

SISTBOOK

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering degree in Computer Science and Engineering

By

DEVESH GAUTAM (37110185)

ADITYA KRISHNA SINGH (37110019)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND
TECHNOLOGY(DEEMED TO BE
UNIVERSITY)**

**Accredited with Grade "A" by NAAC I 12B Status by UGC I Approved by AICTE
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119**

March - 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with “A” grade by NAAC I 12B Status by UGC I

Approved by

AICTE

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of Mr **DEVESH GAUTAM (37110185)**, **ADITYA KRISHNA SINGH (37110019)** who carried out the project entitled "**SISTBOOK**" under my supervision from November 2020 to March 2021.

Internal Guide

Dr. T.SASIKALA M.E., Ph.D.

Head of the Department

Dr. S. VIGNESHWARI, M.E. , Ph.D.

Submitted for Viva voce Examination held on_____

Internal Examiner

External Examiner

DECLARATION

I DEVESH GAUTAM (37110185), ADITYA KRISHNA SINGH (37110019) hereby declare that the Project Report entitled SISTBOOK done by me under the guidance of Dr. T.SASIKALA M.E., Ph.D., Dean, Department of CSE at Sathyabama Institute Of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering / Technology degree in Computer Science and Engineering.

DATE:

**PLACE:
CANDIDATE**

SIGNATURE OF THE

ACKNOWLEDGEMENT

We are pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

We convey our thanks to **Dr. T.Sasikala M.E., Ph.D ., Dean, School of Computing and Dr.L.Lakshmanan M.E., Ph.D.,and Dr.S.Vigneshwari M.E., Ph.D., Heads of the Department of Computer Science and Engineering** for providing me necessary support and details at the right time during the progressive reviews.

We would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. T.Sasikala M.E., Ph.D** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

We wish to express my thanks to all Teaching and Non-teaching staff members of the Department of **Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

The technologies and functions of social media have significantly changed interaction on the Internet. These changes affect the perceived attractiveness of websites. Prior research regarding classic Internet offers has only partly considered these specifics. A social network service focuses on the building and verifying of online social networks for communities of people who share interests and activities, or who are interested in exploring the interests and activities of others, and which necessitates the use of software. Most services are primarily web based and provide a collection of various ways for users to interact, such as messaging, queries, Advices and file uploading. Existing System is having Message Sending Communities and link attachment feature. But this system doesn't have replying feature to a specific post. Which is the draw back of the project. Several websites are beginning to tap into the power of the social networking model for social good. Such models may be highly successful for connecting otherwise fragmented industries and small organizations without the resources to reach a broader audience with interested and passionate users.

TABLE OF CONTENTS

ABSTRACT	iv
LIST OF FIGURES	vii

CHAPTER NO.	TITLE	PAGE NO.
1.	INTRODUCTION	1
	1.1 HISTORY OF WEB DEVELOPMENT	1
	1.1.1 MAIN FEATURES	2
	1.1.2 TYPES OF WEB APP. DEVELOPMENT	3
	1.1.3 ADVANTAGES	5
2.	LITERATURE REVIEW	6
3.	AIM AND SCOPE	7
	3.1 AIM OF PROJECT	7
	3.1.1 OBJECTIVE	8
	3.2 SCOPE OF PROJECT	12
	3.3 ARCHITECTURE	13
4.	SYSTEM DESIGN AND METHODS	16
	4.1 SPECIFICATIONS	16
	4.2 PROJECT REQUIREMENTS	17
	4.2.1 FRONTEND TECHNOLOGIES	19
	4.2.2 BACKEND TECHNOLOGIES	24
	4.2.3 DATABASE MANagements	26
	4.3 IDE USED	28

5.	RESULTS AND DISCUSSION	29
	5.1 PERFORMANCE ANALYSIS	29
6.	SUMMARY AND CONCLUSION	31
	6.1 FUTURE WORK AND DRAWBACKS	31
	REFERENCES	33
	APPENDIX	34
	A. SOURCE CODE	34
	B. SCREENHOTS	38

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE NO
1.1.2	Types Of Web App Development	4
3.1	Main Logo	8
3.3	React Architecture	14
3.3	System Architecture	15
4.2	Waterfall Model	18
4.2.1	Html	20
4.2.1	Css	21
4.2.1	React JS	23
4.2.1	Material UI	24
4.2.2	Node JS	25
4.2.2	Firebase	25
4.2.3	Firebase DB.	26
4.3	Vs Code	27
5	Login Page	28
5.1	Authentication	29
5.1	Analytics	29

CHAPTER 1

INTRODUCTION

According to Hansen, Shneiderman, and Smith (2011), social media technologies have engendered radically new ways of interacting. To this end, Harris (2009) notes that there are literally hundreds of different social media platforms (e.g., social networking, text messaging, shared photos, podcasts, streaming videos, wikis, blogs, discussion groups),

and Anderson and Wolff (2010) highlight the importance of mobile devices for accessing these platforms. Interestingly and, possibly, confusingly it is not easy to discern among types of social media platforms. The technologies and functions of social media have significantly changed interaction on the Internet. These changes affect the perceived attractiveness of websites. Prior research regarding classic Internet offers has only partly considered these specifics.

1.1 HISTORY OF WEB DEVELOPMENT

The rise of the internet within mainstream culture is largely correlated with its history as a visual medium. Like many other computer systems and applications, it took a graphic interface to get the general population to start to understand the internet's potential. The personal computer would not have proliferated throughout our households and work environments without the display monitor, and still did not become widely used until electronic spreadsheets, word processors and video games started attracting users. Similarly, users did not start buying into the World Wide Web until the visual-oriented web browser started coming into its own in the early 1990s. In 1989, Tim Berners-Lee (then a fellow at the CERN Laboratory in Europe) outlined his concept of a computer platform that could facilitate collaboration among researchers who are based in different parts of the world. This led to the invention of the Hypertext Markup Language (HTML) in 1990. Strongly based on the Standard Generalized Markup Language (SGML), HTML became the fundamental building block of the World Wide Web, and remains at the core of its coding and infrastructure. The standard enabled coders with the ability to organize

web page layouts that could be understood and interacted with over interconnected networks.

The web advanced a great deal in the years following the tech crash of 2000–2001. During this time, government started to play an increasingly influential role in the web, while concurrently, strong tech companies emerged from the ashes of the big collapse to set the new course for digital commerce and culture. And as this newer and more solid foundation was laid, the internet increasingly became the main channel for telecommunications in the modern age.

1.1.1 MAIN FEATURES

- **Quality Web Content.** There's one primary reason people use search engines and browse websites, and that is to search for information. People desire information everyday, and want it delivered in a fast and reliable fashion. Whether to entertain, entice or educate, superior content is a must in every frequently visited website, especially if search engine optimisation is part of the website's overarching marketing strategy
- **Simple and Professional Web Design.** Bells and whistles may seem nice in concept, but they rarely add much value to an effectively constructed and sensible web design. Typically, the websites best at effectively converting site visitors into buying customers, maintain an attractive layout, but keep it clean and simple. Google is an excellent example of such a site. Actually, users found Google's initial design over simplified during the initial testing phases.
- **Clear, User-friendly Navigation.** A stellar web design must contain a user-friendly navigation scheme that allows visitors to quickly find the information needed. Important links must be easy to find and given logical, simple, and include easy-to-understand labels. Calls to action are placed in conspicuous spots within the navigation's scheme. If there is a plethora of content, then a

search box is suggested to make it faster to reach more specific pages within a website.

- **Webpage Speed.** People inherently lose patience quickly, and that holds true when visiting a website. The longer a website takes to load, the more likely a person will leave before it fully renders. Beautiful graphics and substantial content become useless if a site's speed hampers its ability to deliver content quickly.
- **Search Engine Optimisation.** A well-designed website generally will receive many visitors, and one method to attract visitors is search engine optimisation. This entails the insertion of search keywords in website content, an appropriate link profile, social media signals, and over 200 other factors.
- **Web Compatibility.** The variety of browser and platforms which one can now be view web design can present a challenge to developers, but talented ones are accustomed to handling such factors. A site should easily render on various resolutions, screen sizes, and browsers; and with the increasing popularity of mobile devices, websites should function properly on the plethora of these types of devices.

1.1.2 TYPES OF WEB APP. DEVELOPMENT

- **Static Web Applications** are usually developed in HTML and CSS but these are not the only platforms to develop the static app; you can use jQuery and Ajax as per your own convenience. You can also include or display animated objects, such as banners, GIFs, videos, etc. in the static app handily

- **Dynamic web applications** are much more complex on a technical level. They use databases for data loading and its contents are updated each time the user accesses them. They usually have an administration panel (called CMS), where administrators can correct or modify application content, whether text or images.
- **E-Commerce web application** is an online store, its development is likely to resemble that of an m-commerce or e-commerce site. This type of application development process is more complicated because it must allow electronic payments that can be made from credit cards, PayPal, or other payment methods.
- **Portal Web Application** Through the portal, we are referring to a type of application that accesses the various sections or categories through a home page. These applications can include many things: forums, chats, e-mail, browsers, areas accessed through registration, the latest content, etc.

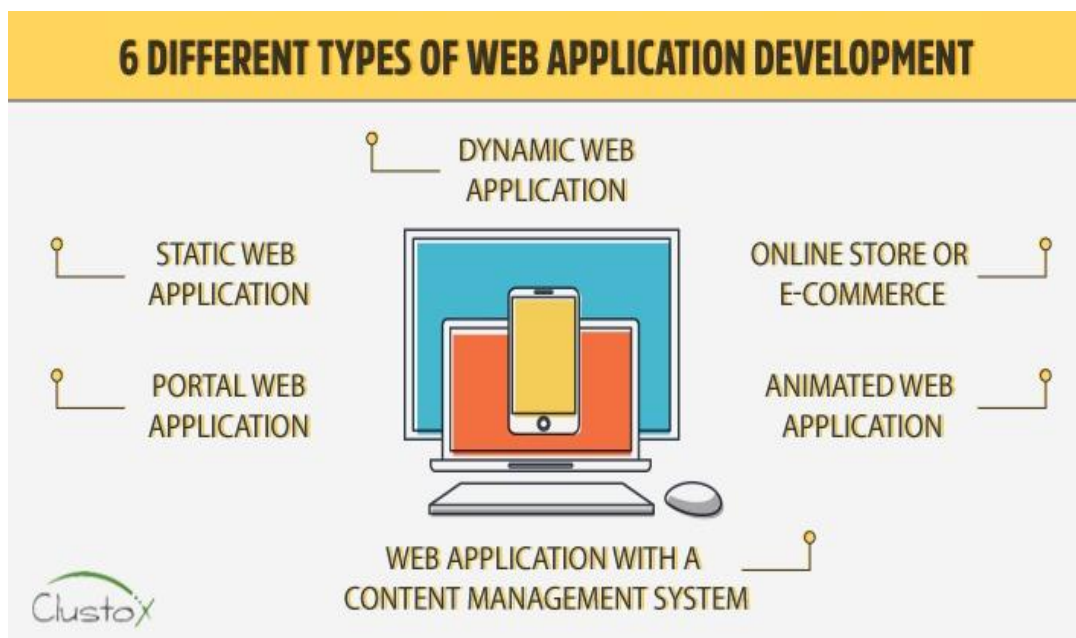


Fig: 1.1.2 TYPES OF WEB APP. DEVELOPMENT

- **Animated Web Applications** The animation is inevitably associated with Flash technology. This programming approach allows you to display content with animated effects. This type of application allows more creative and modern designs and is one of the key technologies used by designers and creative directors.
- **Web Applications With A Content Management System** The content should be constantly updated when it comes to web application development so that the installation of a content management system (CMS) is a serious option to consider. The administrator can make use of this CMS to implement changes and updates alone.

1.1.3 ADVANTAGES

- Web apps are developed with programming languages such as HTML and CSS, which are well known among IT professionals.
- Unlike native apps, a single web app is able to reach out to all devices. It's programmed to run on any operating system. It must appropriately adapt to iOS, Android, and Windows Phone — amongst other operating systems.
- These apps run on the device's own web browser through a simple URL. They don't need to be downloaded and installed from app stores like Google Play or Apple's App Store. This translates into money-saving since having a direct link through a web app is free.
- The most important benefit you can draw off from it is its price. Web app development is the cheaper kind of app development. It consists of creating a link or several links between the application and an URL.

Developing a native app or an interpreted app entails a higher cost but its chances to succeed are far greater.

CHAPTER 2

LITERATURE REVIEW

METHODOLOGY

A selection process has been carried out in order to identify the most relevant research on social media: after ensuring that the content was related to facebook, the studies have been regrouped into eight main research themes: effects on the users. Therefore, the literature employed in this study is composed mainly of published papers or studies presented at academic conventions. We therefore focus on the personality of users: why they joined, how they build networks, and how they interact; and on how organizations may act on, and benefit.

LEARNING ABOUT OTHERS

The term 'surveillance' was first used by Lampe and his colleagues (2006) in one of the most influential studies. Past research has focused on differences behavior relating to gender, personality, social status, age, and race.

Building and maintaining a network

SNSs such as Facebook visualize one's network (Donath, 2007). A quick search reveals that users vary widely in the number of friends they have, ranging from only one to more than a thousand.

CONCLUSION

The biggest problem that researchers face when exploring social media is its ever-changing nature. The phenomenon of social networks has developed extremely quickly in the last years, and so have the uses, habits, and conceptions of Facebook, twitter etc. Starting as a social network between students belonging to the same university.

CHAPTER 3

AIM AND SCOPE

3.1 AIM OF PROJECT

The vast majority of human knowledge is still not on the internet. Most of it is trapped in the form of experience in people's heads, or buried in books and papers that only experts can access. As a consequence most people don't have the knowledge they should, and don't have the time, connections, or skills to get to the knowledge they would have in an ideal world.

The internet was supposed to allow anyone to set up a web page and share their knowledge with the world. But in practice it's too difficult and takes too long and almost no one does it. Blogs are easy to start, but unless the author is famous, it takes years to build a following. More than a billion people use the internet yet only a tiny fraction contribute their knowledge to it.

Our Facebook-clone(SISTBOOK) aims to allow anyone to easily share their knowledge and in the process to dramatically increase the total amount of knowledge available to the world. As we grow, we will be able to provide larger and larger audiences to writers, cover more and more topics, and have greater and greater impact on the world.

We hope to become an internet-scale Library of Alexandria, a place where hundreds of millions of people go to learn about anything and share everything they know. To do that we are going to have to expand. Today it is largely questions and answers, but that is not the ideal format for all knowledge. Other formats will gradually be added as we scale up.

All of this will be guided by our mission.



Fig:3.1 MAIN LOGO

3.1.1 OBJECTIVE

1. Create Brand Awareness:

Making brand mindfulness abbreviates the business procedure, expands a piece of the pie, and positions a brand as an innovator in its area. It's important to play out an inner review to assess brand situating procedures and to comprehend the effect of showcasing activities on the brand picture. Brand separation can create from various activities, for example, the making of unique substance, customized advancements went for the intended interest group, prospect profiling, item

improvement dependent on social bits of knowledge, and correspondence techniques adjusted for socio-socioeconomics of intrigue.

Here are some KPIs to help track and report on the achievement of this target:

Notices: the occasions the brand comes up in online life discussion and the occasions these discussions are shared.

Impressions: the number of potential perspectives that a message has.

Come to: the number of individuals who have gotten seen a distribution.

Social Sessions: the quantity of webpage visits, blog entry sees, and so on.

The portion of Voice: volume of notices gotten by a brand versus volume of notices gotten by all applicable contending brands.

dig mind-social-listening-TREND realistic speaking to the quantity of brand refers to after some time, using Digimind Social.

2. Increment Size of Social Communities and Accurately Target Audiences

The pertinent KPIs here are:

Network Performance: the number of fans, supporters, or endorsers separated by web-based life channel.

Network Progression: the supporter development rate by internet based life channel.

The portion of Community Voices: number of supporters contrasted with those of contenders.

The portion of Voice by Channel: the network size of every internet based life stage as a level of the all outnumber of social adherents over all stages.

Social Visits: guests coming to web-based life records originating from the brand's site or blog.

development of-subscribersThe development of supporters after some time by online networking stage, through Digimind Social.

3. Fortify Engagement Strategies to Increase Customer Loyalty

The essential goal here is to make a dependable connection between the intended interest group and the brand. It's important to connect with web-based life devotees who have a specific degree of believability or impact among the intended interest group. For instance, if your intended interest group is football fans, an NFL player would have a social impact among this statistic. The influencer doesn't really should be hugely powerful on a worldwide level - at times a little, dynamic network is more successful than a bigger,. increasingly latent network. Powerful online life systems ought to incorporate influencers with a submitted following and brand envoys with abnormal amounts of commitment

The KPIs for this goal are:

Communication Performance: the number of associations (likes, retweets, shares,

and so on.) and the number of remarks.

Advancement of the Interactions: the development pace of communications.

Distribution Performance: the number of messages creating a measure of association more prominent than a predefined limit.

Influencer Population: the number of influencers and brand envoys as a level of the network size.

4. Screen Customer Feedback

To improve client experience, it's important to catch customer sentiments concerning applicable items and measure shopper fulfillment and advancement after some time.

The accompanying KPIs ought to be estimated for this goal:

Brand Perception: the estimation related to the brand and its items, particularly in contrast with contenders.

Development of Perception: the level of positive versus negative opinion after some time.

Notoriety Score: the slants related with the brand weighted against the network size of each audit.

Reaction Performance: the reaction rate to web-based life cooperations.

Goals Performance: the number of client tickets settled over a given timeframe.

reactionsbymediaShare of social collaborations by social stage, using Digimind Social.

5. Convert Social Followers into Qualified Leads and New Business

Towards the part of the bargain, brand substance ought to have produced web-based life adherents and associations with planned clients. The related KPIs here are:

Number of Leads from Social Media: the portion of advertising leads starting from online networking which is transformed into qualified business leads and coordinated into the CRM stage.

Lead Growth from Social Media: the development pace of transformations from social prompts potential customers.

These destinations must be normally revealed for beginning examinations, which will at that point be supplemented with the prescient and restorative investigation.

3.2 SCOPE OF PROJECT

This is the age of contacts for interaction between people to people through social platforms in the web. The tremendous popularity of using the social media networking could never have been realized before. In fact, social media have become an important tool of marketing in true sense of customer orientation. The scope of social networking is widening, and today it offers a strong support to the companies in providing the much desired touch of concern. Future of social media networking brings exciting promises as expected from the experience of limited users from a few companies in the top bracket.

This is the beginning of interweaving social media networking as a positive business process bringing two people together than merely two organizational entities. People can communicate their message to the suppliers or customers and also convey any difference of opinion over the social networks. The business community has started to take a keen interest in the enormous possibilities of reaching and serving the buyers in this emerging domain of marketing.

Social media networking is a more practical approach for dealing with the consumers. It may be for improving the products listening to the customers' views with critical appreciations. It may also help improving the ways of dealing with buyers using them as ears and the eye of the organizations. All the big companies are quite optimistic about the positive impact of Social media networking and gone into developing fully integrated processed with a holistic touch. Take the example of activities the companies like IBM, Microsoft and Google performs to strengthen their networking platforms via social networks like Twitter and Facebook.

The bright future prospect of social networking is also proven with the fact that the technology is integrated in the mobile phones as well. Look at the devices designed with powers of interpersonal communication on a globalized outlook. These devices are now being created to with the perspective of global interaction and messaging across geographical boundaries. The concept is one, wherever on the earth; reach the common platform of the people for everybody's benefit.

According to the student's social media platforms is a platform where they can share their personal views, feelings, perception regarding the current or previous events and for their beloved ones.

Most of the students create a proper group and share notes, important guidelines, lectures, and other study material that was given by their teacher or professor. By making use of social media students can easily get the information regarding international universities through reviews that help them to pick the right institute for their academic session.

3.3 ARCHITECTURE

When thinking of JavaScript application architecture, the MVC pattern immediately comes to mind for many people. React isn't a MVC framework like Angular or Ember,

but is instead a library that handles views. If that's the case, then what architecture does a typical large-scale React application use?

On this project, we used Redux. Redux is both a library and an architecture/pattern that is influenced by the Flux architecture from Facebook.

Two distinguishing characteristics of Redux are strict unidirectional data flow (shared with Flux) and storing all application state as a single object. The data lifecycle in a Redux app has four steps:

- Dispatch an action.
- Call to a reducer function.
- A root reducer combines the output of the various reducer functions into a single state tree.
- The store saves the state returned by the root reducer triggering view updates.

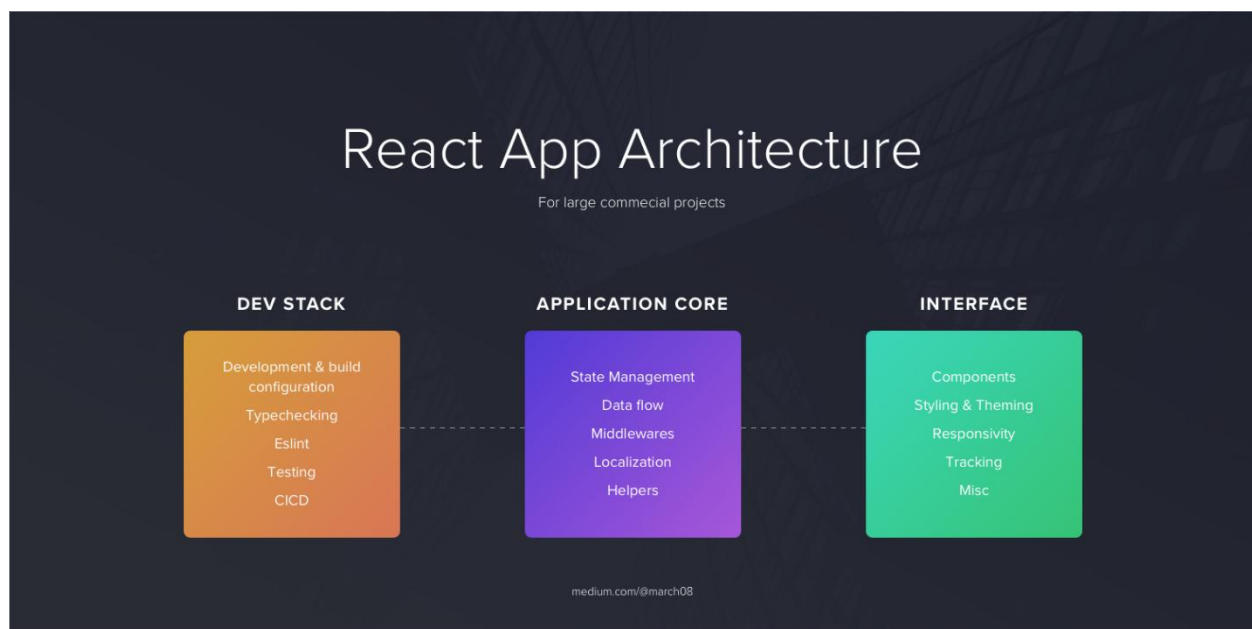


Fig: 3.3 REACT ARCHITECTURE

The API is a critical part of the application. The real business case for an architecture like this is to have a single data source serve content across a range of platforms in a

consistent, efficient way. Client devices make an HTTP request to the API and receive a response with the requested data, usually in JSON format.

Caching Layers Having a caching layer in front of your API and Node.js servers helps reduce load on the API server and decreases response time.

Node.js Server We've finally gotten to where the code for the React web app lives. We used the server-side JavaScript application framework, Express, on the project. Express was used to create an HTTP server that responds to requests from web clients. It's also where we did the server-side rendering of the React application.

Clients In the diagram, I've added icons to represent mobile apps and web browsers, respectively, but any number of devices may consume the API's data. For example, our client serves not only web browsers and a mobile app from the API, but also Roku boxes and Samsung TVs.

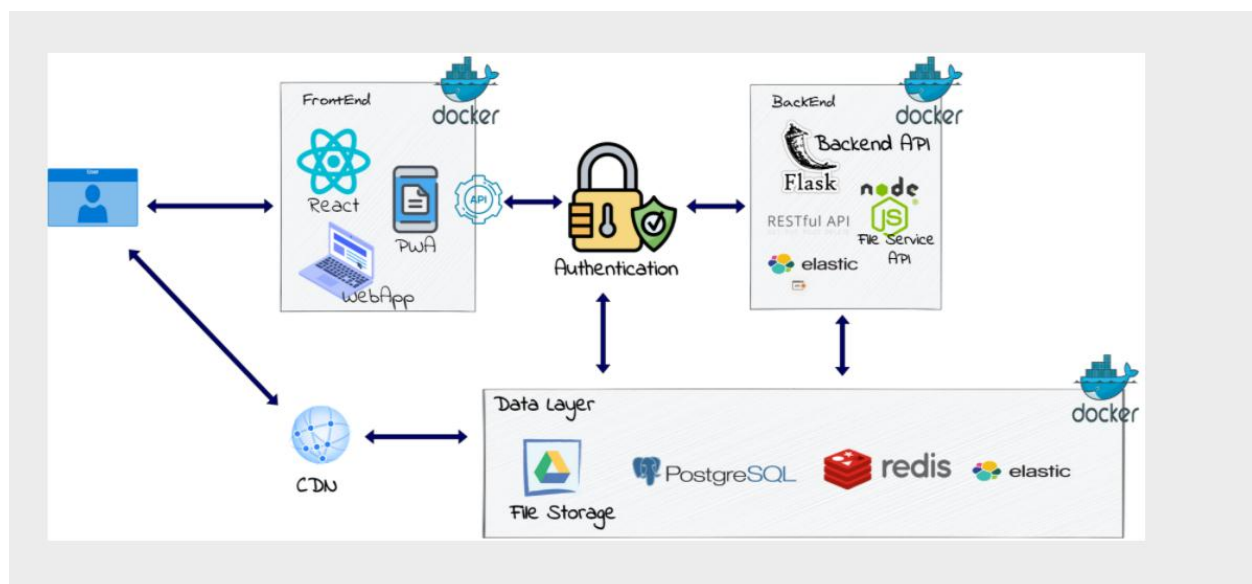


Fig: 3.3 SYSTEM ARCHITECTURE

CHAPTER-4

SYSTEM DESIGN AND METHODS

4.1 SPECIFICATIONS

When programming, one of the most important things to consider is which programming language you will be using. Whether it's something like Python for game development or JavaScript for web design, each programming language does things better than others. Likewise, knowing the best computer specs for programming is super important.

Resolution: Upgrade Your Monitor

While programming, web developers and software engineers spend hours at a time looking at their monitors. As computer experts, they spend time parsing through different lines of code and determining what works and what doesn't. This is why choosing the perfect monitor is key to a successful run at programming. We recommend you always consider a monitor with a full 1920x1080p HD resolution. While you may

think that 1920×1080 is only useful for things like streaming movies in 4K and playing games, it offers so much more.

GPU Power: The Graphics Processor

As a programmer, you're going to need to ensure programs that utilize graphics can run properly. While this may be more specific to individual graphic programming, it's still a fantastic idea to update your graphics card as well. A good rule of thumb regarding graphics cards is checking out anything from NVIDIA and AMD. Notably, the NVIDIA Geforce and AMD Ryzen are fantastic GPUs for desktops.

More Storage Space: An Up-to-Date SSD/Hard Drive

Arguably the most significant upgrade to your processor is your SSD, or solid-state drive, for optimal speed. The last thing you need while programming is for your system to bog down and move at a snail's pace using a traditional hard drive. That's why having a state-of-the-art SSD drive is arguably more critical than other minimum requirements.

RAM

As a developer, you're going to want to increase every bit of memory that you can. Demanding projects can take up a lot of time and hard drive space, so you need as much RAM as you can get. The best specs for RAM are 16GB. Ideally, it would be best if you shot for something like 32GB. At a minimum, 8GB might suffice.

A Fast CPU

Your CPU must be up to date. Central processing units are considerably more challenging to replace than graphics cards and SSD drives, especially in laptops. Minimum i3-4750 7th gen should do.

4.2 PROJECT REQUIREMENTS

First of all we need to implement a software development method. Here we are using Waterfall model –

The waterfall model is a linear, sequential approach to the software development life cycle (SDLC) that is popular in software engineering and product development. The waterfall model emphasizes a logical progression of steps. Similar to the direction water flows over the edge of a cliff, distinct endpoints or goals are set for each phase of development and cannot be revisited after completion. The term was first introduced in a paper published in 1970 by Dr. Winston W. Royce and continues to be used in applications of industrial design.

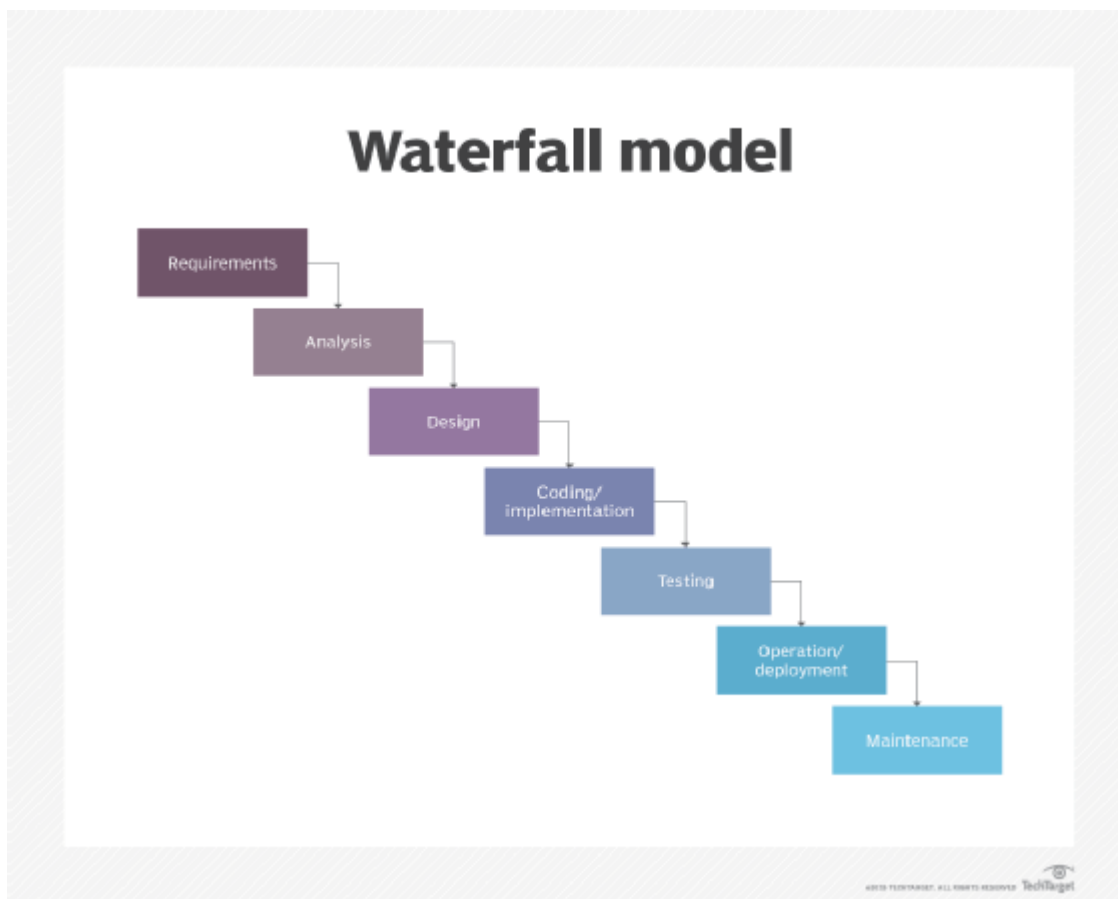


Fig: 4.2 WATERFALL MODEL

The waterfall methodology is composed of seven non-overlapping stages:

Requirements:

Potential requirements, deadlines and guidelines for the project are analyzed and placed into a functional specification. This stage handles the defining and planning of the project without mentioning specific processes.

Analysis:

The system specifications are analyzed to generate product models and business logic that will guide production. This is also when financial and technical resources are audited for feasibility.

Design:

A design specification document is created to outline technical design requirements such as programming language, hardware, data sources, architecture and services. Coding/Implementation: The source code is developed using the models, logic and requirements designated in the prior stages. Typically, the system is designed in smaller components, or units, before being implemented together.

Testing:

This is when quality assurance, unit, system and beta tests take place to report issues that may need to be resolved. This may cause a forced repeat of the coding stage for debugging. If the system passes the tests, the waterfall continues forward.

Operation/Deployment:

The product or application is deemed fully functional and is deployed to a live environment.

Maintenance:

Corrective, adaptive and perfective maintenance is carried out indefinitely to improve, update and enhance the final product. This could include releasing patch updates or releasing new versions.

After planning the timeline of the project we need to bring in the Modules.

4.2.1 Frontend Technologies

4.2.2 Backend Technology

4.2.3 Database Management

4.2.1 FRONTEND TECHNOLOGIES

HTML – First developed by Tim Berners-Lee in 1990, HTML is short for Hypertext Markup Language. HTML is used to create electronic documents (called pages) that are displayed on the World Wide Web. Each page contains a series of connections to other pages called hyperlinks. Every web page you see on the Internet is written using one version of HTML code or another. HTML code ensures the proper formatting of text and images for your Internet browser. Without HTML, a browser would not know how to display text as elements or load images or other elements. HTML also provides a basic structure of the page, upon which Cascading Style Sheets are overlaid to change its appearance. One could think of HTML as the bones (structure) of a web page, and CSS as its skin (appearance). HTML can embed programs written in a scripting language such as JavaScript, which affects the behavior and content of web pages. Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), former maintainer of the HTML and current maintainer of the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.



Fig: 4.2.1 HTML

CSS - Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript. CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file which reduces complexity and repetition in the structural content as well as enabling the .css file to be cached to improve the page load speed between the pages that share the file and its formatting. Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such

as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobile device.



Fig: 4.2.1 CSS

JavaScript - JavaScript (JS) is a lightweight, interpreted, or just-in-time compiled programming language with first-class functions. While it is most well-known as the scripting language for Web pages, many non-browser environments also use it, such as Node.js, Apache CouchDB and Adobe Acrobat. JavaScript is a prototype-based, multi-paradigm, single-threaded, dynamic language, supporting object-oriented, imperative, and declarative (e.g. functional programming)A scripting language like JavaScript (JS, for those in the know) does the heavy lifting by telling computer programs like websites or web applications to “do something.” In the case of JavaScript, this means telling those dynamic features described earlier to do whatever it is they do—like telling images to animate themselves, photos to cycle through a slideshow, or autocomplete suggestions to respond to prompts. It’s the “script” in JavaScript that makes these things happen seemingly on their own.

ReactJs - React was created by Jordan Walke, a software engineer at Facebook, who released an early prototype of React called "FaxJS". He was influenced by XHP, an HTML component library for PHP. It was first deployed on Facebook's News Feed in 2011 and later on Instagram in 2012. It was open-sourced at JSConf US in May 2013. React (also known as React.js or ReactJS) is an open-source, front end, JavaScript library for building user interfaces or UI components. It is maintained by Facebook and a community of individual developers and companies. React can be used as a base in the development of single-page or mobile applications. However, React is only concerned with state management and rendering that state to the DOM, so creating React applications usually requires the use of additional libraries for routing, as well as certain client-side functionality.

The Context API makes it easier to have our global and app-wide data available to all components therefore, making it easier and more accessible. The Context API has been considered as one of the key features of the react framework and a prominent improvement in the recent releases of the framework.

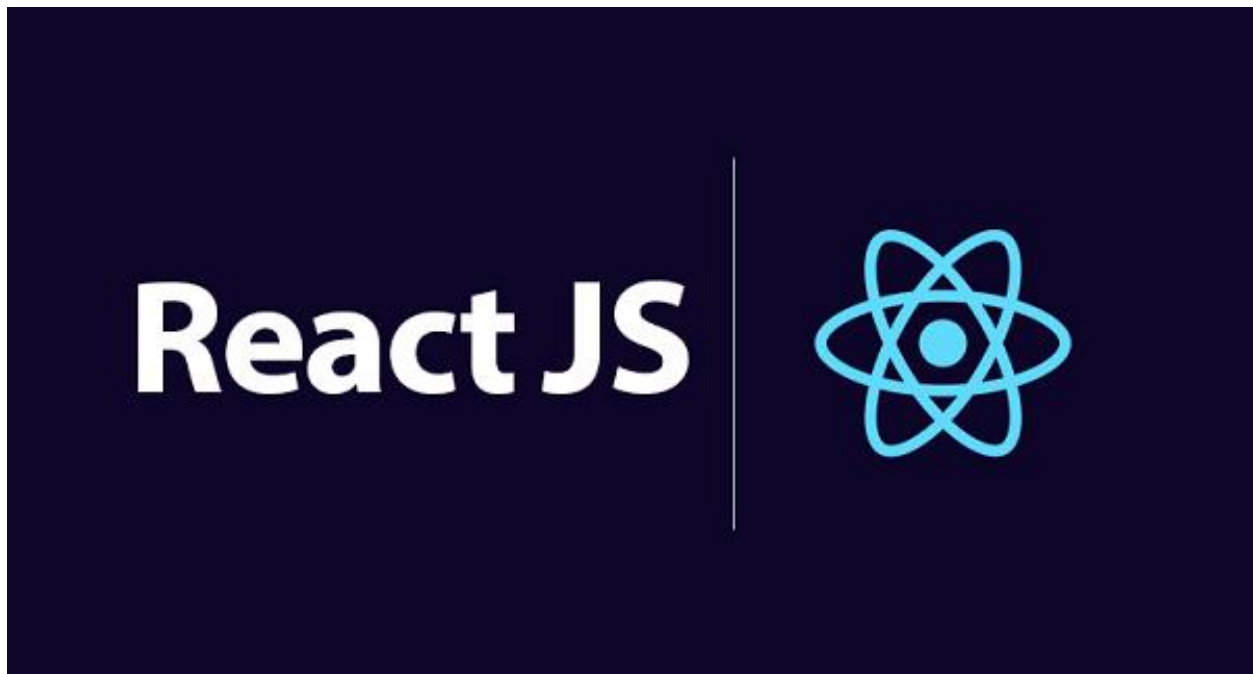


Fig: 4.2.1 REACT JS.

Material UI – It kick-started in 2014, not long after React came out to the public, and has grown in popularity ever since. With over 35,000 stars on GitHub, Material-UI is one of the top user interface libraries for React out there. Its success didn't come without challenges though. Designed with LESS, Material-UI v0.x was prone to common CSS pitfalls, such as global scope, which lead the project on the CSS-in-JS trajectory. This is how next came about in 2016. Material UI is a component library for React teeming with powerful components that you should be using in your projects. If you're just looking to create a good looking app, Material UI can provide you with solid pre-styled components that will get the job done. If you are looking to customize those components Material UI gives you the tools to do that as well.



Fig: 4.2.1 MATERIAL UI

4.2.2 BACKEND TECHNOLOGIES

Nodejs - Node.js is an open-source, cross-platform, back-end JavaScript runtime environment that runs on the V8 engine and executes JavaScript code outside a web browser. Node.js lets developers use JavaScript to write command line tools and for server-side scripting—running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser. Consequently, Node.js represents a "JavaScript everywhere" paradigm, unifying web-application development around a single programming language, rather than different languages for server-side and client-side scripts. Node.js also provides a rich library of various JavaScript modules which simplifies the development of web applications using Node.js to a great extent.



Fig:4.2.2 NODE JS.

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



Fig:4.2.2 FIREBASE

4.2.3 DATABASE MANAGERMENTS

Firestore DB - The Firestore Realtime Database is a cloud-hosted database in which data is stored as JSON. The data is synchronized in real-time to every connected client. All of our clients share one Realtime Database instances and automatically receive updates with the newest data, when we build cross-platform applications with our iOS, and JavaScript SDKs. The Firestore Realtime Database is a NoSQL database from which we can store and sync the data between our users in real-time. It is a big JSON object which the developers can manage in real-time. By using a single API, the Firestore database provides the application with the current value of the data and updates to that data. Real-time syncing makes it easy for our users to access their data from any device, be it web or mobile. The Realtime database helps our users collaborate with one another.



Fig:4.2.3 FIREBASE DB.

4.3 IDE USED

An integrated development environment (IDE) is a software application that provides comprehensive facilities to computer programmers for software development. An IDE normally consists of at least a source code editor, build automation tools and a debugger. Some IDEs, such as NetBeans and Eclipse, contain the necessary compiler, interpreter, or both; others, such as SharpDevelop and Lazarus, do not. The boundary between an IDE and other parts of the broader software development environment is not well-defined; sometimes a version control system or various tools to simplify the construction of a graphical user interface (GUI) are integrated.

VS CODE - Visual Studio Code is a freeware source-code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality. Microsoft has released Visual Studio Code's source code on the microsoft/vscode (Code - OSS) repository of GitHub, under the permissive MIT License, while the releases by Microsoft are freeware. In the Stack Overflow 2019 Developer Survey, Visual Studio Code was ranked the most popular developer environment tool, with 50.7% of 87,317 respondents reporting that they use it. Visual Studio Code was first announced on April 29, 2015, by Microsoft at the 2015 Build conference. A Preview build was released shortly thereafter. On November 18, 2015, Visual Studio Code was released under the MIT License, having its source code available on GitHub. Extension support was also announced. On April 14, 2016, Visual Studio Code graduated from the public preview stage and was released to the Web.



Fig:4.3 VS CODE

CHAPTER-5

RESULTS AND DISCUSSION

By the end of the project we hope to achieve our main reason of creating something like this to decrease the communication gap between the students inside the campus. And help them with any information or help they need at that point of time. And we believe we have been successful in doing so, but with limitations to its scalability because we think there is a lot of room to scale up this project. So now we have a fully fledged working web application in which 100% of the work has been done manually and keeping facebook as a reference. It is not a desktop only application which is the reason we were successful in implementing Reactjs because of which it works perfectly on almost every device with a browser and internet facility.

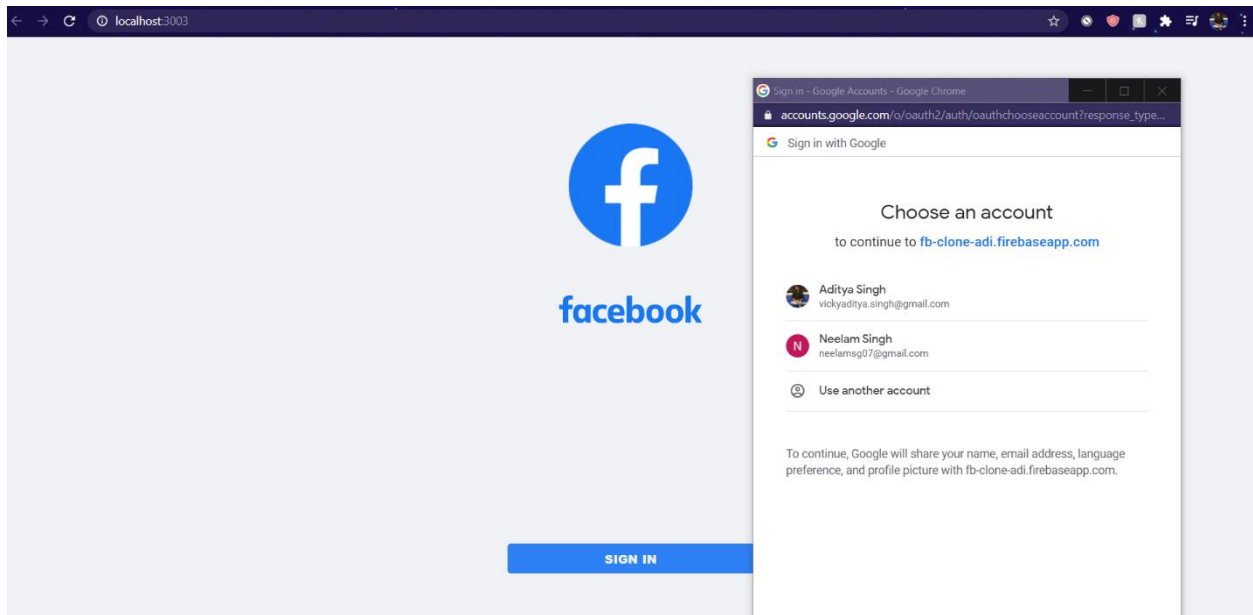


Fig: 5 LOGIN PAGE

5.1 PERFORMANCE ANALYSIS

All the users who visit this application, will definately be worried about the security of their respective gmail accounts. This was the main reason why we chose Firebase because of its google background and along with all the features it provides. One of them being the firestore database which keeps track of all the user logins. We have attached some screenshots to its support below.

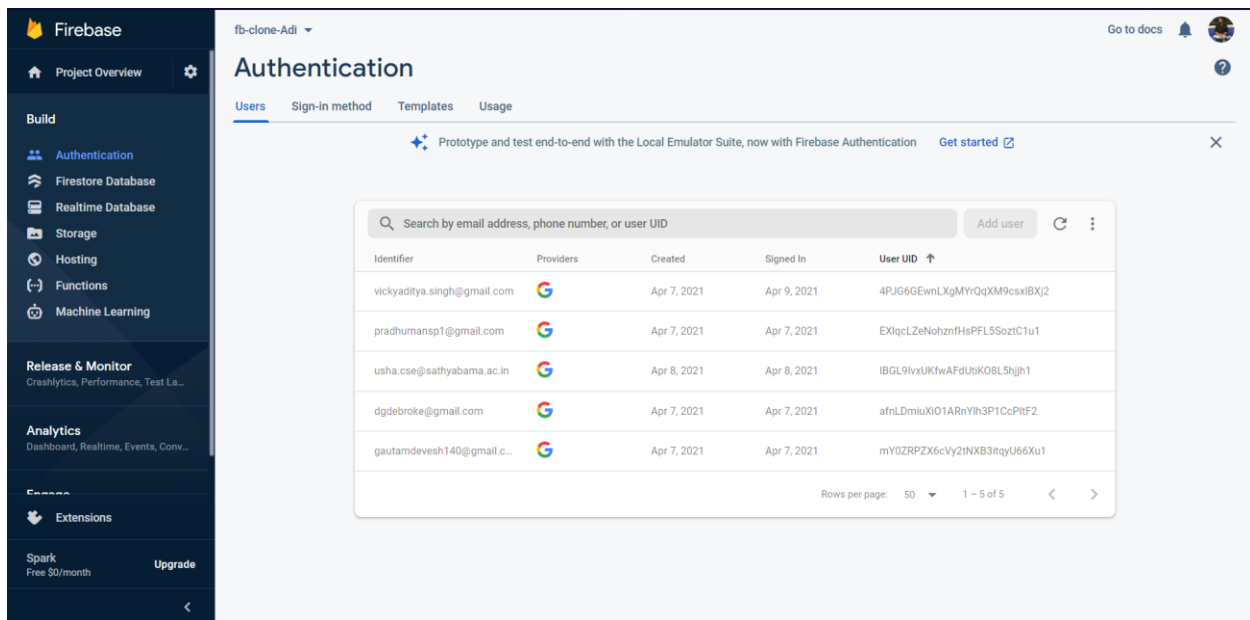


Fig: 5.1 AUTHENTICATION

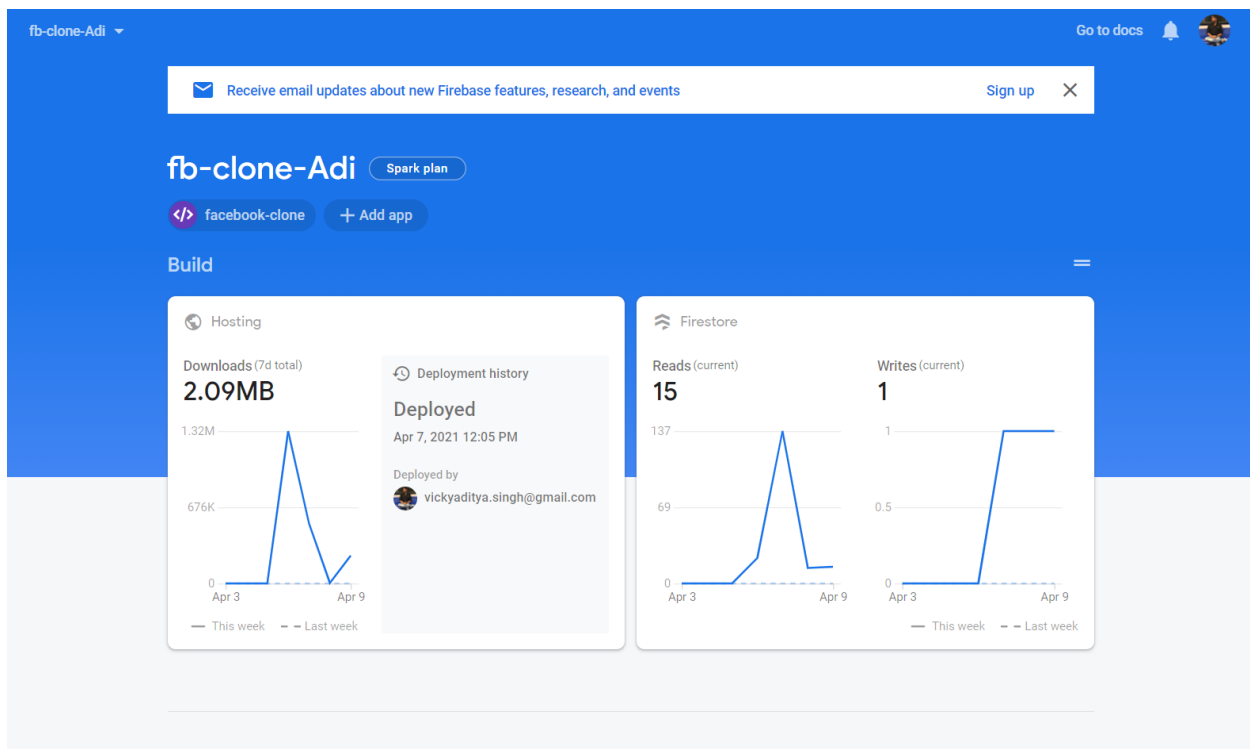


Fig:5.1 ANALYTICS

CHAPTER-6

SUMMARY AND CONCLUSION

All the users who visit this application, will definately be worried about the security of their respective gmail accounts. This was the main reason why we chose Firebase because of its google background and along with all the features it provides. One of them being the firestore database which keeps track of all the user logins. We have attached some screenshots to its support below.

6.1 FUTURE WORK AND DRAWBACKS

As I mentioned in the begining of the report that there is a lot of room to scale up this project, like adding more features such as chat options, audio input, video input and adding more layers of security. One of the features we personally wanted to add was to limit the users to the students of our college only with some login credentials which can only be used by the people who are registered in only our university. To add more features and technicalities we will require a bigger team with specific skillsets. For, now we are planning to upload this project to public on github so that we can get more contributions.

Drawbacks

There are mainly 2 drawbacks

1. We can only upload Messages, gifs and pictures.
2. The speed of page rendering may be affected with increasing traffic.
3. ReactJS uses JSX. It's a syntax extension that allows HTML with JavaScript mixed together. This approach has its own benefits, but some members of the development community consider JSX as a barrier.
4. ReactJS Covers only the UI Layers of the app and nothing else. So you still need to choose some other technologies to get a complete tooling set for development in the project.

REFERENCES

- Occhino, Tom; Walke, Jordan. ["JS Apps at Facebook"](#). YouTube. Retrieved 22 Oct 2018.
- Krill, Paul (May 15, 2014). ["React: Making faster, smoother UIs for data-driven Web apps"](#). [InfoWorld](#).
- ["Using the Effect Hook – React"](#). [reactjs.org](#). Retrieved 2020-01-24.
- Frederic Lardinois (18 April 2017). ["Facebook announces React Fiber, a rewrite of its React library"](#). TechCrunch. Retrieved 19 April 2017.
- Dere, Mohan (2018-02-19). ["How to integrate create-react-app with all the libraries you need to make a great app"](#)
- ["Explaining React's License"](#). *Facebook*. Retrieved 2017-08-18
- Abramov, Dan. ["The History of React and Flux with Dan Abramov"](#). *Three Devs and a Maybe*. Retrieved 7 April 2018.
- Clark, Andrew (September 26, 2017). ["React v16.0\\$MIT licensed"](#). *React Blog*.

APPENDIX

A. SOURCE CODE

```
import React from 'react';
import './App.css';
import Login from './Login';
import Feed from './Feed';
import Header from './Header';
import Sidebar from './Sidebar';
import Widgets from './Widgets';
import { useStateValue } from './StateProvider';

function App() {
  const [{ user }, dispatch] = useStateValue();
  return (
    <div className="app">
      {!user ? (
        <Login />
      ) : (
        <>

          <Header />

          <div className="app__body">
            <Sidebar />
            <Feed />

            <Widgets />

          </div>
        </>
      )}

    </div>
  );
}

export default App;

import { Button } from '@material-ui/core';
import React from 'react';
import './Login.css';
import { auth, provider } from './firebase';
import { actionTypes } from './reducer';
import { useStateValue } from './StateProvider';
```

```

function Login() {
  const [state, dispatch] = useStateValue();

  const signIn = () => {

    auth.signInWithPopup(provider)
    .then(result => {
      dispatch({
        type: actionTypes.SET_USER,
        user: result.user,
      });
      console.log(result)
    }).catch(error => alert(error.message))

  };
  return (
    <div className = "Login">
      <div className="login__logo">
        
        
      </div>
      <Button type="submit" onClick={signIn}>
        >
        Sign In
      </Button>

    </div>
  );
}

```

```

export default Login;

```

```

import React from "react";
import "./Post.css";
import { Avatar } from "@material-ui/core";
import ThumbUpIcon from "@material-ui/icons/ThumbUp";
import ChatBubbleOutlineIcon from "@material-ui/icons/ChatBubbleOutline";
import AccountCircleIcon from "@material-ui/icons/AccountCircle";
import NearMeIcon from "@material-ui/icons/NearMe";

```

```

import { ExpandMoreOutlined } from "@material-ui/icons";

function Post({ profilePic, image, username, timestamp, message }) {
  return (
    <div className="post">
      <div className="post__top">
        <Avatar src={profilePic}
          className="post__avatar" />
        <div className="post__topInfo">
          <h3>{username}</h3>
          <p>{new Date(timestamp?.toDate()).toUTCString()}</p>
        </div>
      </div>

      <div className="post__bottom">
        <p>{message}</p>
      </div>

      <div className="post__image">
        <img src={image} alt="" />
      </div>

      <div className="post__options">
        <div className="post__option">
          <ThumbUpIcon />
          <p>Like</p>
        </div>
        <div className="post__option">
          <ChatBubbleOutlineIcon />
          <p>Comment</p>
        </div>
        <div className="post__option">
          <NearMeIcon />
          <p>Share</p>
        </div>
        <div className="post__option">
          <AccountCircleIcon />
          <ExpandMoreOutlined />
        </div>
      </div>

    </div>
  );
}

export default Post;

```

```

import React from "react";
import "./Sidebar.css";
import SidebarRow from "./SidebarRow";
import LocalHospitalIcon from "@material-ui/icons/LocalHospital";
import EmojiFlagsIcon from "@material-ui/icons/People";
import PeopleIcon from "@material-ui/icons/People";
import ChatIcon from "@material-ui/icons/Chat";
import StorefrontIcon from "@material-ui/icons/Storefront";
import VideoLibraryIcon from "@material-ui/icons/VideoLibrary";
import { ExpandMoreOutlined } from "@material-ui/icons";
import { useStateValue } from "./stateProvider";

function Sidebar() {
  const [{ user }, dispatch] = useStateValue();
  return (
    <div className="sidebar">
      <SidebarRow src={user.photoURL} title={user.displayName} />
      <SidebarRow Icon={LocalHospitalIcon} title="COVID-19 Information Center"/>
      <SidebarRow Icon={EmojiFlagsIcon} title="Pages"/>
      <SidebarRow Icon={PeopleIcon} title="Friends"/>
      <SidebarRow Icon={ChatIcon} title="Messenger"/>
      <SidebarRow Icon={StorefrontIcon} title="Marketplace"/>
      <SidebarRow Icon={VideoLibraryIcon} title="Videos"/>
      <SidebarRow Icon={ExpandMoreOutlined} title="Marketplace"/>

    </div>

  );
}

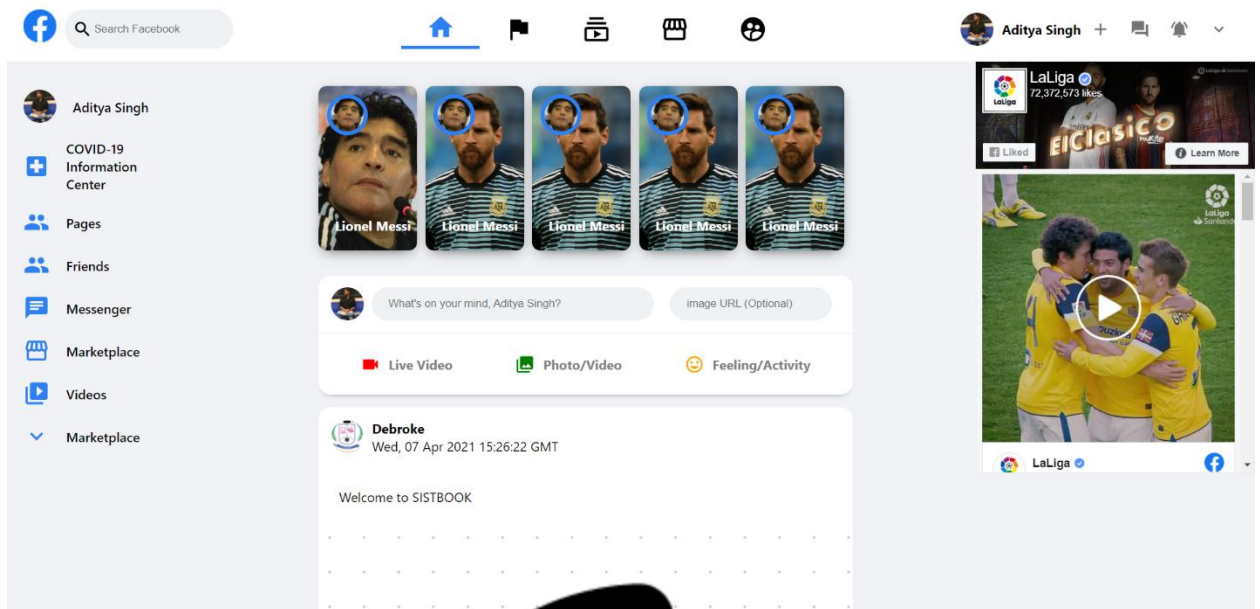
export default Sidebar;

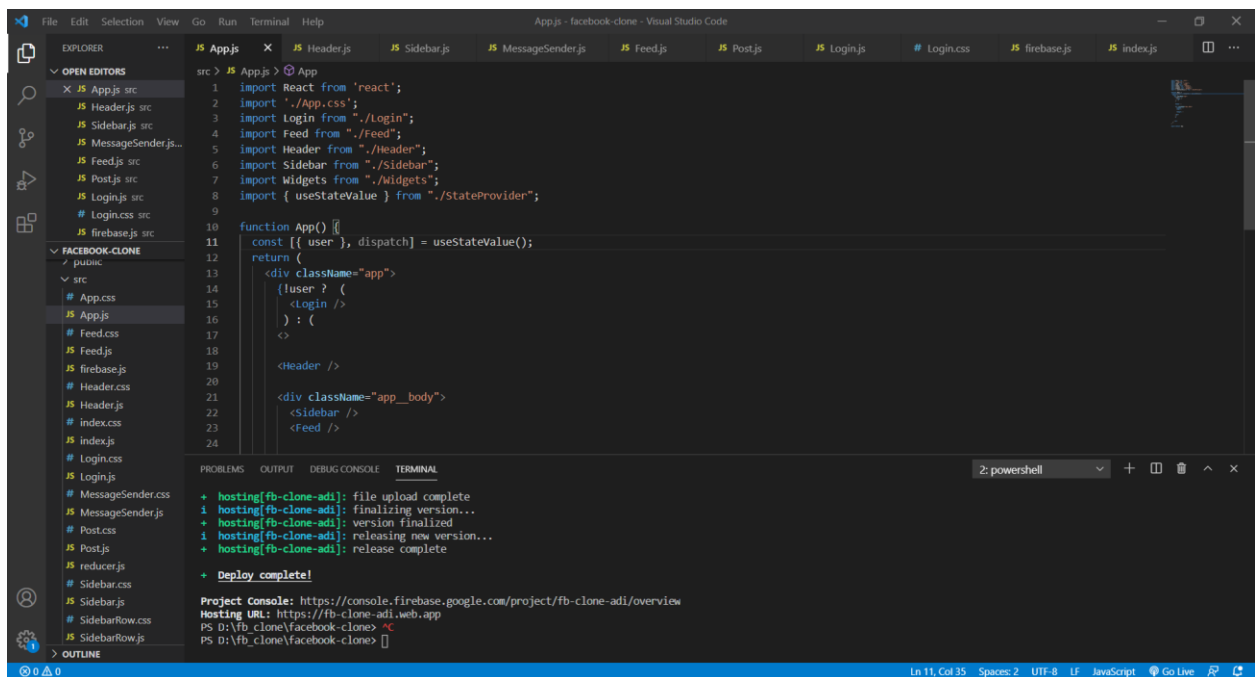
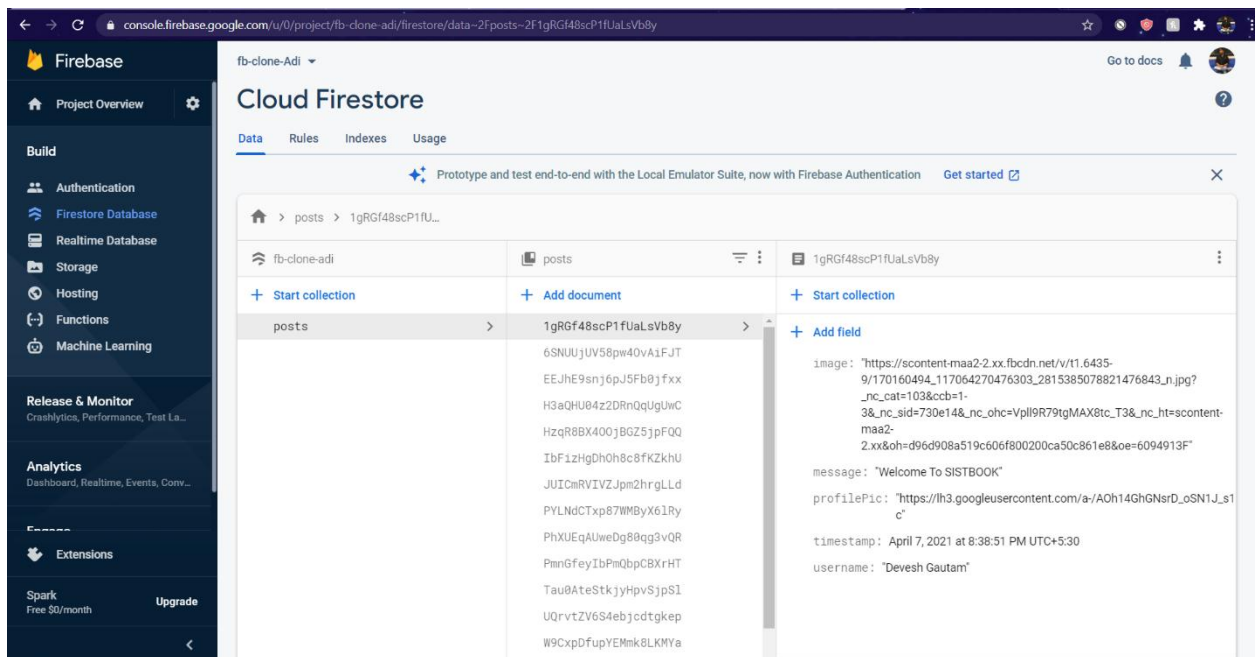
```

B. SCREENHOTS

```
1 import firebase from 'firebase';
2 const firebaseConfig = {
3   apiKey: "AIzaSyB2MjwDcsQpvc8I8mk51bjPzJQq1Jwkw",
4   authDomain: "fb-clone-adi.firebaseio.com",
5   projectId: "fb-clone-adi",
6   storageBucket: "fb-clone-adi.appspot.com",
7   messagingSenderId: "871760157031",
8   appId: "1:871760157031:web:0bac8407b37e3e3ef7c2",
9   measurementId: "G-799LWT2NP8"
10 };
11
12 const firebaseApp = firebase.initializeApp(firebaseConfig);
13 const db = firebaseApp.firestore();
14 const auth = firebase.auth();
15 const provider = new firebase.auth.GoogleAuthProvider();
16
17 export { auth, provider };
18 export default db;
```

Project Console: <https://console.firebase.google.com/project/fb-clone-adi/overview>
Hosting URL: <https://fb-clone-adi.web.app>
PS D:\fb_clone\facebook-clone>
PS D:\fb_clone\facebook-clone>





The screenshot shows the Visual Studio Code editor with the `MessageSender.js` file open. The file contains the following code:

```
src > JS MessageSender.js > | MessageSender | handleSubmit
1 import { Avatar } from "@material-ui/core";
2 import React, { useState } from "react";
3 import "./MessageSender.css";
4 import VideocamIcon from "@material-ui/icons/Videocam";
5 import PhotoLibraryIcon from "@material-ui/icons/PhotoLibrary";
6 import InsertEmoticonIcon from "@material-ui/icons/InsertEmoticon";
7 import { useStateValue } from "./stateProvider";
8 import db from "./firebase";
9 import firebase from "firebase";
10
11 function MessageSender() {
12   const [{ user }, dispatch ] = useStateValue();
13   const [input, setInput] = useState("");
14   const [imageUrl, setImageUrl] = useState("");
15
16   const handleSubmit = (e) => {
17     e.preventDefault();
18
19     db.collection('posts').add({
20       message: input,
21       timestamp: firebase.firestore.FieldValue.serverTimestamp(),
22       profilePic: user.photoURL,
23     });
24   }
25 }
```

The terminal output shows the following messages:

```
+ hosting[fb-clone-adi]: file upload complete
+ hosting[fb-clone-adi]: finalizing version...
+ hosting[fb-clone-adi]: version finalized
+ hosting[fb-clone-adi]: releasing new version...
+ hosting[fb-clone-adi]: release complete

+ Deploy complete!

Project Console: https://console.firebase.google.com/project/fb-clone-adi/overview
Hosting URL: https://fb-clone-adi.web.app
PS D:\fb_clone\facebook-clone>
PS D:\fb_clone\facebook-clone>
```

The screenshot shows the Visual Studio Code editor with the `StateProvider.js` file open. The file contains the following code:

```
src > JS StateProvider.js > ...
1 import React, { createContext, useContext, useReducer } from "react";
2
3 export const StateContext = createContext();
4
5 export const StateProvider = ({
6   reducer, initialState, children
7 }) => (
8   <StateContext.Provider value={useReducer(reducer, initialState)}>
9     {children}
10   </StateContext.Provider>
11 );
12
13 export const useStateValue = () => useContext(StateContext);
```

The terminal output shows the following messages:

```
+ hosting[fb-clone-adi]: file upload complete
+ hosting[fb-clone-adi]: finalizing version...
+ hosting[fb-clone-adi]: version finalized
+ hosting[fb-clone-adi]: releasing new version...
+ hosting[fb-clone-adi]: release complete

+ Deploy complete!

Project Console: https://console.firebase.google.com/project/fb-clone-adi/overview
Hosting URL: https://fb-clone-adi.web.app
PS D:\fb_clone\facebook-clone>
PS D:\fb_clone\facebook-clone>
```