	-Anurag Kumar -Ravi Kumar Singh	Comparative analysis of AngularJS and ReactJS	12 Apr 2021	Dilemma of choosing a javascript framework	Strengths and weaknesses of both the frameworks
5	-Dipika Damodaran B -Shirin Salim -Surekha Mariam Vargese	Performance Evaluation of MySQL and MongoDB databases	April 2016	How to compare relational and non-relational databases	For lesser records, both the types work similarly. But as the number of records increase, MongoDB takes less time for execution than MySQL
6	Tung Khuat	Developing a Frontend application using ReactJS and Redux	June 2018	How to build a super interactive frontend application	ReactJS + Redux framework can help in achieving quality requirements like uncluttered UI & no shuttering while
7	-Igibek Koishybayev -Alexandros Kapravelos	Reducing the attack surface of Node.js applications	2020	How to address threats faced by node.js applications.	How Minimode measures and removes unused code and dependencies.
8	Anjali Chauhan	A review on various aspects of MongoDB databases	5 May 2019	Effective usage of MongoDB database	Usage scenarios of MongoDB like Content Management and Delivery, User Data Management, Mobile and Social Infrastructure
9	-Anuja Pande -Prof. A.B. Deshmukh -Prof. M.D. Tambakhe	E-payment Gateway Model	May 2020	How to ensure that transactions taking place are genuine	Transaction happens only after authorisation of the user and is thus secured.
10	-Md Arif Hassan -Zarina Shukur -Mohammad Kamrul Hasan	An Efficient Electronic Payment System for E-Commerce	27 Aug 2020	Verifying genuine transactions	This model successfully checks all parameters of security evaluation.

Fig. 1 Literature Survey

V. PROPOSED MODEL

A. Working

Users of the system should be able to retrieve notes based on the authentication token according to the user name and password. The main agenda of the project is to make sure that there is no conflict or data sharing and each user get to see data of their own. Users can also have access to the online extension and the Mobile app, which can be handy and can be used to access all the data on the go. Customers will have access to customer functions, and the employees will have access to both customer and notes management functions. The customer should be able to do the following functions:

1) Make a new note
2) Delete a note
3) Update a note
4) Access the Mobile Application
5) Access the Web application
6) Access the Cloud Storage.

Whenever you need to take notes and write down important ideas or information, you will find this website very helpful. You can use it as a note-taking, organizational, and brainstorming tool.

Each note you create can be stored on a virtual page of the website server. You can store an unlimited number of separate notes. Each can be as long as you desire and have a separate title associated. The CloudBook website works seamlessly with cloud to automatically back up and sync your notes if the computer is connected to the Internet.

It can also type, write, and draw notes on the device of choice just as one would do using pen and paper. You can add files, multimedia, and live recordings to your notes to enrich the meaning and context. You can also collaborate and share notes with others instantaneously and in real-time. It is a user friendly website and is designed especially for personal use. As we see around us, most of the software is designed for professional use and often personal needs are ignored, CloudBook is exactly for that purpose.

We have used the waterfall model of the Software Development Life Cycle models due to the following reasons:

- Requirements are clear and not changing frequently
- Application is not complicated and big
- Environment is stable
- Technology and tools used are not dynamic and is stable

Below diagram represents how once a customer initiates a request for an account or note, the below dependencies are implemented.

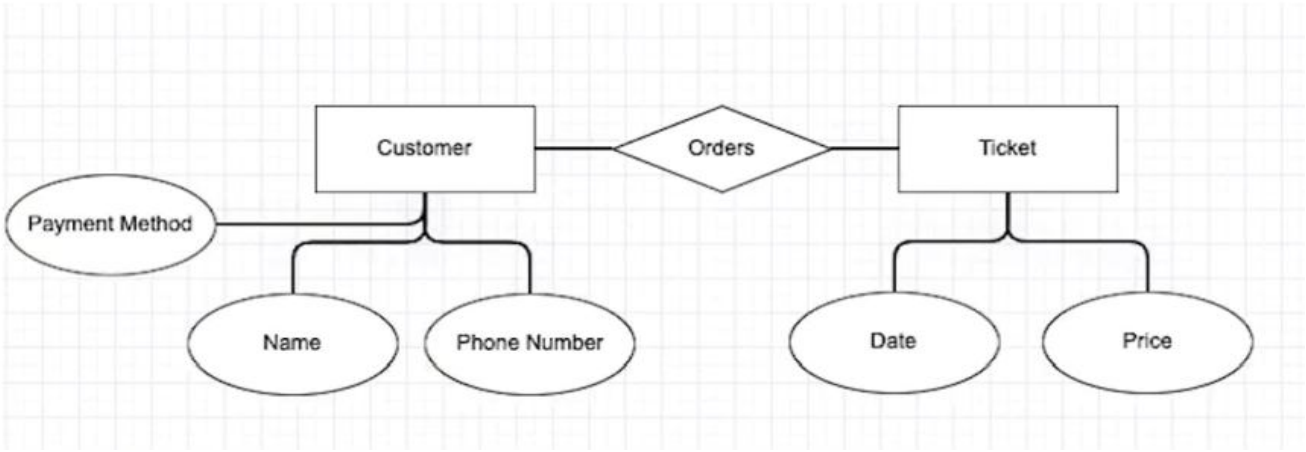


Fig. 2 Dependencies between Customer and account

Along with that, other database parameters are attached to it as shown in the figure below:

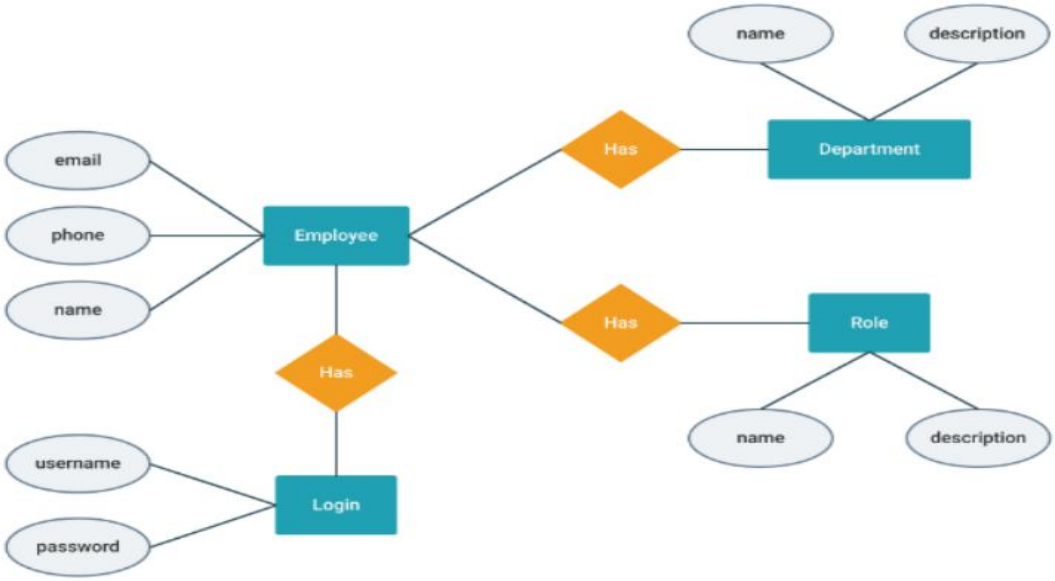


Fig. 3 ER Diagram

Once we got familiar with the database, below is a detailed representation of how the technology is used in CloudBook.

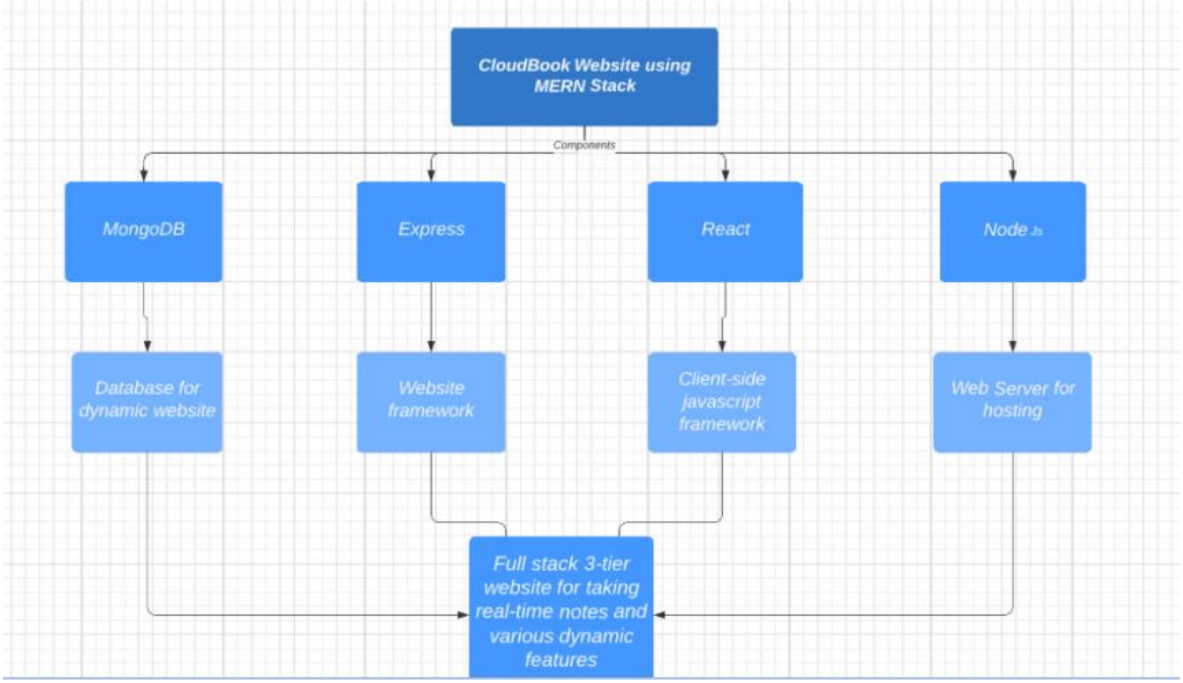


Fig. 3 Full Stack 3 tier Structure

VI. SOFTWARE REQUIREMENT SPECIFICATION

A. Project Scope

The purpose of Cloudbook is to ease taking notes and creating to-do lists and to create a convenient and easy-to-use application for users, trying to keep notes. The system is based on a relational database. We will have a database server supporting hundreds of users. Above all, we hope to provide a comfortable user experience along with the best pricing available.

B. Assumptions and Dependencies

Let us assume that this is a distributed notes management system and it is used in the following application:
A request for creation/updating of a note from any source to any operating system, giving connected records in case there is no direct access between the specified Source-database pair.
The main aim of the project is to save notes on the Mongo DB Database assuming that each user has their own account and the data will be stored according to the authenticated token respectively.
The Mobile Application and the Chrome Web Application can be accessed after the user decides to purchase a premium service from the website using the payment gateway with various types of payment modes available.

C. Functional Requirements

1) Distributed Database

Distributed database implies that a single application should be able to operate transparently on data that is spread across a variety of different databases and connected by a communication network as shown in below figure.

2) Client/Server System

The term client/server refers primarily to an architecture or logical division of responsibilities, the client is the application (also known as the front-end), and the server is the DBMS (also known as the back-end).
A client/server system is a distributed system in which,

- Some sites are client sites and others are server sites.
- All the data resides at the server sites.
- All applications execute at the client sites.

D. External Interface Requirements

1) User Interfaces

- Front-end software: React version
- Back-end software: Mongo DB

2) Hardware Interfaces

- Windows.
- A browser which supports CGI, HTML & Javascript.

3) Communication Interfaces

This project supports all types of web browsers like Google Chrome, Microsoft Edge, Mozilla Firefox,etc. We are using simple electronic forms for the Login forms, notes updating, etc.

4) Software Interfaces

Following are the software used for the Notes management online application.

Software used	Description
Operating system	We have chosen Windows operating system for its best support and user-friendliness.
Database	To save the notes records, users record we have chosen Mongo DB database.
React	To implement the project, we have chosen React language for its more interactive support.

E. Non-Functional Requirements

1) Performance Requirements

The steps involved to perform the implementation of notes management database are as listed below.

a) E-R Diagram

The E-R Diagram constitutes a technique for representing the logical structure of a database in a pictorial manner. This analysis is then used to organize data as a relation, normalizing relation and finally obtaining a relation database.

- *Entities:* Which specify distinct real-world items in an application.
- *Properties/Attributes:* Which specify properties of an entity and relationships.
- *Relationships:* Which connect entities and represent meaningful dependencies between them.

b) Normalization

The basic objective of normalization is to reduce redundancy which means that information is to be stored only once. Storing information several times leads to wastage of storage space and increase in the total size of the data stored.

If a database is not properly designed it can give rise to modification anomalies. Modification anomalies arise when data is added to, changed or deleted from a database table. Similarly, in traditional databases as well as improperly designed relational databases, data redundancy can be a problem. These can be eliminated by normalizing a database.

Normalization is the process of breaking down a table into smaller tables. So that each table deals with a single theme. There are three different kinds of modifications of anomalies and formulated the first, second and third normal forms (3NF) is considered sufficient for most practical purposes. It should be considered only after a thorough analysis and complete understanding of its implications.

2) Safety Requirements

If there is extensive damage to a wide portion of the database due to catastrophic failure, such as a disk crash, the recovery method restores a past copy of the database that was backed up to archival storage (typically tape) and reconstructs a more current state by reapplying or redoing the operations of committed transactions from the back up log, up to the time of failure.

3) Security Requirements

Security systems need database storage just like many other applications. However, the special requirements of the security market mean that vendors must choose their database partner carefully.

4) Software Quality Attributes

- *Availability:* The note should be available on the specified date and specified time as many customers are doing advance updating.
- *Correctness:* The note should reach start from correct start terminal and should reach the correct destination.
- *Maintainability:* The administrators and note in chargers should maintain correct schedules of notes.
- *Usability:* The note schedules should satisfy a maximum number of customer's needs.

F. System Requirements –

1) Database Requirements

- Mongo DB

2) Software Requirements (Platform choice)

OS	OS Minimum	Hardware	Bits
Windows 10 (all editions)	And future OS fixpacks	x86-64	32(64-tolerate)
Windows 2019 (all editions)	And future OS fixpacks	x86-64	64
MacOS11.x(BigSur) *	And future OS fixpacks	x86-64	64
MacOS10.15(Catalina) *	And future OS fixpacks	x86-64	64
Citrix Virtual Apps & Desktop 7.1808+ *	And future OS fixpacks	x86-64	64
CitrixXenApp7.15 *	And future OS fixpacks	x86-64	64
VMWare Virtual Desktop Infrastructure (VDI) 7 *	And future OS fixpacks	x86-64	64
Microsoft Remote Desktop Services 2019	And future OS fixpacks	x86-64	64

3) Hardware Requirements

- Disk Space

Components	Requirement	Applicable OS
Notes Mac64 Standard	Install directory (/Applications) - 2 Gb or more recommended.	All supported Mac operating systems
Notes Windows Basic	Install directory (C:\): 1.5 GB to install the Notes Basic Client 900 MB, needed to run the Notes Basic Client, excluding the size of local mail files.	All supported Windows operating systems
Notes Windows Standard	Install directory (C:\): 2.8GB to install the Notes Standard Client 900 MB, needed to run the Notes Standard Client, excluding the size of local mail files.	All supported Windows operating systems

- Memory

Components	Requirement	Applicable OS
Notes Mac64 Standard	1 GB minimum	All supported Mac operating systems
Notes Windows Basic	1 GB minimum unless OS requires higher minimum	All supported Windows operating systems
Notes Windows Standard	1 GB minimum unless OS requires higher minimum	All supported Windows operating systems

- Display

Components	Requirement	Applicable OS
Notes Mac64 Standard	Monitor: 1024 x 768 screen resolution minimum 32 Mb video RAM or more recommended	All supported Mac operating systems
Notes Windows Basic	Monitor: 1024 x 768 screen resolution minimum Hardware support for OpenGLended	All supported Windows operating systems
Notes Windows Standard	Monitor: 1024 x 768 screen resolution minimum Hardware support for OpenGLended	All supported Windows operating systems

G. Analysis of Models

We have used the waterfall model of the SDLC models due to the following reasons:

1) Requirements are not changing frequently
2) Application is not complicated and big
3) Project is short
4) Requirement is clear
5) Environment is stable
6) Technology and tools used are not dynamic and is stable
7) Resources are available and train

Different phases	Activities performed in each stage
Requirement Gathering stage	During this phase, we planned how to cater to all the requirements that would have been needed in further stages.
Design Stage	Planned the programming language i.e., JavaScript and database i.e., MongoDB based on the functioning along with ther high-level technical details of the project.
Built Stage	Coded the CloudBook website and app along with additional functionalities such as Chrome extension, theme switching, etc.
Test Stage	We tested the software to verify that it is built as per our planning.
Deployment stage	We will deploy the application in the respective environment.
Maintenance stage	Once system is ready to use, we may later require to change the code as per customer request.

VII. OTHER SPECIFICATIONS

A. Advantages

1) Store all notes and important information digitally, usually in a cloud-based storage system.
2) Type, write, and draw notes on the device of choice just as one would using pen and paper.
3) Add files, multimedia, and live recordings to their notes to enrich the meaning and context.
4) Collaborate and share notes with others instantaneously and in real-time.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

H. Limitations and actions performed to overcome -

Risks	Actions
1. Not enough server power.	Reserved additional time for management of server.
2. Bugs and errors occurred in the programming of the website and android app	Referring to the step-by-step technology processes to find out the error.
3. Wrong data/No data sent to the API	Creating exception handling blocks for each expected risk management.

B. Applications

1) Whenever you need to take notes and write down important ideas or information, you will find this helpful.
2) You can use it as a note-taking, organizational, and brainstorming tool.
3) Each note you create can be stored on a virtual page of the notepad.
4) You can store an unlimited number of separate notes. Each can be as long as you desire and have a separate title associated.
5) The Cloudbook works seamlessly with cloud to automatically back up and sync your notes if the computer is connected to the Internet.

VIII. RESULTS

A. Android Application

Notes are displayed in a staggered layout form in the android application. The user opens the app and it will open in a temporary guest account.

1) *Add Notes:* We can add notes from the plus button at the bottom or from left options menu.
2) *Edit/Update Note:* Upon opening a note we can see a pencil button at the bottom right, that allows us to edit the note and then a save button replaces that which allows us to save all changes.
3) *Sync Notes:* If we want to sync all the changes made in the guest account to our actual account we can use Sync notes in the option menu.
4) *Logout:* We can logout from the account using this button.

Below is the android application built with all our proposed model and works in an user-friendly and reactive manner.



Fig. 4 Homepage of the App



Fig. 5 Options Menu

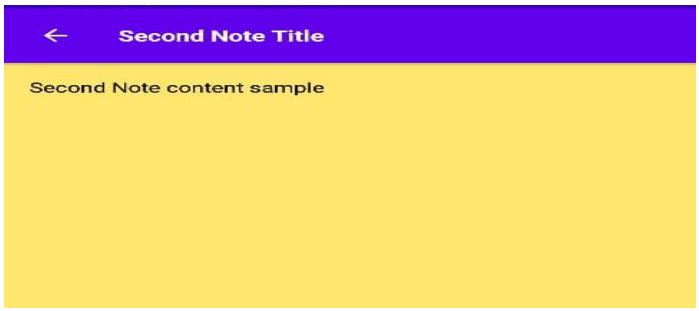


Fig. 6 Insert New note

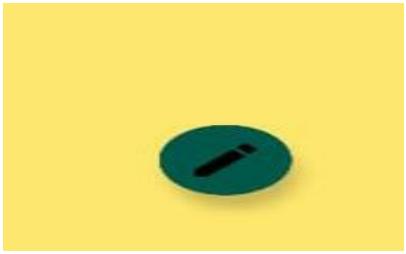


Fig. 7 Edit Button

B. Web Application

Below is the website built with all our proposed model and works in an user-friendly and reactive manner.

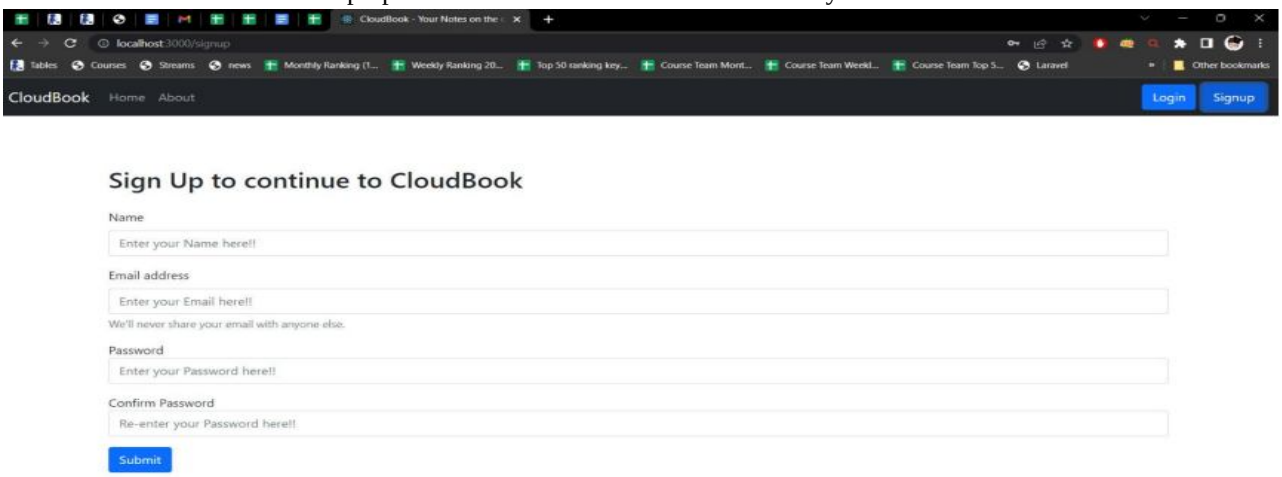


Fig. 8 Sign Up page

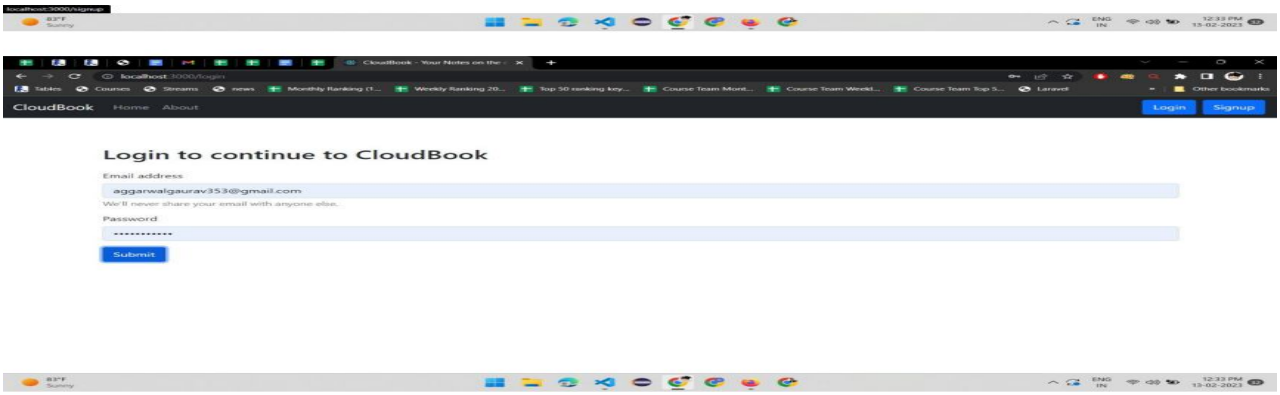


Fig. 5 Login Page

IX. CONCLUSIONS

We have successfully created Cloudbook that can ease taking notes and creating to-do lists and create a convenient and easy-to-use application for users, trying to keep notes. We have created a system that is based on a relational database. We have a database server supporting hundreds of users. Above all, we hope to provide a comfortable user experience along with the best pricing available. Development and Programming: Using the approved design, site architecture, technical brief, and sitemap, the development and programming of the project will begin. This is where the website will come to life and all the elements of the site will come together to produce a high-quality website. If custom web applications and integrations are required within the site, the scope for this project can stretch longer.

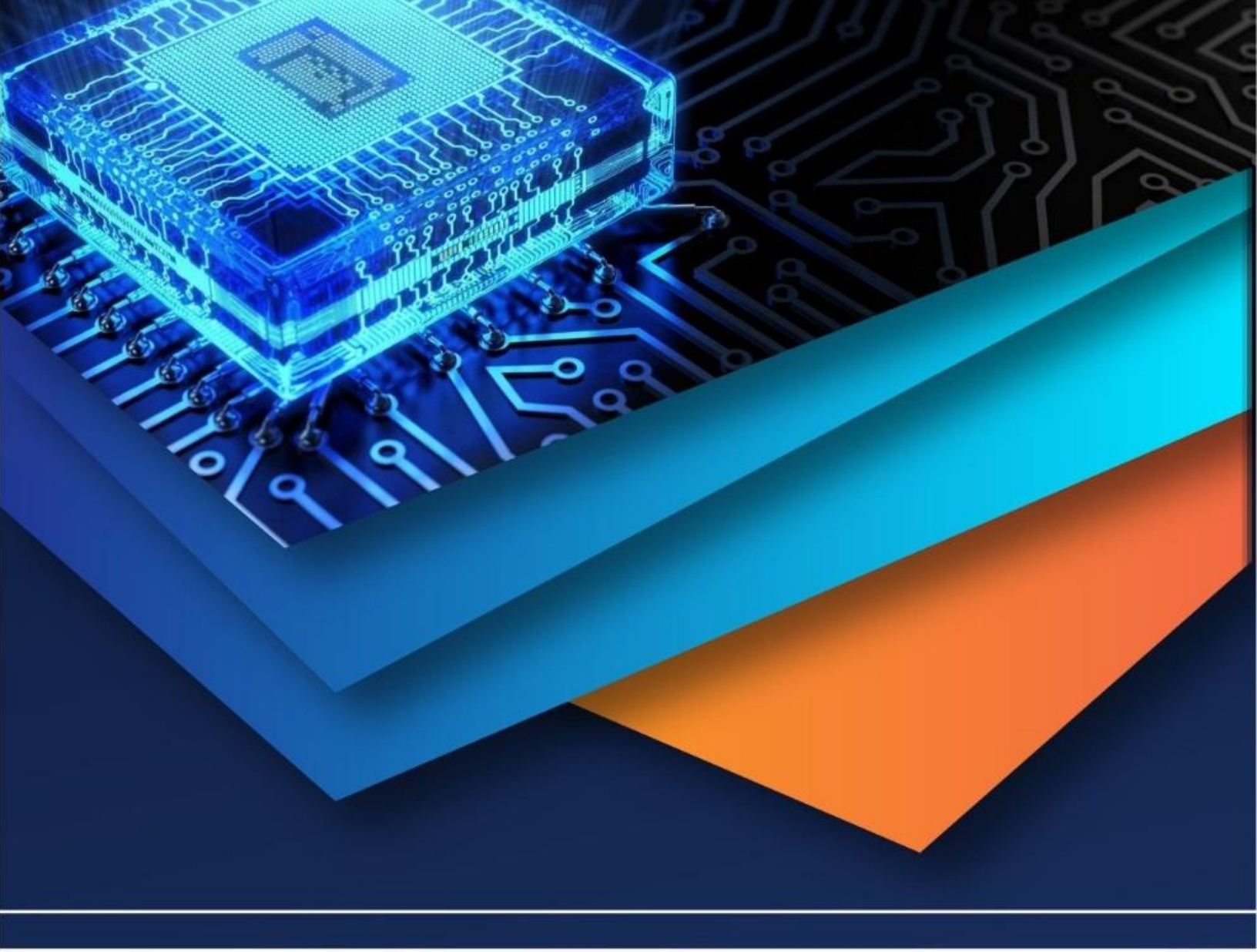
X. ACKNOWLEDGMENT

We would like to express our profound gratitude to our mentor Prof. G.D. Jadhav Sir for his commendable guidance, constant support and encouragement and valuable feedback throughout the project duration distribution. We also want to thank the Department of Computer Science, Shrimati Kashibai Navale College of Engineering for providing resources and research guidance. Finally, we want to thank our publishers for publishing CloudBook Website paper.

REFERENCES

[1] S. M[1] Hanen Abbes, Faiez Gargouri (2016). “Big Data Integration: A MongoDB Database and Modular Ontologies based Approach.”
[2] [2] Piero Fraternali, Sara Comai, Alessandro Bozzon, Giovanni Toffetti (2010). “Engineering Rich Internet Applications with a Model-Driven Approach.”
[3] Sanchit Aggarwal (2018). “Modern Web-Development using ReactJS.”
[4] Anurag Kumar, Ravi Kumar Singh (2021). “Comparative analysis of AngularJS and ReactJS.”
[5] Dipika Damodaran B, Shirin Sali, Surekha Marium Vargese (2016). “Performance Evaluation of MySQL and MongoDB databases.”
[6] Tung Khuat (2018). “Developing a Frontend application using ReactJS and Redux.”
[7] Igibek Koishyabayev, Alexandros Kapravelos (2020). “Reducing the attack surface of Node.js applications.”
[8] Anjali Chauhan (2019). “A review on various aspects of MongoDB databases.”
[9] Anuja Pande, Prof. A.B. Deshmukh, Prof. M.D. Tambakhe (2020). “E-payment Gateway Model.”
[10] Md Arif Hassan, Zarina Shukur, Mohammad Kamrul Hasan (2020). “An Efficient Electronic Payment System for E-Commer

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)