

YouTube Clone

Project report submitted in partial fulfillment of the requirement for the
degree of Bachelor of Technology

in

Computer Science and Engineering/Information Technology

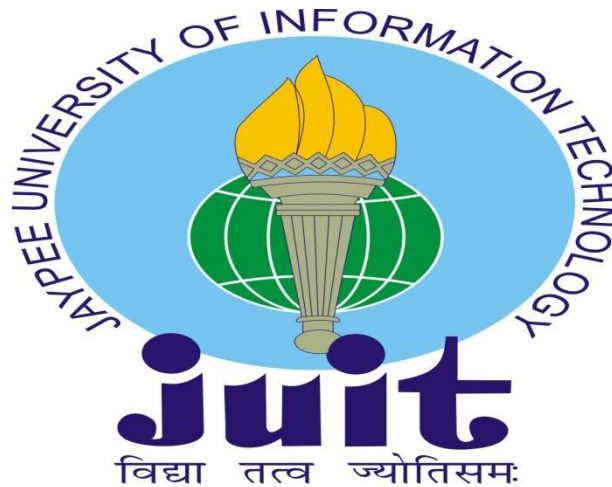
By

(Radhika Gupta (191439))

Under the supervision of

(Dr. Rajni Mohana)

to



Department of Computer Science & Engineering and Information
Technology

**Jaypee University of Information Technology Waknaghat, Solan-
173234, Himachal Pradesh**

Candidate's Declaration

I hereby declare that the work presented in this report entitled “ **YouTube Clone**” in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering/Information Technology** submitted in the department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from July 2022 to May 2023 under the supervision of **Dr. Rajni Mohana** , Associate Professor , Computer Science & Engineering and Information Technology.

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Radhika Gupta

191439

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Dr. Rajni Mohana

Associate Professor

Computer Science & Information Technology

Dated: 05-05-2023

ACKNOWLEDGEMENT

Firstly, I express my heartiest thanks and gratefulness to almighty God for his divine blessing that made it possible to complete the project work successfully.

I am quite grateful to my supervisor, Dr. Rajni Mohana, Associate Professor, Department of CSE Jaypee University of Information Technology, Wagnaghat, for her assistance. To complete this assignment, my supervisor has extensive knowledge and a deep interest in the subject of Web Development. Her never-ending patience, intellectual direction, constant encouragement, constant and energetic supervision, constructive criticism, good suggestions, and reading many poor versions and fixing them at all stages made it possible to finish this job.

I wish to express my sincere gratitude to my manager Mr. Divneet Singh, for providing me an opportunity to do my internship and project work in Paxcom. I would like to thank all my team members and seniors in Paxcom who helped me in every aspect of this project.

I would also like to express my gratitude to everyone who has directly or indirectly assisted me in making this project a success. In this unique scenario, I'd want to appreciate the different staff members, both teaching and non-teaching, who have developed their helpful assistance and facilitated my project. Finally, I must express my gratitude for my parents' unwavering support and patience.

Finally, I must acknowledge with due respect the constant support and patients of my parents.

Radhika Gupta

(191439)

Table of Content

Content	Page No.
CANDIDATE'S DECLARATION	i
ACKNOWLEDGEMENT	ii
LIST OF ABBREVIATION	iv
LIST OF FIGURES	v
ABSTRACT	vi
Chapter-1 INTRODUCTION	1-20
1.1 INTRODUCTION	1
1.2 OBJECTIVE	2
1.3 METHODOLOGY	3
1.4 ORGANIZATION	17
Chapter-2 LITERATURE SURVEY	21-22
Chapter-3 SYSTEM DEVELOPMENT	23-38
Chapter-4 PERFORMANCE ANALYSIS	39-40
Chapter-5 CONCLUSIONS	41-42
5.1 CONCLUSIONS	41
5.2 FUTURE SCOPE	41
REFERENCE	43

List of Abbreviations

- | | | |
|---------|---|----------------------------|
| 1. JS | - | JavaScript |
| 2. CRUD | - | Create Read Update Delete |
| 3. JWT | - | JSON Web Token |
| 4. JSON | - | JavaScript Object Notation |
| 5. NPX | - | Node Package eXecute |
| 6. NPM | - | Node Package Manager |
| 7. API | - | Application |
| 8. HTML | - | Hypertext |
| 9. CSS | - | Cascading Style Sheet |
| 10. GCP | - | Google |
| 11. DOM | - | Document Object Model |
| 12. UI | - | User Interface |

List of Figures

Figure 1.1: Working of Virtual DOM	6
Figure 1.2: Working of Virtual DOM	6
Figure 1.3: Working of Node.js	11
Figure 1.4: Journey of Paxcom	19
Figure 3.1: System Design	24
Figure 3.2: Connection to MongoDB	26
Figure 3.3: MongoDB Connected	26
Figure 3.4: To SignUp The New User	27
Figure 3.5: SignIn the User	28
Figure 3.6: Node.js Server	28
Figure 3.7: To fetch the videos	29
Figure 3.8: To render the video on the browser	29
Figure 3.9: Functionality to like the video	30
Figure 3.10: Functionality to dislike the video	30
Figure 3.11: Subscription of Channel	31
Figure 3.12: Upload the details of project	31
Figure 3.13: To handle the change	31
Figure 3.14: To upload file	32
Figure 3.15: For changing the state	32
Figure 3.16: Error when state is not changed	32
Figure 3.17: Hierarchy of Components	33
Figure 3.18: Home Component	34
Figure 3.19: Search Page Component	35
Figure 3.20: WatchScreen Component	36
Figure 4.1: App.js	40
Figure 4.2: SearchPage.js	40
Figure 4.3: VideoPlayer.js	41

Abstract

YouTube is one of the most popular video-sharing platform that allow the users with the ability to view, search recommended videos, upload, create and manage their own channels etc. The YouTube platform includes a user-friendly interface, allowing users to easily upload and view the popular or demanded videos.

This abstract shows the front end of the web application and the API used to build the dynamic web application. The front end of the application was developed using the JavaScript language that required React library for developing user interface based components. React was used as it allows to build the reusable user interface components. To fetch the details and the statistics of the video, channel etc. YouTube v3 API was used.

The project also included static data inclusion using the json file. It required to add the details of the youtube video manually. The project focused on front end development using the reusable user interface components.

The future scope of the project is to add the backend layer so that any changes done on the video in clone are added and changed on the given video. We have to add security layer to the project so that no one knows from which server the data is being fetched. This will block the user from sending multiple requests to the same server thus leading to it's failure.

CHAPTER-1

INTRODUCTION

1.1 Introduction

The YouTube clone project is a software development project that aims to recreate the functionality and user experience of YouTube, but with a focus on using different and new programming languages and frameworks to build the platform.

YouTube is a popular video streaming platform that has revolutionized the way people consume video content. It allows users to upload, view, and share videos on a wide range of topics, including music, education, entertainment, and more. With over 2 billion monthly active users and over a billion hours of video watched daily, YouTube has become one of the most significant platforms for video content on the internet. The platform has also introduced features such as live streaming, video recommendations, and a wide range of user-generated content, making it an essential platform for content creators, businesses, and individuals looking to share their ideas, knowledge, and creativity with a global audience. YouTube's vast library of video content and its easy-to-use interface make it a powerful tool for entertainment, education, and communication, making it an indispensable part of our digital lives.

The project focuses to build a basic YouTube clone in ReactJs. The project used YouTube API to get the videos. These videos are displayed to the user or developer. It allowed to display all the details of the videos just as the original YouTube displays. This project allows you to search any video you want to watch.

To fetch the real time data from google's YouTube database this project used google's YouTube v3 API. The results worked exactly like the real YouTube works. YouTube API made it easy to show features that were added to the application. The API returns JSON data as result on receiving the HTTP request from YouTube v3. The YouTube API enables the developer to manage the videos, upload videos, and many more. The API supports searches for any video, channel, playlist etc.

The YouTube clone project works as a user interface and displays the data that was fetched from YouTube database. It allows you to search any kind of video ranging from Hollywood to

blogs. This application provides the video that matches the searched keyword and also provide more suggestion or recommendations for similar videos or the searched keyword.

The YouTube clone project is a challenging but exciting endeavor that requires expertise in various programming languages and frameworks. The project requires developers to have a good understanding of ReactJS, which is a popular front-end library that allows for the creation of complex and interactive user interfaces. Additionally, it is essential to have a thorough understanding of the YouTube API and how it can be leveraged to access the vast amount of data that is available on YouTube.

One of the most critical aspects of the YouTube clone project is the user interface. The user interface needs to be intuitive, easy to navigate, and provide an experience that is similar to the original YouTube platform. The user interface must be designed to allow users to search for videos, view video details, and watch videos seamlessly. Additionally, the user interface needs to be optimized for various devices, such as desktops, tablets, and mobile devices.

The YouTube clone project can also provide developers with an opportunity to learn about modern web development concepts, such as serverless computing and cloud services. The project can be built using serverless architectures, which eliminates the need for traditional servers and allows for more scalable, cost-effective, and reliable applications. Additionally, cloud services such as Amazon Web Services (AWS) or Google Cloud Platform (GCP) can be used to host the application and provide access to various cloud-based services, such as machine learning, analytics, and security.

Overall, the YouTube clone project is an exciting and challenging software development project that requires a good understanding of ReactJS and the YouTube API. The project can provide developers with an opportunity to learn about modern web development concepts and create a high-quality user interface that provides a similar experience to the original YouTube platform. By leveraging modern technologies such as serverless computing and cloud services, the project can be scalable, cost-effective, and reliable.

1.2 Objective

- To gain experience of new programming languages and technologies.
- Explore the challenges and complexities of building a large-scale web application.
- By building a video sharing platform with the features and functionality of YouTube, developers can gain valuable experience in areas such as web development, API, user interface design, database management, front-end development.
- To get introduced to frontend and backend for the development of web application.
- The project provides an opportunity to learn new skills, explore complex challenges, and build a real-world application that can be used as a portfolio piece or as a starting point for future projects.

1.3 Methodology

Problem statement

We need to build a web application that will allow the user to view and search videos from this YouTube clone similar to original YouTube application. This application should have user friendly interface that will allow the user to interact with this application easily. Various features that should be supported by this application are search bar, video player, comments under the video, recommended videos, and subscriptions and other features that are supported by the YouTube application.

This application should allow the user to search the videos according to the keyword entered by the user and should be able to filter the results by published date, popularity and duration. The video that is imported from YouTube API should be allowed with several features like play, pause, full screen, and other more features. The recommended video component should display the videos according to viewers interest.

Users should be able to like and dislike the videos which should be updated in the backend. The users should be allowed to comment under the video comment section. The user should also be allowed to subscribe to the channel and add the subscribed video to the subscription portion in the sidebar component.

The project should be developed using React, Node.js and the YouTube API. Create the backend API that will handle authentication, video data and user data. After completing the frontend and backend the application should be deployed on cloud hosting services.

Technologies and Languages used:

For creating the YouTube clone project we have used various technologies for frontend and backend.

Frontend:

To design the frontend of the project the technologies used were:

- JavaScript

JavaScript is the language that is used for client-side programming in the web development. This language helps the developer to add dynamic behaviour and interactivity to the web pages that are to be designed by the user. [1]

JavaScript is object-oriented programming language. JavaScript provides the developer with various Application Programming Interfaces also abbreviated as APIs to work with regular expression, some of the standard data structures, dates, texts and also with Document Object Model that is abbreviated as DOM. [1]

JavaScript provides the developers to create interactive websites that can change the web page content as well as the styles dynamically. The best part about JavaScript is that it provides frameworks that add more functionality to the web applications. [1]

The frameworks are ReactJS, ReactNative, Node.js etc. JavaScript even provides some functions so that web applications could interact with users. Some of the functions are alert, prompt and confirm. When we submit anything on the web then sometimes we get a prompt or alert from the web application. So, this is possible because of JavaScript functionality. [2]

The arrow functions that will be used in our project are part of javascript. The arrow functions allow the developer to write the shorter function syntax. These functions are used when developers want to access property of this, inside the callback. [2]

- React

React is a JavaScript library that helps developer in building user interfaces for mobile applications as well as for web applications. React creates a reusable user interface which is the best part about it as developer can use the same component again and again without creating it again in another component. It helps the developer to build the web applications efficiently. [3]

The developers cannot use react directly. To build a react project developer needs to install some environments and packages like NodeJS. [3]

React was created by Facebook by SE Jordan Walke. [3]

As React is flexible, reusable so it is preferred by the developers to build the complex and interactive applications. [3]

React is not a framework but library of JavaScript. It uses a basic syntax for the development of web applications that is JSX. JSX basically allows the developer to write HTML in react. JSX is the extension in React that allows the developer to write JavaScript code that looks like code in HTML. JSX makes it easier for the developers to maintain the complex code and write a complex code for developing applications like YouTube clone. [3]

React also has the ability to handle and work with state and props. Properties in react are called as props that are passed from their parent component to another component. The data that is specific to a component are known as state in react and these can be changed over time. These features of react help the developer to make create the responsive and interactive web pages or the user interfaces. The user interface can get updated in real time depending on the user interactions with data. [4]

Another feature of react is virtual document object model (DOM). This is the representation of actual document object model. The virtual DOM is lightweight representation of real DOM. It allows react to update the user interface efficiently. It re-renders that parts of user interface that were changes and does not re-renders the entire user interface. This feature makes react different from other user interface libraries. [4]

Below images shows how virtual DOM works.

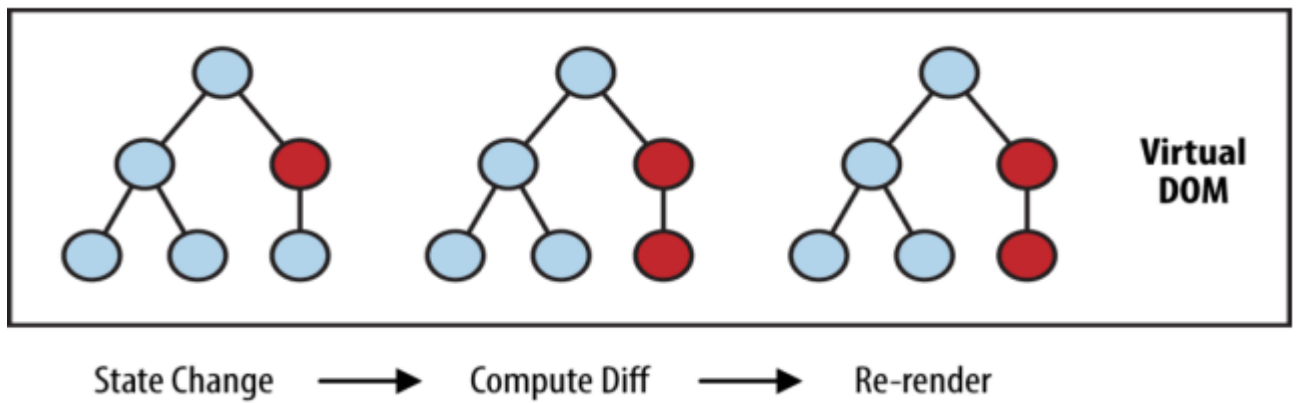


Figure 1.1: Working of Virtual DOM

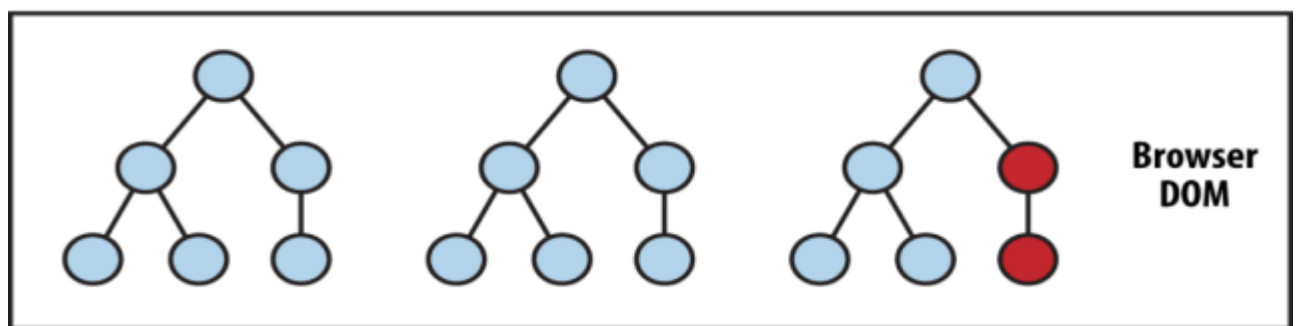


Figure 1.2: Working of Virtual DOM

React also supports react router that is used to manage routing. Routing is the method of navigating between the different components of react or different pages within a web or mobile application. React Router needs various components so that it could navigate between the pages or could manage the routes in react application. Various components that were used in the YouTube clone project of react router are Link, Route, Switch, BrowserRouter, NavLink, Redirect. [4]

The top level component of react router is BrowserRouter. BrowserRouter ensures the browser-based routing. BrowserRouter listens to URL changes which are made by user interactions. On the basis of current URL, BrowserRouter render the components to the web application. [4]

Route defines the route and when the route is matched that particular component is rendered to the web application. It accepts strings as well as query as part of URL.

To create the clickable links that will allow the user to navigate to routes of application React Router provide with Link component as well as with NavLink component. NavLink provides styleLink option or features for the active links. [4]

First route that matches the current URL is rendered using the switch component. Routes with varying level of specificity are defined with switch component.

When the current URL matches a specific route redirect component redirects the user to a different URL.

To manage the state, react supports the library redux. Redux is basically state management library. There are three main parts of redux namely the store, the actions and the reducers.

Store holds the state tree of the application and is considered as single source of truth for the application state. Store also provide the different methods to update the state tree.

Actions represent the intention to change the state. Actions are the javascript objects. Actions have some payload data and have property named as type.

Reducers are considered as functions that take the present state and action and return the new state. Reducers work on the actions that are dispatched to the store and specify the way state tree transforms on the corresponding action. [4]

Redux manages the application state like current video being played, search results and many more. Redux is used with react by connecting the components of react to redux store. It is connected to react using the connect function that is available in the react redux library.

It provides centralized way to manage the application state and maintain it. [4]

- HTML

HTML is used to create the web pages for the web application. To define the content and layout of web pages HTML provides the structure and syntax to them. These web pages are then rendered by the web browser. [5]

To make the user interface HTML is used to define the basic structure of web pages. It provides the layout of header, search bar , video player and many more. The tags used in the YouTube clone project are used to define various parts of the web pages like headings, paragraphs, lists, links etc. These elements are defined using opening and closing tags. [5]

HTML is the fundamental part of web development that is important for creating the user interface for the web applications.

The word hypertext in the HTML proposes the link between the web pages and the word markup language in the HTML proposes the text document that is within the tags.

- CSS

This is style sheet language that is used to add style and layout to HTML documents. It makes the web application's user interface visually pleasing. It adds different colours, fonts, padding, margins to the web application. [6]

CSS adds these styling properties by different selectors like element selectors, class selectors and the ID selectors. These styles can be defined in a separate CSS file or within the HTML file and these are known as external and internal styling respectively.[6]

CSS basically simplifies the process of making web pages more presentable to the users. It provides the powerful control over the presentation of the web pages.

CSS has much more wider array of attributes than that are there in HTML. So, CSS attributes helps the document to look more presentable in comparison to HTML attributes.

To apply one common property to more than one element it is required to use CSS rather than HTML. To apply the properties with HTML it will take to add those properties in every element again and again but if these properties are added with CSS then we have to write them once and can use them again and again. Thus, we can reuse the same sheet again and again.[6]

- Material UI

Material UI consists of pre-designed components that follow the design language. The design language that is followed by Material UI is based on Google's Material Design language. [7]

Material UI has many components such as buttons, icons, cards, dialogs, inputs and many more. These components help to create visually appealing user interface across the overall application. These components are easy to use and customize. These components can be used using the import command from their respective libraries. These components can also be customized using react and various properties and can be made useable for the users.[7]

Material UI provides various set of utilities that customize the styles, manage them and customize the theme of the web application.

Material UI components are designed in such a way that they work very well on a variety of screen devices with varying sizes. This helps the user to make mobile application too.

Material UI help to speed up the development and create more visually appealing interface for the user.

Backend:

To build the backend for the project technologies required:

- Node.js

Node.js is cross platform, open source runtime environment. Node.js is server side JavaScript runtime environment. Node.js is built on the Chrome's V8 JavaScript engine. To build scalable network application Node.js is widely used by the developers. Node.js was released by Ryan Dahl in the year 2009. For building the real-time, data-intensive and scalable applications, Node.js has a non-blocking, event-driven input/output model which makes it suitable for building such type of web applications.[8]

There are many advantages of Node.js but one of its important advantage is that it can handle very large number of connections that are running concurrently with a small number of threads. As it was explained in the above paragraph that Node.js is event-driven, non-blocking I/O model. So above advantage of Node.js is achieved because of this feature of Node.js. It means

that when a request is received the Node.js does not block the thread that is waiting for the input/output operations to get complete.[8]

Node.js assigns the input/output operation to a separate thread that allow the thread to continue the processing for other requests. The processing of the request is resumed when the input/output operation is complete and Node.js triggers the callback function.[8]

Node.js makes the building of scalable network applications very easy as it has many built-in modules that supports the Node.js to do so. The various modules that Node.js has built-in or supported by it are http, https, fs, path, querystring. To work with many more modules there are various third party modules that are supported by node.js. These modules that are supported by Node.js are provided by the node package manager also known as npm. The functionality that are provided by these modules are web frameworks, authorization and authentication and database access.[8]

It is very easy to make the data-intensive applications, real-time applications using Node.js. To build the applications that require real-time data updates use Node.js for their development. Node.js allow the technologies of server-sent events to push the data to client in real-time that provide a seamless and responsive experience to the user.[8]

Node.js is not only available for the server-side runtime environment but it can also be used for building the command-line tools and other applications like desktop applications. For building the desktop applications other frameworks like Electron in conjunction with Node.js that allows the developers to build cross-platform applications using the web technologies like HTML, CSS and Javascript.[8]

So Node.js is a powerful and flexible technology for building real-time, data-intensive and real-time network applications. Node.js non-blocking, number of built-in modules, real-time and large number of third-party modules that make Node.js a popular choice of developers for building the applications from simple applications to real-time complex systems. [8]

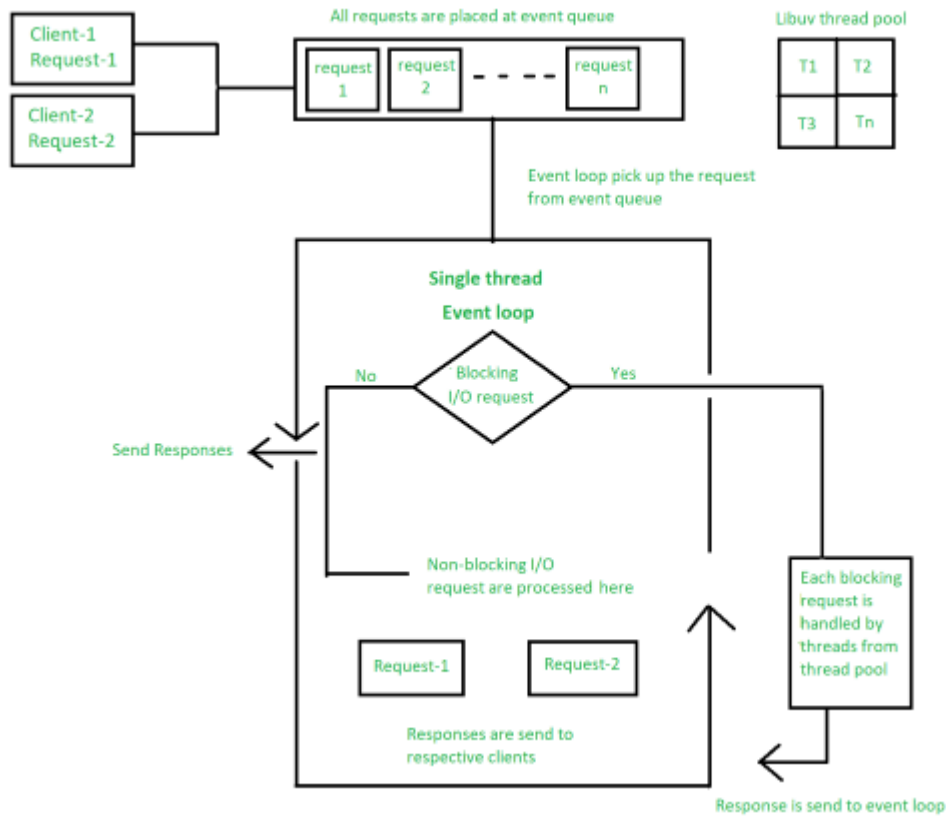


Figure 1.3: Working of Node.js

- Express

Express.js is a popular, open-source, fast and flexible web application framework for Node.js. It was created in 2010 by TJ Holowaychuk and is now maintained by a large community of developers. Express.js is designed to simplify the process of building web applications, particularly RESTful APIs, by providing a simple and consistent API for handling HTTP requests and responses.[9]

Express.js is built on top of Node.js and provides a layer of abstraction over Node.js's built-in http module. It provides a set of middleware functions that can be used to handle various stages of the request-response cycle, such as logging, parsing request bodies, handling cookies, and error handling. These middleware functions can be chained together to create powerful and flexible middleware pipelines. [9]

One of the key features of Express.js is its routing capabilities. Express.js provides a simple and flexible way to define routes for handling HTTP requests. Routes can be defined using a

combination of HTTP methods, URL patterns, and middleware functions. This makes it easy to build RESTful APIs that follow the principles of resource-oriented architecture. [9]

Express.js also provides a robust set of tools for working with views and templates. It supports a wide range of templating engines, including Pug, EJS, and Handlebars, allowing developers to choose the one that best suits their needs. Express.js provides a set of conventions and helpers for working with views and templates, such as layout templates, partials, and helpers. [9]

Express.js is highly extensible, and its flexibility has led to the development of a vast ecosystem of third-party middleware and plugins. There are many community-contributed modules available through the npm registry, such as authentication middleware, database connectors, and logging middleware. This means that developers can quickly and easily add additional functionality to their Express.js applications without having to reinvent the wheel. [9]

Overall, Express.js is a powerful and flexible web application framework that simplifies the process of building web applications, particularly RESTful APIs. Its middleware pipeline, routing capabilities, and support for views and templates make it a popular choice for building web applications of all sizes and complexities. Its extensibility and large ecosystem of third-party middleware and plugins make it easy for developers to add additional functionality to their applications. [9]

- MongoDB

MongoDB is a popular NoSQL database that is designed to store and manage large volumes of data in a highly scalable and flexible manner. It was first released in 2009 and has quickly gained popularity due to its ease of use, flexibility, and scalability. [10]

MongoDB is a document-oriented database, which means that it stores data in JSON-like documents that can be easily accessed and manipulated using a variety of programming languages. Each document in MongoDB can have its own structure, which allows developers to store complex and heterogeneous data in a single database. [10]

One of the key features of MongoDB is its ability to scale horizontally across multiple nodes. This allows developers to easily scale their databases to handle large volumes of data and high

traffic loads. MongoDB also supports automatic sharding, which means that it can distribute data across multiple servers without requiring any intervention from the developer. [10]

MongoDB supports a wide range of data types, including strings, numbers, booleans, arrays, and objects. It also supports a range of advanced data types, such as geospatial data, dates, and regular expressions. This makes it a powerful tool for handling a wide range of data types and use cases.

Another key feature of MongoDB is its flexible querying capabilities. MongoDB provides a powerful query language that allows developers to retrieve data based on a wide range of criteria, including text search, geospatial queries, and complex aggregation pipelines. This makes it easy for developers to retrieve the data they need, regardless of how it is structured or stored. [10]

MongoDB also provides a range of tools for data visualization, management, and administration. It has a web-based administration tool called MongoDB Compass, which allows developers to easily explore and manipulate their data. MongoDB also provides drivers for a wide range of programming languages, making it easy to integrate MongoDB with existing applications. [10]

Overall, MongoDB is a powerful and flexible NoSQL database that is designed to handle large volumes of data in a highly scalable and flexible manner. Its support for document-oriented data storage, horizontal scaling, and flexible querying make it a popular choice for building modern web applications and data-intensive applications. [10]

- **Firestore**

Firestore is a popular mobile and web application development platform that offers a range of tools and services to help developers build high-quality, scalable applications. It was first launched in 2011 by Firebase, Inc., and was acquired by Google in 2014.[11]

Firestore provides a wide range of features and services, including cloud-based databases, authentication and authorization tools, analytics and monitoring tools, cloud messaging services, hosting and deployment services, and more. One of the key benefits of Firestore is that

it is designed to be easy to use and integrate with existing applications, making it a popular choice for developers of all skill levels.[11]

Firebase provides several different database options, including a real-time database and a cloud-based Firestore database. These databases are designed to be highly scalable and flexible, making it easy to store and manage large volumes of data. Firebase also provides authentication and authorization tools, which allow developers to easily add user authentication and authorization to their applications. [11]

Another key feature of Firebase is its analytics and monitoring tools, which allow developers to track user engagement and behaviour, monitor application performance, and identify and troubleshoot issues. Firebase also provides cloud messaging services, which allow developers to send notifications and messages to users across multiple platforms and devices. [11]

Firebase provides hosting and deployment services, which allow developers to quickly and easily deploy their applications to the cloud. Firebase also integrates with a wide range of other tools and services, including popular front-end frameworks like React and Angular, as well as other Google services like Google Cloud Storage and Google Maps. [11]

Overall, Firebase is a powerful and versatile platform that offers a range of tools and services to help developers build high-quality, scalable applications. Its ease of use, flexibility, and integration with other tools and services make it a popular choice for developers looking to build modern mobile and web applications. [11]

- **CRUD**

CRUD stands for Create, Read, Update, and Delete, which are the four basic operations that can be performed on data in a database. In the context of a YouTube clone, CRUD operations can be used to manage various aspects of the platform, such as user accounts, videos, comments, and playlists.[12]

Here's a detailed description of how CRUD can be used in a YouTube clone:

1. Create: This operation is used to add new data to the database. In the case of a YouTube clone, users can create new accounts, upload new videos, and create new comments or playlists. When a user creates a new account, their information (such as username, email, and password) is added to the database. When a user uploads a new video, the video file is stored on the server and information about the video (such as title, description, and tags) is added to the database. When a user creates a new comment or playlist, information about the comment or playlist is added to the database.[12]

2. Read: This operation is used to retrieve data from the database. In the case of a YouTube clone, users can read information about their own accounts, videos, comments, and playlists, as well as information about other users' accounts, videos, comments, and playlists. When a user wants to view their own account information, the database is queried to retrieve their username, email, and other details. When a user wants to view a video, the database is queried to retrieve information about the video, such as the title, description, and tags. When a user wants to view comments or playlists, the database is queried to retrieve information about those items. [12]

3. Update: This operation is used to modify existing data in the database. In the case of a YouTube clone, users can update their own account information, modify the details of their own videos, and edit or delete their own comments or playlists. When a user wants to update their account information, the database is updated with the new information. When a user wants to modify the details of a video, the database is updated with the new information (such as a new title, description, or tags). When a user wants to edit or delete a comment or playlist, the database is updated to reflect those changes. [12]

4. Delete: This operation is used to remove data from the database. In the case of a YouTube clone, users can delete their own accounts, videos, comments, and playlists. When a user deletes their account, all of their information is removed from the database. When a user deletes a video, the video file is deleted from the server and all associated information (such as the title, description, and tags) is removed from the database. When a user deletes a comment or playlist, the associated information is removed from the database. [12]

Overall, CRUD operations are essential for managing the data on a YouTube clone platform. By using these operations, users can create, read, update, and delete various types of data, which allows them to interact with the platform in a meaningful way.

- Dotenv

To manage the environment variables in Node.js application dotenv is used which is a package in node.js package. To store the sensitive information like API keys, database passwords are stored in dotenv. It disables the developers to not hard code the sensitive information in the code but to add credentials as environment variables.

To use dotenv in the project developer need to install it using `npm install dotenv` command. By just installing it we cannot use it in the application but we have to configure this also. This will allow to use the environment variables to the application.

To use this environment variables in the application we need to use the command `process.env` object.

- JWT

JWT (JSON Web Token) cookie authentication is a way to authenticate users in web applications using JWTs and cookies. This approach is commonly used in modern web applications because it offers several advantages over other authentication methods.[13]

Here's how JWT cookie authentication works:

1. User logs in: When a user logs in to your application, their credentials are verified and a JWT is generated on the server. This JWT contains information about the user (such as their ID and roles) and is signed with a secret key that only the server knows. [13]
2. JWT is sent to the client: Once the JWT is generated, it is sent to the client (usually in the form of a JSON response). The client then stores the JWT in a cookie. [13]
3. JWT is sent with each request: From this point forward, the client sends the JWT with each subsequent request to the server. This is usually done by including the JWT as an HTTP header or as part of the request body. [13]

4. Server verifies the JWT: When the server receives a request with a JWT, it verifies the signature of the JWT using the secret key. If the signature is valid, the server can extract the user information from the JWT and use it to authorize the request. [13]

5. Access granted or denied: Based on the user information extracted from the JWT, the server either grants or denies access to the requested resource. [13]

The use of cookies in this authentication method adds an extra layer of security. By storing the JWT in a cookie, the client does not have direct access to the JWT itself. This helps to prevent cross-site scripting (XSS) attacks, as the client cannot easily tamper with the contents of the JWT. [13]

Some advantages of JWT cookie authentication include:

- Scalability: JWTs are stateless, meaning that the server does not need to store any session data for each user. This makes it easier to scale your application horizontally.
- Security: JWTs are signed with a secret key, which makes them very difficult to tamper with or forge. This helps to prevent unauthorized access to your application.
- Simplicity: JWTs are easy to use and integrate with modern web frameworks and libraries.

Overall, JWT cookie authentication is a powerful and secure way to authenticate users in web applications. It offers many advantages over other authentication methods and is widely used in modern web development. [13]

1.4 Organization

PAXCOM

For Commerce, PAXCOM develops multi-channel processing administration services based on cloud (referred to as “Services” from here on). PAXCOM is a group of more than 400 e-commerce enthusiasts who like using technology experience and knowledge to enable the digital trade of services and products across all international marketplaces.[14]

Some of the most well-known companies in the world, like Adani, Mondelez, Britannia, HipHop skincare, Lenovo, Abbott, and Paper Boat, rely on our data, analytics, advice and marketing solutions when putting their e-commerce-focused marketing plan into practise.

Our Core Values

1. Honesty

Paxcomer values honesty and integrity above all else in our work and in the people we collaborate with. We place a great importance for honesty and are dedicated to providing realistic results.

2. Customer attraction

What we are currently accomplishing in a substantial part thanks to our customers. We consider our clients as partners who genuinely care about their success, and we put forth a lot of effort to satisfy their demands and earn their trust by exceeding their expectations.

3. Long-distance travel

We devote a lot of consideration to innovation, finding fresh opportunities, streamlining processes, and advancing our knowledge and skills. Neither to our clients nor to ourselves, we have never learned to neglect.

4. Speed

Instead than merely reacting to situations, Paxcomers believe in anticipating, palnning, preparedness and action.

5. Collaboration

As a big family, we all cooperate to meet the same objectives. We have mutual regard and trust for one another and we understand how crucial each person is for the success.

6. Build a Paxcom logo

Design a Paxcom logo, a Paxcomer acts as both a product promoter and a Paxcom agent. Each of us integrates and reflects the values we uphold in all we do, and we work to promote and strengthen the brand.

7. Have fun

According to us, work can be enjoyable and gratifying. We work hard to establish a workplace where Paxcomers can have fun and be productive at the same time.

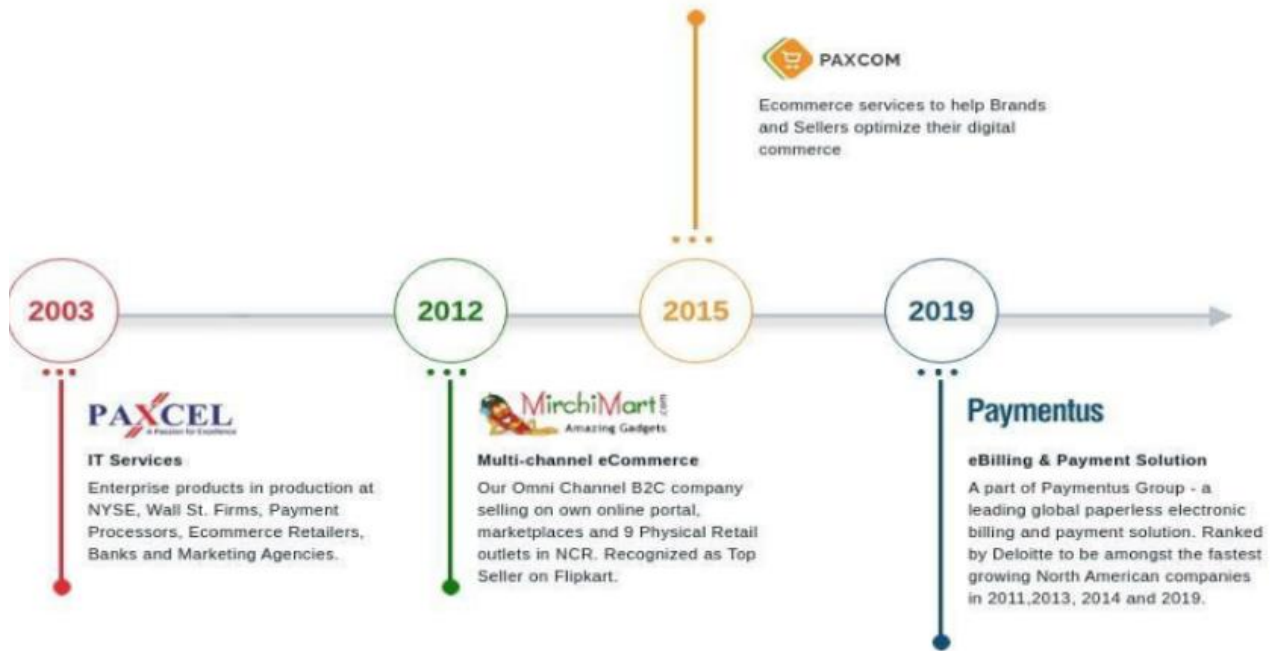


Figure 1.4: Journey of Paxcom

PAYMENTUS

The major provider of electronic payments and payment solutions worldwide, Paymentus, has Paxcom in its roster. A leading supplier of cloud-based payment technology solutions and services is Paymentus (NYSE: PAY). With the help of cutting-edge technology, we deliver our next-generation product lineup to more than 1,300 bill payers in North America. [15]

With their preferred payment channel and brand, consumers can have an easy-to-use, flexible, and secure electronic payment billing experience with our omni-channel platform. With the help of our integrated payment, billing, and reconciliation capabilities, Paymentus's unique Instant Payment Network TM, or IPN, broadens our reach by linking IPN partner forums with tens of thousands of debt collectors. Paymentus provides charges for all taxpayers in a variety of vertical industries, including services, financial services, insurance, government, communications, and healthcare. [15]

In an effort to make debt repayment more efficient, Paymentus was established in 2004. With 1,700 clients, including some of the largest taxpayers in North America, Paymentus has become the market leader in electronic payment solutions and payment solutions thanks to its vision, innovation, and exceptional service.

We are aware that we require individuals with knowledge, drive, and the capacity to champion the most thrilling customer experience in order to maintain the relevance and current status of our solutions. [15]

A secure Customer Partnership, established by SAAS and Payment Forum, has been created by our incredibly hardworking, bright staff. It allows direct billing organisations to provide integrated customer information and raises the acceptance of electronic payments and payment services. Paymentus, which has been named one of the fastest-growing companies in North America by Deloitte for four years, continually works to create platforms for payments that are better, faster, more secure, and less expensive. We are constantly seeking the greatest number of clients for our products and services. That is what caused our phenomenal growth over the last ten years. [15]

When our clients prosper, we also succeed. When their client ties are strengthened, they prosper because more of their customers use these expensive but cost-effective electrical services.

A PAYMENT TECHNOLOGY THAT IMPROVES YOUR COMPLIANCE, EFFICIENCY, KEY, AND CUSTOMER SKILLS is what makes PAYMENTUS DIFFERENT.

1. MOST FUNNY

Instead of customisation, we favour configuration. Your business needs and payment standards are satisfied by our improved, law-based engine without customisation.

2. ACKNOWLEDGMENTS

A variety of APIs, iFrames and fully customized solutions that give you complete control over your user experience are used to access our cloud-based platform.

3. NEXT

We assist companies in minimising and getting rid of the burden of PCI as a PCI Level I compliant provider.

4. SAFE AND RELIABLE

Our system is built with maximum security and zero downtime. The transaction's dependability and security are guaranteed by a multi-layer detection and blocking system, multi-level authentication, encryption, and token processing.

5. VERY CONNECTED

The CIS, accounting, and ERP systems, which enable smooth payments, reporting, and reconciliation to important financial systems and operations, are just a few of the more than 350 major programmes that we cover.

6. BUILT FOR THE FUTURE

It is completely new construction. We can continuously innovate and remain ahead of the always changing needs since we only use one code and have no version.

CHAPTER-2

LITERATURE SURVEY

Introduction:

YouTube is the world's largest video-sharing platform with over 1.8 billion monthly active users, generating millions of views every day. The popularity of YouTube has led to the creation of several YouTube clones or video-sharing platforms that aim to replicate its success.

The YouTube Clone Project is a web application that copies or imitates the functionality of the video-sharing platform, YouTube. The project uses React for the frontend, MongoDB for the backend database, and Node.js for building the server.

YouTube was started in the year 2005 as a website that was further acquired by google in the year 2006. This video-sharing platform allows everyone to share and upload videos. It allows the world to connect with each other despite of the largest distance between them. All the data that is uploaded on YouTube are hosted on YouTube server.

The objective of YouTube is to connect people of same interest and to share and learn new things. This platform is an integrated platform that allows the user to share any kind of information from education to their personal life and so on.

The tagline of this video-sharing platform is Broadcast Yourself. YouTube is a perfect platform to gather any information related to your concern. The information is exclusively at one place thus making it easy for the user to check the information without any hustle.

YouTube is not only the video-sharing platform for the entertainment but it is also a good platform to earn money. So, platforms like youtube are a must to be built.

In recent years, there has been an increase in the popularity of web applications that are built using React, MongoDB, and Node.js. Many developers use these technologies as they offer a high degree of flexibility and scalability. Many of the studies have been done to check and evaluate the performance and usability of web applications built using the web technologies like react, node.js and mongodb.

One of such study that was conducted by Khosravi and Jamshidi in the year 2021 evaluated and checked the performance of a web application that was built using React, MongoDB, and Node.js. The study they conducted found that the application was highly responsive as it had a low response time, making it ideal for real-time web applications.

The other study that was conducted by other two people by Zhang and Chen in the year 2020 evaluated and checked the usability of web applications that were built using React, MongoDB, and Node.js. This study that was conducted by them found that the applications were easy to use and highly customizable, that made developers ideal for building complex web applications.

There are several YouTube clones and video-sharing platforms available that offer similar or unique features. Vimeo offers high-quality videos and a clean user interface, while Dailymotion offers monetization options through advertising. Twitch focuses on live streaming and community-based interaction, while Vevo focuses on music videos and related content. TikTok offers short-form videos and a range of editing tools. These platforms offer different features and cater to different audiences, and users can choose the one that best fits their needs.

Conclusion:

The YouTube Clone Project using React, MongoDB, and Node.js is a powerful and flexible web application that can be easily customized and scaled. These technologies offer several advantages, including better performance, easier debugging, and faster rendering. Several studies have shown that web applications built using these technologies are highly responsive, scalable, and easy to use. Therefore, it is a good choice for building web applications that require fast data access, scalability, and flexibility.

CHAPTER – 3

SYSTEM DESIGN & DEVELOPMENT

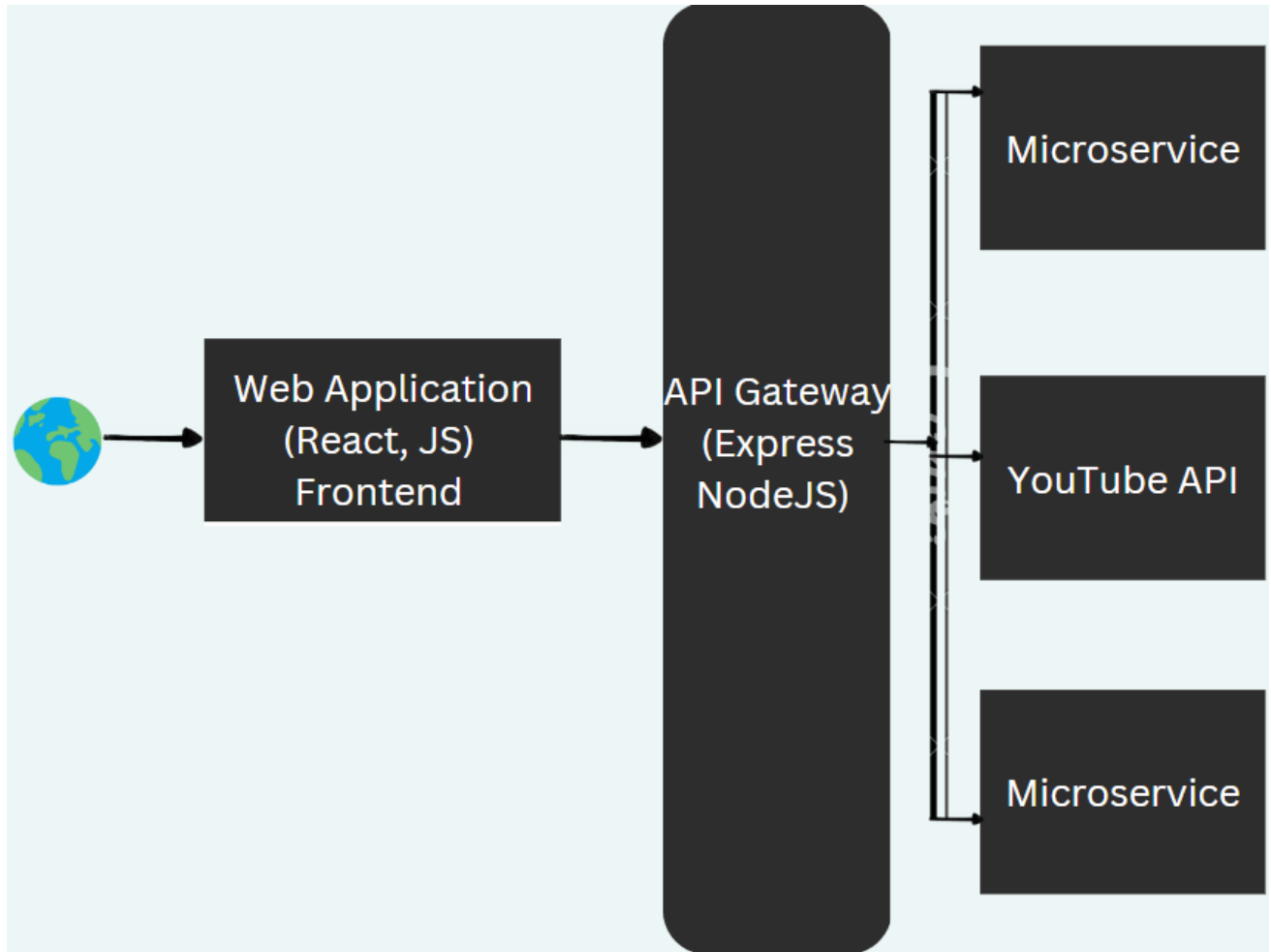


Figure 3.1: System Design

This is the system architecture of the YouTube clone where the frontend of the clone is built using javascript and react. The server file consists of express js that stores the data of YouTube videos. To render the YouTube videos dynamically we need the YouTube API to link with react application. For this purpose we need to login into google developers console where we have to enable the YouTube v3 API. It will create project and API key that will be rendered to the web application. API gateway will provide the security to the servers. The request will be sent to the API gateway first and from there to YouTube servers.

Algorithm:

It is basic algorithm to create a YouTube clone using React:

1. Create a new React project using 'create-react-app'.
2. Install the necessary packages such as material-ui, axios, and for the backend express.
3. Design the layout of the YouTube clone. Create a header component, a search bar component, and a video player component, sidebar component, video info component, sidebar row component, and recommended component.
4. Create a component for displaying search results. This component should be responsible for displaying a list of videos based on the user's search query.
5. Implement the search functionality. When the user types a query into the search bar, the application should query the YouTube API for videos matching the search query.
6. Display the search results in the search results component.
7. Implement the video player component. This component should be responsible for playing the selected video and displaying its title, description, and other relevant information.
8. Add features like comments, likes, and subscriptions.
9. Style the components using CSS.
10. Deploy the application to a hosting service like Netlify, Heroku, Firebase or Vercel.

System design

The above system design is possible when Node.js is connected to the frontend of the project as well as connected to the backend of the project like MongoDB. The backend of the project is necessary for running the application and for making it dynamic. When the user hits something on the frontend of the project or we can say the user interface of the web application than that updated component should be shown to the user and that is possible only if the project is connected to backend where all the updation will take place.

Node.js will be save in one of the file of react project in separate folder that will be run on the separate URL or localhost port. My Node.js server will run on localhost:5000.

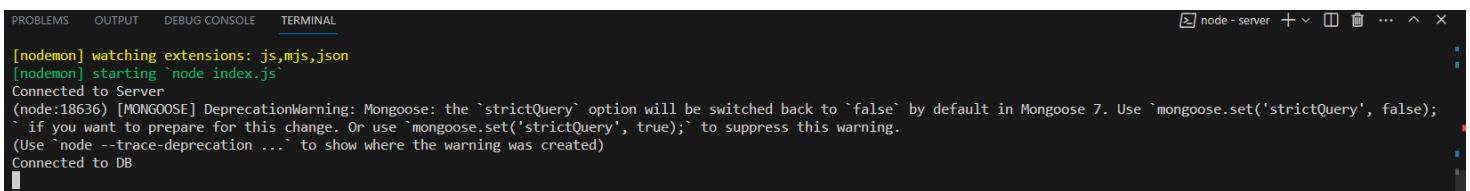
As Node.js has real-time architecture and push based, so it helps to update the application data very well.

To connect Node.js to MongoDB we need to make account in MongoDB first and then generate the username and password so that we can access that database and to allow the other users to access this database we do not need to enter our pc's IP address otherwise the user or any other person will not be able to access the application from any other device.

```
const app = express();
dotenv.config();

const connect = () => {
  mongoose
    .connect(process.env.MONGO)
    .then(() => {
      console.log("Connected to DB");
    })
    .catch((err) => {
      throw err;
    });
};
```

Figure 3.2: Connection to MongoDB

A terminal window showing the output of a Node.js application. The output includes: [nodemon] watching extensions: js,mjs,json; [nodemon] starting `node index.js`; Connected to Server; (node:18636) [MONGODB] DeprecationWarning: Mongoose: the `strictQuery` option will be switched back to `false` by default in Mongoose 7. Use `mongoose.set('strictQuery', false);` if you want to prepare for this change. Or use `mongoose.set('strictQuery', true);` to suppress this warning. (Use `node --trace-deprecation ...` to show where the warning was created); Connected to DB.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node index.js`
Connected to Server
(node:18636) [MONGODB] DeprecationWarning: Mongoose: the `strictQuery` option will be switched back to `false` by default in Mongoose 7. Use `mongoose.set('strictQuery', false);` if you want to prepare for this change. Or use `mongoose.set('strictQuery', true);` to suppress this warning.
(Use `node --trace-deprecation ...` to show where the warning was created)
Connected to DB
```

Figure 3.3: MongoDB Connected

The Rest API is created using MongoDB and Node.js All the CRUD operation for every component are stored in the controllers folder. Any updation regarding the comment done by the user, any video that will be liked or disliked by the user, all the information of the users action will be done using this file.

The routes to different paths are saved in the routes folder that is connected to the controllers folder. The API is built using the express. The routes folder and controllers folder will be able to display all together the changes that will occur in the application.

To use this application user must be authenticated by the application as it is necessary to check and verify that the identity of the user. We need authentication to prove the facts which are stated to be true by the user. Authentication needs password and username.

To sign up the user we need this thing to create a new user that will added back in the MongoDB users schema.

```
export const signup = async (req, res, next) => {
  try {
    const salt = bcrypt.genSaltSync(10);
    const hash = bcrypt.hashSync(req.body.password, salt);
    const newUser = new User({ ...req.body, password: hash });

    await newUser.save();
    res.status(200).send("User has been created!");
  } catch (err) {
    next(err);
  }
};
```

Figure 3.4: To SignUp The New User

To authenticate the user we need username and password so this will be done using the sign in option.

```

export const signin = async (req, res, next) => {
  try {
    const user = await User.findOne({ name: req.body.name });
    if (!user) return next(createError(404, "User not found!"));

    const isCorrect = await bcrypt.compare(req.body.password, user.password);

    if (!isCorrect) return next(createError(400, "Wrong Credentials!"));

    const token = jwt.sign({ id: user._id }, process.env.JWT);
    const { password, ...others } = user._doc;

    res
      .cookie("access_token", token, {
        httpOnly: true,
      })
      .status(200)
      .json(others);
  } catch (err) {
    next(err);
  }
};

```

Figure 3.5: SignIn the User

To send the information to the localhost server we will require api to do so. The file will be sent in the form of json file.

To fetch the data from Node.js API we need the MongoDB. To use this Node.js API we will store the URL in the package.json file of the client folder so that there will be no need to call this URL again and again in the other files.

```

},
"proxy": "http://localhost:6600/api/"
}

```

Figure 3.6: Node.js Server

We need to fetch the videos to our front end so that is done using various technologies like redux functions and the map function that allow the random videos to be rendered to the front end.

```

const Home = ({type}) => {
  const [videos, setVideos] = useState([]);

  useEffect(() => {
    const fetchVideos = async () => {
      const res = await axios.get(`/videos/${type}`);
      setVideos(res.data);
    };
    fetchVideos();
  }, [type]);
}

```

Figure 3.7: To fetch the videos

```

return (
  <Container>
    {videos.map((video) => (
      <Card key={video._id} video={video}/>
    ))}
  </Container>
);
}

```

Figure 3.8: To render the video on the browser

As youtube clone project is not only about creating a dummy of actual youtube but also we need to create some functionality that is given by the actual youtube also.

For this purpose we are adding the functionality of liking the video and disliking the video by the user.

For creating this functionality we will require again Redux that will have state and action property. This functionality will store the current state of the video and will update the data as soon as user hit some action the application. This is how the redux helps the developer to make the real life project work with ease.

```

like: (state, action) => {
  if (!state.currentVideo.likes.includes(action.payload)) {
    state.currentVideo.likes.push(action.payload);
    state.currentVideo.dislikes.splice(
      state.currentVideo.dislikes.findIndex(
        (userId) => userId === action.payload
      ),
      1
    );
  }
},

```

Figure 3.9: Functionality to like the video

```

dislike: (state, action) => {
  if (!state.currentVideo.dislikes.includes(action.payload)) {
    state.currentVideo.dislikes.push(action.payload);
    state.currentVideo.likes.splice(
      state.currentVideo.likes.findIndex(
        (userId) => userId === action.payload
      ),
      1
    );
  }
},

```

Figure 3.10: Functionality to dislike the video

The other functionality added to this youtube clone project is to allow the user to subscribe the channel. Again to do this redux action and state will help the developer to achieve this goal.

Initially it will store the current state of the video and as soon as the user will hit the action on the front end the state of the video will be changes that is the channel will get subscribed the user. This will change the number of subscribers of that channel in the backend that is in the MongoDB subscribers schema. This is how the Node.js and MongoDB are working simultaneously.

```

},
subscription: (state, action) => {
  if (state.currentUser.subscribedUsers.includes(action.payload)) {
    state.currentUser.subscribedUsers.splice(
      state.currentUser.subscribedUsers.findIndex(
        (channelId) => channelId === action.payload
      ),
      1
    );
  } else {
    state.currentUser.subscribedUsers.push(action.payload);
  }
}
},

```

Figure 3.11: Subscription of Channel

The other main functionality that is added in this project is of allowing the user to upload the video on to his channel. This component is easy to add using the web technologies but when the user will upload the video that video will be stored in the firebase.

```

const Upload = ({ setOpen }) => {
  const [img, setImg] = useState(undefined);
  const [video, setVideo] = useState(undefined);
  const [imgPerc, setImgPerc] = useState(0);
  const [videoPerc, setVideoPerc] = useState(0);
  const [inputs, setInputs] = useState({});
  const [tags, setTags] = useState([]);

```

Figure 3.12: Upload the details of project

```

const handleChange = (e) => {
  setInputs((prev) => {
    return { ...prev, [e.target.name]: e.target.value };
  });
};

```

Figure 3.13: To handle the change

```

const uploadFile = (file, urlType) => {
  const storage = getStorage(app);
  const fileName = new Date().getTime() + file.name;
  const storageRef = ref(storage, fileName);
  const uploadTask = uploadBytesResumable(storageRef, file);

```

Figure 3.14: To upload file

```

uploadTask.on(
  "state_changed",
  (snapshot) => {
    const progress =
      (snapshot.bytesTransferred / snapshot.totalBytes) * 100;
    urlType === "imgUrl" ? setImgPerc(Math.round(progress)) : setVideoPerc(Math.round(progress));
    switch (snapshot.state) {
      case "paused":
        console.log("Upload is paused");
        break;
      case "running":
        console.log("Upload is running");
        break;
      default:
        break;
    }
  }
),

```

Figure 3.15: For changing the state

```

    (error) => {},
    () => {
      getDownloadURL(uploadTask.snapshot.ref).then((downloadURL) => {
        setInputs((prev) => {
          return { ...prev, [urlType]: downloadURL };
        });
      });
    });

```

Figure 3.16: Error when state is not changed

Components (Hierarchy of react components):

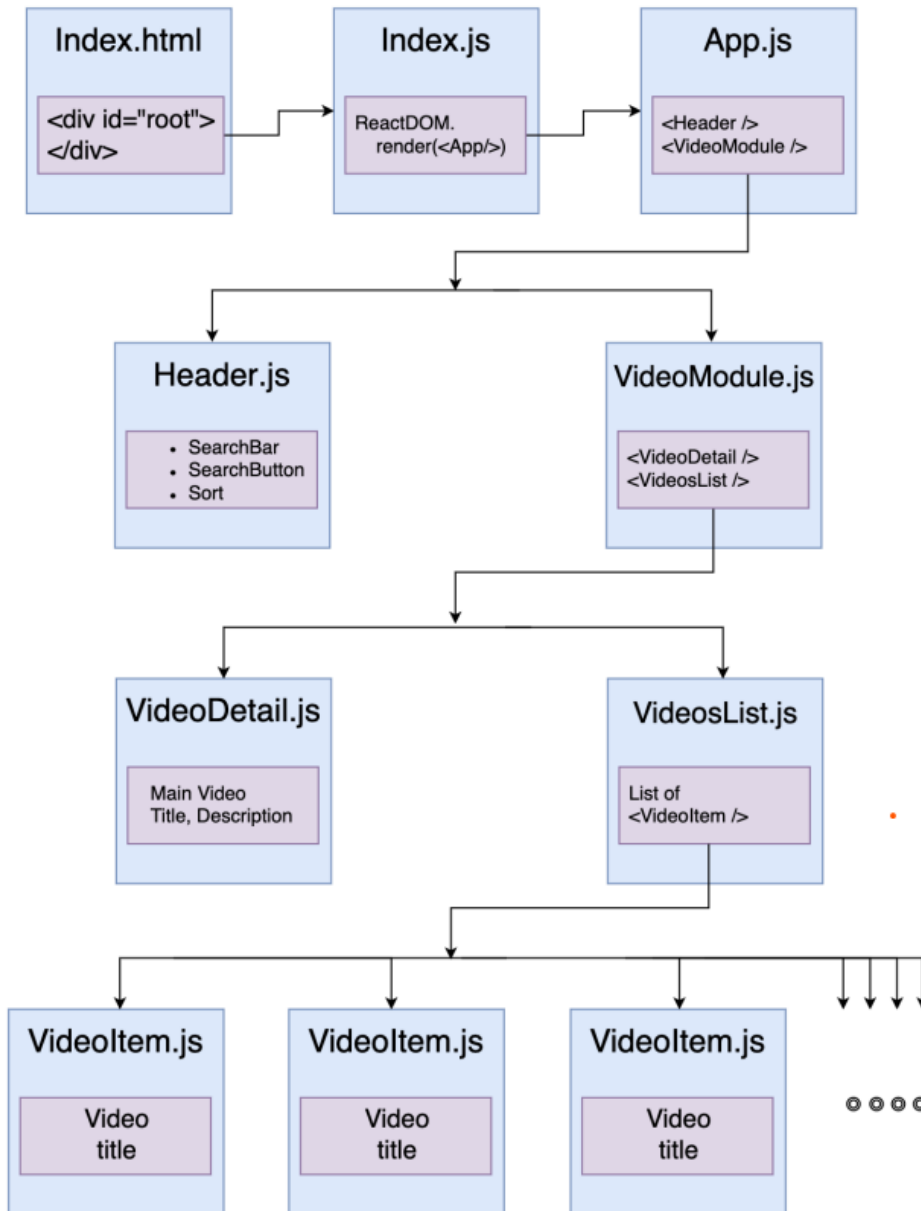


Figure 3.17: Hierarchy of Components

The main output window

The main window that consists of parent component i.e. App.js has other sub components that are also called as the child component of the App.js The parent component consists of the header component , sidebar component , video player component, search page component and recommended videos component.

The Header component consists of various icons that will be further connected to perform their activities like notification icons will pop up the notification of notified channel. The menu icon will be responsive for the sidebar components. The YouTube icon refreshes the page to home page. The search bar searches the videos corresponding to the keyword entered by the developer. The search button hits the YouTube API and shows up the result of popular videos.

The SideBar component consists of all the lists related to activities of user. The subscribed channels will be shown, history will show the videos that will be played by the user and further all the functions will be made active as soon as the buttons are hit up.

The videos will be displayed from the recommended videos component and other video information will be displayed from the sub components of the recommended component. The video card component helps to display the further information.

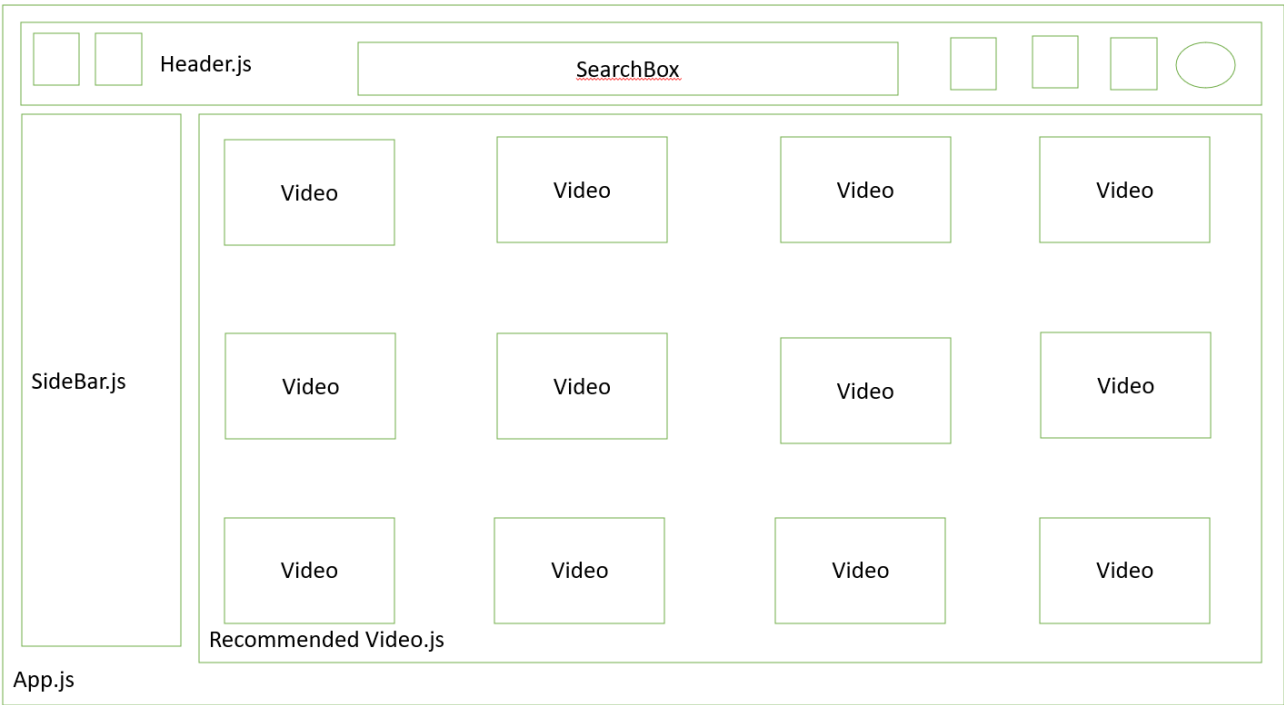


Figure 3.18: Home Component

The Search Page component

This component consists of the header component, sidebar component, and the search page component. Whatever keyword is entered by the developer the channel related to it and the top 10 videos are displayed on the component. After clicking the required video person reaches to

other component where the developer can watch the video and the related information is displayed. The filter icon will be made working after adding some functionality to it.

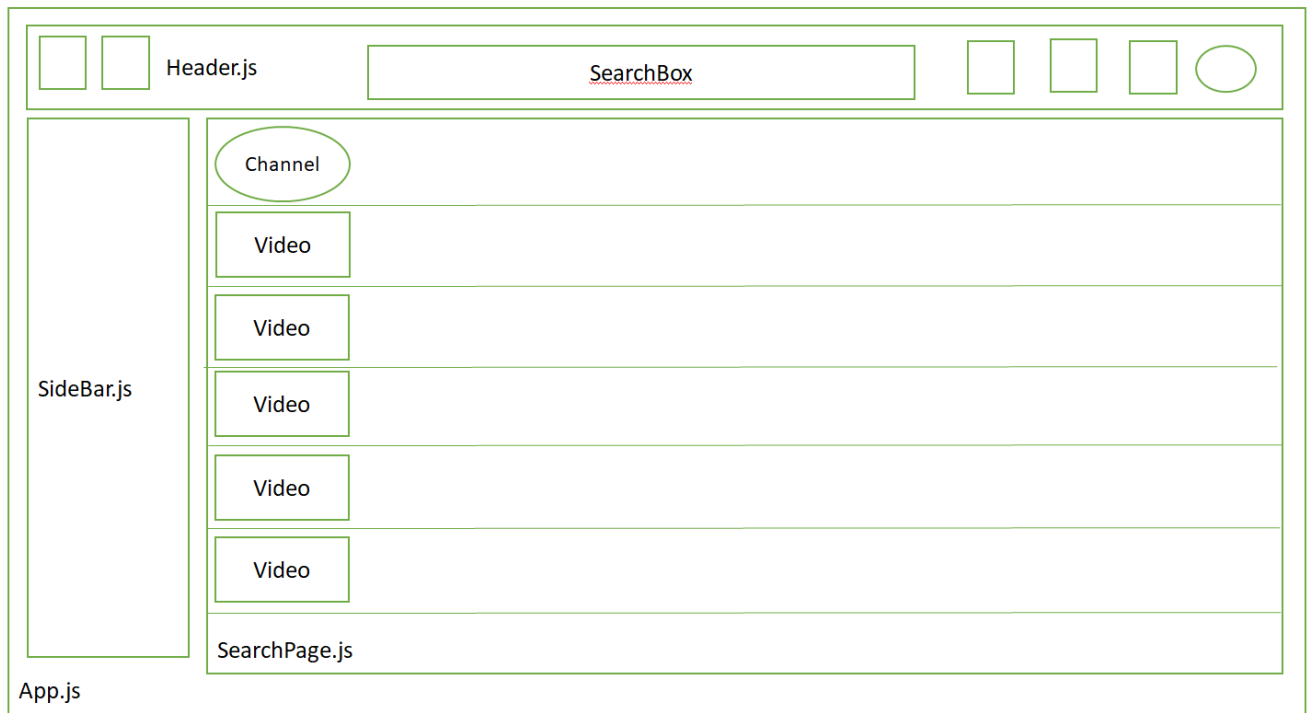


Figure 3.19: Search Page Component

The WatchScreen Component

This component consists of header component the watch screen component and SideBarRow component. The header component works the same way as it was working in the previous components. The watchscreen part shows the video that developer has clicked and the related information of the video. The SideBarRow component shows the recommended videos so that user can see the related videos without much hustle.

In this component the details of video that person wants to see will be shown like the description of video, the channel name which uploaded the video, number of views on the video, the published date of video, the number of likes on the video were displayed.

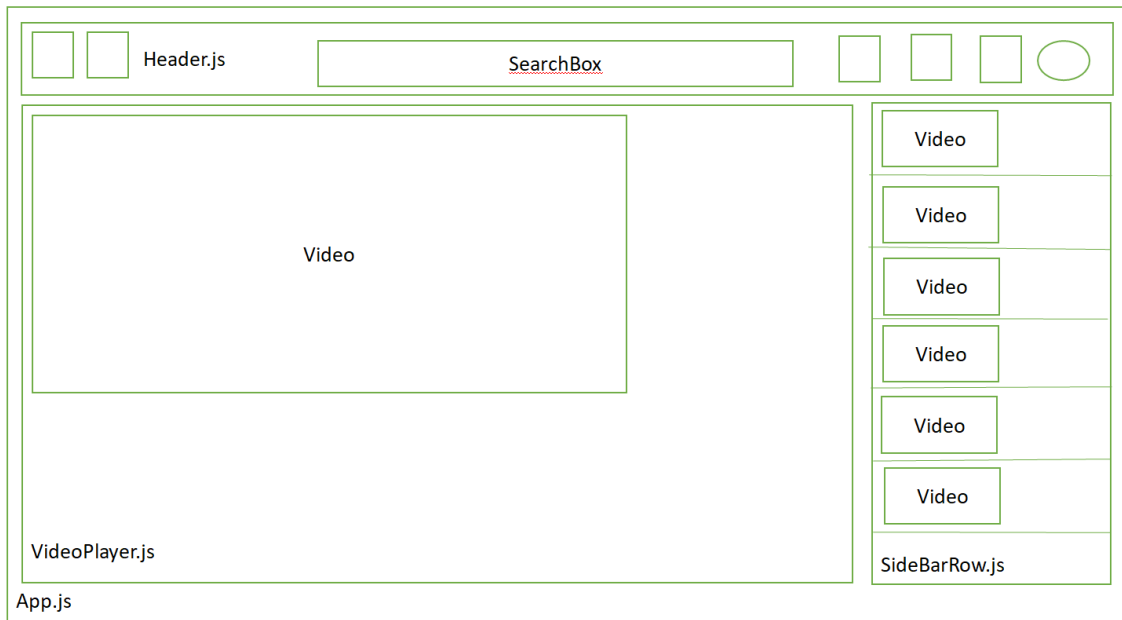


Figure 3.20: WatchScreen Component

Requirement Analysis

The following are the requirements explained that are required to build a react project:

- **Node js and node package manager (npm):**

To execute the javascript code from outside a browser Node js provides the runtime environment. This is used for server side programming. For building input/ output intensive web application Node.js is used. Node .js is asynchronous and event driven. Node.js is asynchronous means that it is non-blocking that is server of Node.js never waits for API to return the data. Node.js outputs the data in the form of chunks and avoids the buffering of data. Node.js uses event looping and single threaded model.

The full form of npm is node package manager. NPM is the world's largest software library. It is package manager for javascript.

- **npx:**

The full form of npx is node package eXecute. If any package is not installed in the npm library then npx allows the developers to execute that javascript package.

npx is a command-line tool that was introduced in version 5.2 of the Node.js package manager, npm. It allows developers to easily execute packages installed in their local or remote npm

registry without having to install them globally or to their project. [16]

The primary use case for npx is to run a command-line interface (CLI) tool that is available as an npm package, without the need to first install it globally or locally in your project. This makes it particularly useful for trying out new packages, testing out a package before adding it to your project dependencies, or using a package that you only need once. [16]

npx works by temporarily installing the requested package and then executing it. Once the command has completed, the package is uninstalled automatically. This means that you can use any package available on the npm registry without having to worry about polluting your global or project environment.

Another use case for npx is to run a different version of a package than the one that is installed in your project. By specifying the package version or location, you can execute a different version of the package without having to install it separately. [16]

npx also provides a range of options that allow you to customize its behavior, such as specifying the package version or location, passing command-line arguments, and choosing how to handle global packages. This level of customization makes npx a powerful tool that can be used in a wide range of development environments. [16]

In addition to executing packages, npx also provides a number of other useful features. For example, you can use it to execute scripts that are included in packages, or to create and run shell scripts on the fly. You can even use it to execute commands from a GitHub repository or from a file URL, making it easy to test and use packages that are not available on the npm registry. [16]

Overall, npx is a powerful and flexible tool that provides a convenient way to execute packages from the npm registry without having to install them globally or locally in your project. It is a valuable tool for developers who want to try out new packages or test out different versions of existing packages, and its range of options and features make it a powerful addition to any Node.js development workflow. [16]

- **Create-react-app:**

This command helps to build the react applications and installs all the basic packages at one go. The developer need not to install the packages one by one. It save the developers time to setup and configure. To use this command the developer needs to install the node and npm. It creates a file named package.json that writes all the dependencies and libraries that were installed in that project.

- **API data from YouTube API:**

To fetch the data from the YouTube API we need to enable the API and with that create a new project.

To do the above task one needs a google id that will be logged onto the google developers console.

Then create a new project and give it a new name.

Select the project and enable the APIs and Services button.

Then click on the credentials where the youtube api key is provided for the particular project.

The required API key is pasted onto the project.

- **To run the project:**

To start the react application uses the command npm start. This command automatically opens the browser and whenever any changes are made in App.js the changes are reflected in the application without reloading the page. The project runs in the localhost:3000.

- **Nodemon**

When the changes are detected in the directory it helps the node.js applications to restart automatically. To use nodemon one needs to install nodemon first and then use it to open the backend file. The wrapping of application is done by nodemon that helps to pass arguments that were passed normally to the application. To run the backend file write the command as nodemon filename.js.

Nodemon is a popular utility for automating the development workflow of Node.js applications. It monitors files in a Node.js project and automatically restarts the application when changes are detected, making the development process more efficient and streamlined.[17]

Nodemon is particularly useful for developers who work on large and complex Node.js projects, where making frequent changes to the code is a common occurrence. Instead of manually restarting the server every time a change is made, Nodemon automates the process and reduces

the need for repetitive tasks. This results in increased productivity and faster development cycles.[17]

One of the key features of Nodemon is its ability to monitor multiple files simultaneously. This means that when any file in the project is modified, Nodemon automatically restarts the application. It can also watch for changes in specific files or directories, allowing developers to focus on the parts of the application that they are currently working on. [17]

Nodemon is highly configurable, with a range of options that can be set to tailor its behavior to the specific needs of a project. For example, developers can set the delay time between restarts, specify the files and directories to monitor, and choose the events that trigger a restart. This level of customization ensures that Nodemon can be used in a wide range of development environments, from small personal projects to large enterprise applications. [17]

Another benefit of Nodemon is its ease of use. It can be installed globally via NPM, and developers can quickly get started by running a simple command in the terminal. Once started, Nodemon will run in the background and automatically monitor the project for changes, without requiring any additional input from the developer. [17]

Nodemon also includes a range of additional features that make it a powerful tool for Node.js development. For example, it can log changes to the console, display errors and warnings, and even run custom scripts before or after a restart. These features help to improve the overall development experience and make it easier for developers to identify and fix issues in their code.

In summary, Nodemon is a valuable utility for any developer working with Node.js. Its ability to automate the development workflow, monitor multiple files simultaneously, and provide a range of configuration options make it an essential tool for streamlining the development process and improving productivity. Whether working on a small personal project or a large enterprise application, Nodemon is a powerful tool that can help developers to work more efficiently and effectively. [17]

CHAPTER – 4

EXPERIMENTS & RESULT ANALYSIS

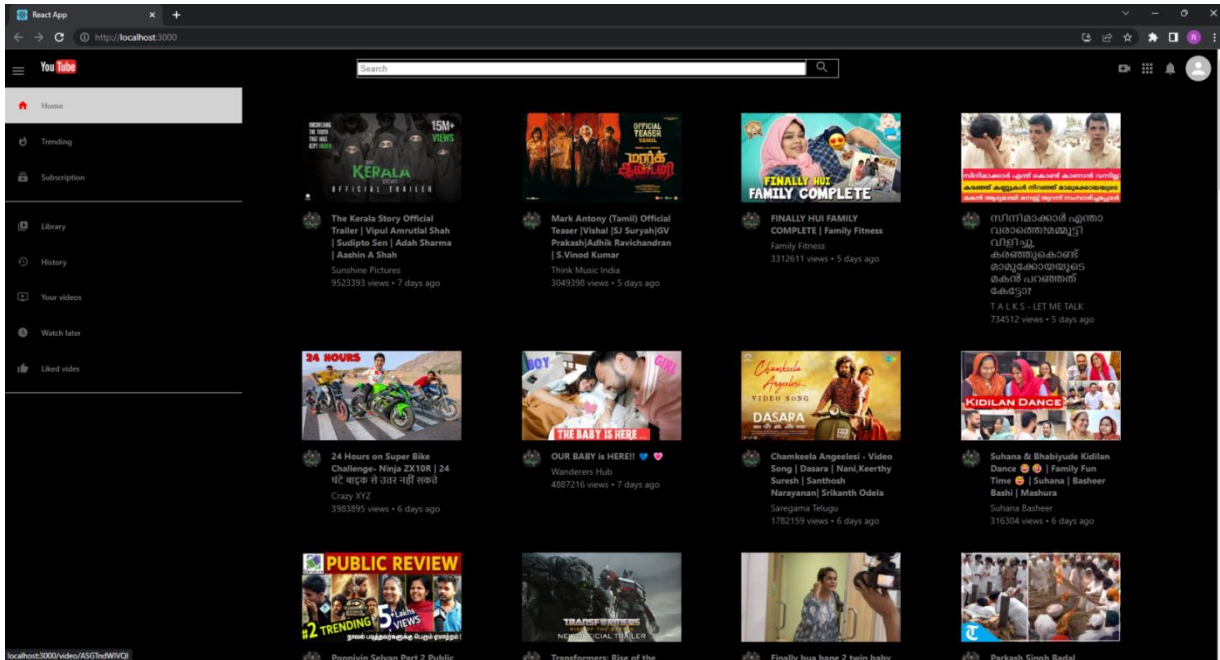


Figure 4.1: App.js

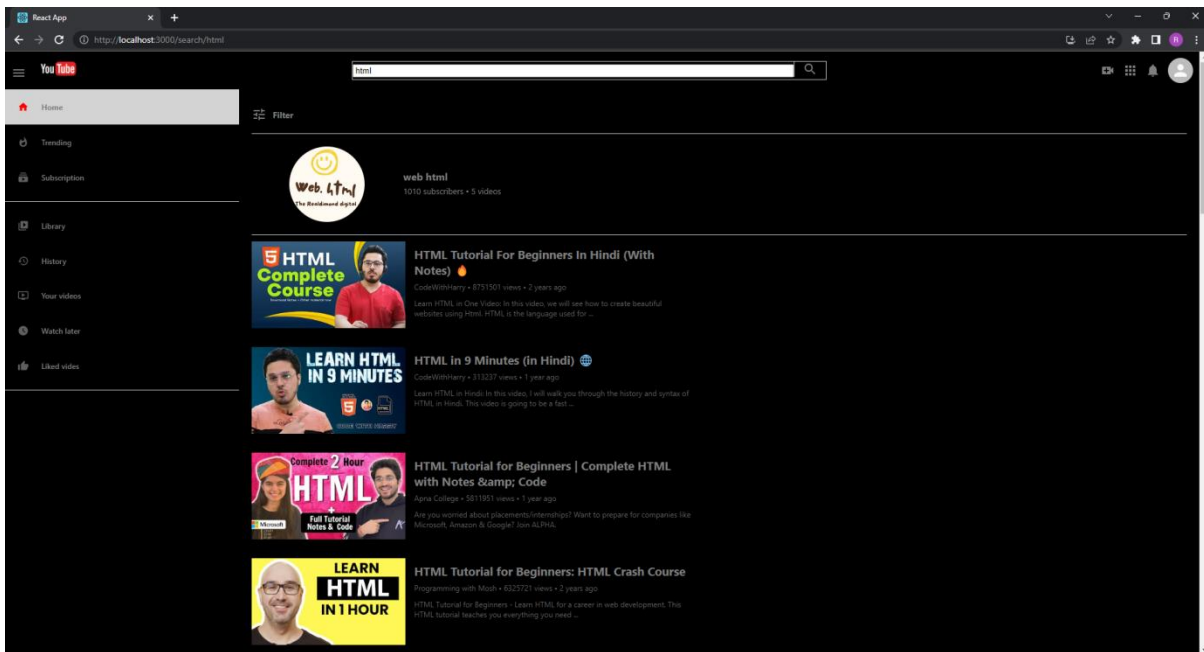


Figure 4.2: SearchPage.js

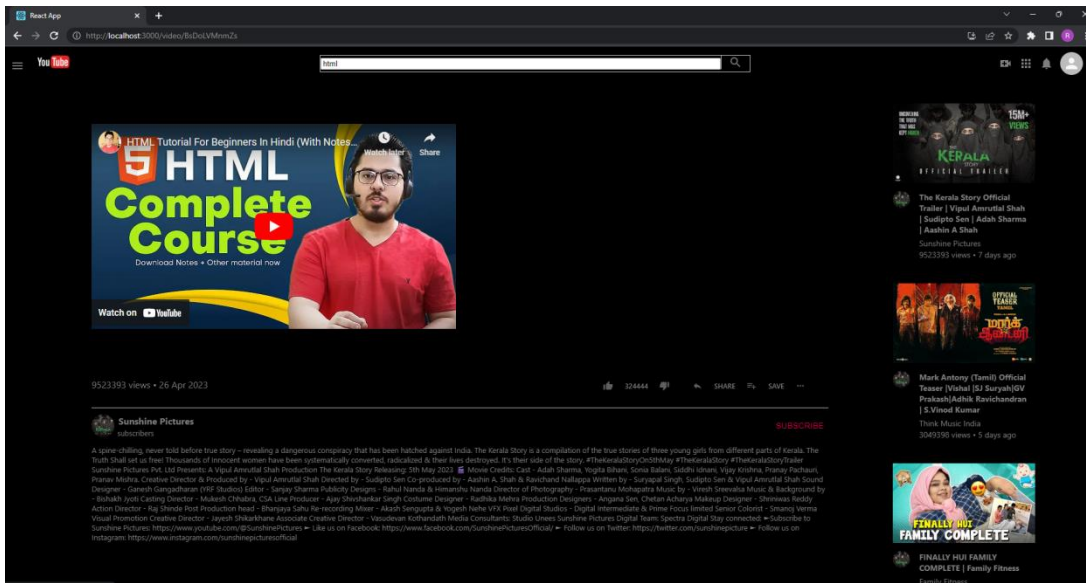


Figure 4.3: VideoPlayer.js

CHAPTER – 5

CONCLUSIONS

5.1 Conclusions

Developing this project was a challenging task for me as this was the first time I built a web application using react, javascript, node and youtube api and managing to play the videos on the application that were fetched from YouTube API data. I utilized all the new concepts in it that were tough and challenging. The main point was to introduce me to backend, frontend and connecting the both together to make an application working. It helped me to understand how node js layer provides security to the backend data as the calls are first made to node js and further the calls are transferred to the YouTube API that renders the video on the react application.

In conclusion, the YouTube Clone Project using React, MongoDB and Node.js is a modern and powerful web application that mimics the features of the popular video sharing platform YouTube.

Using React for the user interface, MongoDB for the backend database, and Node.js to build the server, the project offers a flexible, scalable and efficient solution for building web applications. React provides a declarative approach to creating user interfaces, making it easy to create interactive and reusable components. MongoDB is a NoSQL document-oriented database that is highly scalable and flexible, enabling fast data access and retrieval.

Node.js is a server-side JavaScript runtime that offers an event-driven architecture and a blocking I/O model, making it very efficient and fast. Using these techniques in a YouTube clone project offers several benefits, including better performance, easier debugging, and faster rendering. In addition, the project can be easily customized and scaled to meet the needs of many different web applications.

Overall, the YouTube Clone Project using React, MongoDB and Node.js is a promising solution for building modern web applications that require fast access to data, scalability and flexibility. Thanks to its efficient and flexible architecture, the project can be easily adapted to the needs of different companies and industries, making it an excellent choice for web development projects.

5.2 Future Scope

With React, MongoDB and Node.js, the YouTube clone project has a lot of potential for further development and improvement. Here are some possible future scope areas:

1. **Mobile development:** With the increasing use of mobile devices, developing a mobile app for a YouTube clone project can increase its user base and engagement. React Native can be used to develop a mobile application for both Android and iOS platforms.
2. **Advanced Search function:** Additional search functions can be added to the YouTube clone project to improve the user experience. Machine learning algorithms can be used to recommend personalized content to users based on their search history and viewing habits.
3. **User Generated Content:** User generated content is a great way to increase user engagement and improve the overall user experience. The YouTube clone project can be improved by allowing users to upload their own videos and share them with the community.
4. **Video Editing:** Providing users with video editing tools can improve the quality of videos uploaded to the platform. Features such as cropping, adding subtitles and using filters can be added to the platform to improve the overall user experience.
5. **Monetization:** Monetization of a YouTube clone project can be achieved by offering paid content subscriptions to users or by showing ads on the platform. Payment gateway integration can be added to the platform to facilitate paid subscriptions.
6. **Social Media Integration:** Social media integration can increase the reach of a YouTube clone project by allowing users to share content on popular social media platforms such as Facebook, Twitter and Instagram.
7. **Accessibility:** Making the YouTube clone project accessible to users with disabilities can improve the user experience for a wider audience. Adding features such as subtitles and audio descriptions can improve the usability of the platform.

The YouTube clone project using React, MongoDB and Node.js has a lot of potential for further development and improvement. By adding new features and functions to the platform, the user

experience can be improved and the platform can be made available to a wider audience. These improvements can increase user engagement, revenue and make the platform more valuable to users.

REFERENCES:

- [1] <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
- [2] <https://www.javatpoint.com/javascript-tutorial>
- [3] <https://legacy.reactjs.org/docs/getting-started.html>
- [4] <https://www.javatpoint.com/reactjs-tutorial>
- [5] <https://developer.mozilla.org/en-US/docs/Web/HTML>
- [6] <https://developer.mozilla.org/en-US/docs/Web/CSS>
- [7] <https://mui.com/>
- [8] <https://nodejs.org/en/download>
- [9] <https://expressjs.com/en/5x/api.html>
- [10] <https://www.mongodb.com/cloud>
- [11] <https://firebase.google.com/>
- [12] <https://www.codecademy.com/article/what-is-crud>
- [13] <https://jwt.io/introduction>
- [14] <https://paxcom.ai/>
- [15] <https://www.payments.com/>
- [16] <https://www.npmjs.com/package/npx>
- [17] <https://www.npmjs.com/package/nodemon>