Mini Project Report

on

YouTube Clone

(Report submitted in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Information Technology of BPUT)

Submitted by:

Swagat Senapati (1801227671)
Bikash Ranjan Ojha (1801227668)

Department of Computer Science & Information Technology
C. V. Raman Global University, Bhubaneswar, Odisha, India

26th Novemeber 2020

Contents

Sl	Topics	Page No.
1	Abstract	1
2	Introduction	2
3	Importance / Relevance of the topic	
4	Literature Survey	3
5	Description	
	5.1 Technology	4
	5.2 Methodology / Process	4
6	Results and Discussions	11
7	Conclusion (Advantage/Disadvantage) and Future Work	13
8	References	14

1 Abstract

This mini project is based on cloning the frontend of 'YouTube' website and deploying it in Firebase. YouTube is the most used and largest Video Streaming website in the world. This project website will be a simple replication of complex website with Facebook's framework i.e. **React**. The main pages of this website are Login, Register and Home Page. Along with it, we are using Javascript as it is very simple to learn and implement.

Main objective of this project is to clone the famous streaming website 'YouTube'. Then deploying it with the help of **Firebase** project. Register, Login and Home page will be cloned using React, **Material UI**, HTML, CSS and JS.

2 Introduction

React is a declarative, efficient, and flexible JavaScript library for building user interfaces. React is a **JavaScript** framework built and used by Facebook. It lets you compose complex UIs from small and isolated pieces of code called “components”. Instead of manipulating the browser's DOM directly, React creates a **Virtual DOM** in memory, where it does all the necessary manipulating, before making the changes in the browser DOM. Material UI is also a popular component of React library. It provides many design templates, icons and many components which can be used for a better improvement in our website.

This project is solely targeted towards cloning the entire YouTube Home and Search page. The Project contains JavaScript files for all the parts of the page. Main parts of this YouTube Home page are the **Header, Left Bar** (Side Bar section) and **Recommended Videos** Section. In the Search page of YouTube, main part is the Channel overview and the related videos section. There are separate JavaScript files for separate components i.e. Header part of the page contains separate components for its construction and the left bar of the page contains separate components for its construction. The Recommended videos section contains all the Video Cards and for that Video card, separate JavaScript file is also constructed for its simple use. Then the deployment is done by using Firebase which is the easiest and simplest way to host your website online.

3 Importance/Relevance

React is a simple, yet powerful, UI Library. React's most important advances are the emphasis on components, one-way data flow, the Virtual DOM, JSX, and architecture that extends beyond all other frameworks.

- **Environment** - The popularity of React is now very high due to this features that only this framework provides. The ease of installing it and running it with all the components viewing it in real time makes this the most reliable and simple framework for this website building task. The
- **Virtual DOM** allows the UI to be efficient in defining UI changes. This in-between process between what the component should render and what the actual rendering is that it allows React to work on different platforms.
- **One way data flow** – It uses one-way data flow, so only changes to the data result in changes to the UI. It makes it easier to inspect changes in the app and captures the different states of the app.

4 Literature Survey

Proper design has become a critical element needed to engage website and mobile application users. However, little research has been conducted to define the specific elements used in effective website design. The design elements mentioned most frequently in the reviewed literature were navigation, graphical representation, organization, content utility, purpose, simplicity, and readability.

One of the key design measures is **Usability** which is defined as the extent to which users can achieve desired tasks (e.g., access desired information or place a purchase) with effectiveness (completeness and accuracy of the task), efficiency (time spent on the task), and satisfaction (user experience) within a system. It should also be noted that different disciplines and industries have different objectives in designing websites and should thus prioritize different website design elements.

When it comes to website designing, YouTube has its complex designs which can be very useful to learn and create. The complex designing of menu bar, head bar and the videos section makes it a great example of learning frontend web designing for beginners.

5 Description

5.1 Technology:

Visual Studio Code, Firebase, React JS, Material UI, Node JS, HTML, CSS and JS.

- **Visual Studio Code** – It is a free open source Code Editor in which we are going to build this React app. It is a light weight code editor with all its intelligent functions to help you while writing codes. Extensions in its marketplace make it the most used and most simplified code editor in laptops.

5.2 Methodology:

Firstly, a new app is created using the command 'npx create-react-app <Folder Name>" by which a new react app is created within that folder which contains all the necessary elements required to run a React app. To create this you should have your **React** and **Node.js** installed. **NPM(Node Package Manager)** should be installed to use npm in the command. To use all the main Material UI components to be used in the react app, we should first install the node modules of Material UI. To install the Material UI components in our app, we should install Material UI node modules by command 'npm install @material-ui/core' and to install all Material UI Icons we use the command 'npm install @material-ui/icons'.

It will create essential files required to run your app. This contains a src folder which contains mainly **App.js** and Index.js importantly. App.js file contains the main page of your website. **Index.js** stores main Render call from ReactDOM. It imports our App.js component which tells React where to render it. **react-dom** is for rendering the components in the DOM. '**react-dom**' acts as a glue between components and DOM. We will be using render() method of the react-dom to render components in the DOM.

The components of the Home page of YouTube is divided into 3 sections: Header, Left Bar (Menu Bar) and Recommended Videos Section.

Home Page:

Header:

- Firstly, the main component to be created is Header part of the YouTube, This contains of 3 parts- left, middle and right. The left part contains the **Menu Icon** and the **YouTube icon**.
- The middle part contains a long **search text bar** with a border and a **Search icon** button.
- The right part contains all the icons i.e. **Avatar, notifications icon, videos icon and all apps** icon.
- The Head.js file contains all the HTML part for the header and all the parts are divided with ** ** tags for 3 separate parts.
- The Head.css part contains all the styling required for all the HTML part in Head.js.
- **Avatar** icon is imported from Material UI Core part and all the other icons are imported from Material UI Icons pack.
- The Header part is made **sticky** on the top with a white background so that if you scroll the page down, the header will remain as it was before and it will not be **scrollable**.

Fig: Header of YouTube

Left Bar (Menu Bar):

- This is the Menu bar of YouTube on the left side in which it contains all the different sections of YouTube i.e. Home, Library, Trending, etc.
- Firstly, a LeftBarRow.js file is created in which an icon and its title is arranged. This is for one particular row in which one icon and its title should be arranged which can be used for multiple rows in the left bar.
- LeftBarRow.js takes props/components as Icon and Title for the values to be entered.
- Next, In the LeftBar.js, the props are entered as variables and this variables are assigned the values as Icons are given the imported Material UI Icons and titles are given according to it.

- **Home Icon** for Icon, **Whatshot Icon** for Trending, **Subscriptions Icon** for Subscriptions, **VideoLibrary Icon** for Library, **History Icon** for History, **OndemandVideo Icon** for Your Videos, **WatchLater Icon** for Watch Later, **ThumbUpAltOutlined** Icon for Liked Videos and **ExpandMoreOutlined Icon** for Show More section.
- For more simplicity, mouse hovering option is added for which the icon color and style of title gets changed when the mouse pointer is hovered over it. This styling is done in LeftBarRow.css for all the rows.
- A horizontal line <hr/> is also added to separate the sections as you can see in the real YouTube website.

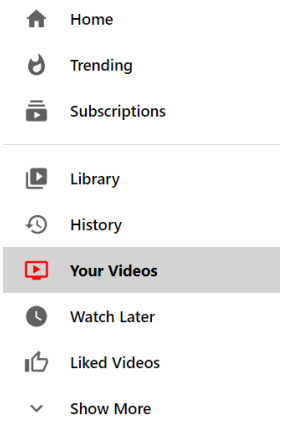

Fig: Menu Bar of YouTube

Recommended Videos:

- This is the main section of the page where all the videos are displayed to an user. The videos are shown in a Card format arranged in rows and columns.
- Firstly, a **VideoCard.js** file is created for creating a one unique specific video card with video Thumbnail, channel name, channel image, video title, no of views and the time since it got uploaded.
- **Thumbnail Image, Channel Name, Image, Views, Timestamp** and the video **Title** are taken as props here in the VideoCard function.
- Whole styling for a particular video card is done in VideoCard.css for its total overview of a card.
- Then this function is used as a tag in **Videos.js** file where <VideoCard> tag is used multiple times for multiple videos. Each tag holds the above props in section to decorate that card in that particular manner.

- Video thumbnail has a height of **140px** and width **250px** for a perfect thumbnail image.

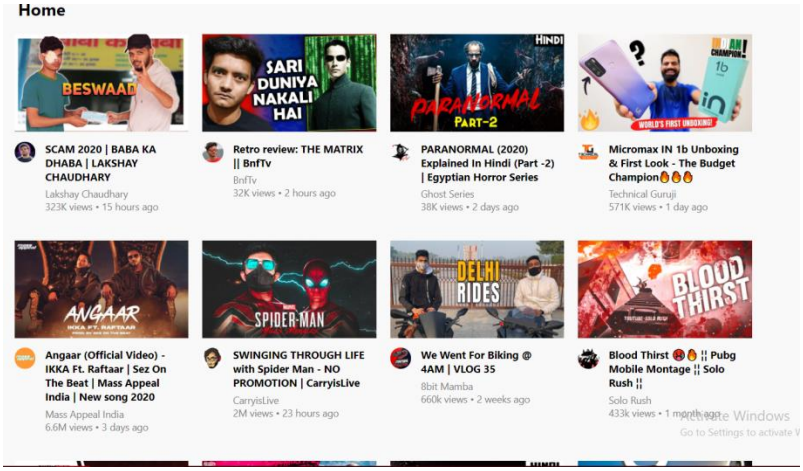

Fig: Recommended Videos Section

Search Page:

- This page contains the search list of the videos tags you searched for in the search bar.
- Divided into 2 parts. One is the List of **Channels** and other is the List of **Videos**.
- At the top, Channels will be displayed and bottom there will be the list of Videos.
- Both Left Bar and Header is also present in this page.

Channels Section:

- This section contains the channels list that you searched for. As it is only for frontend part, therefore entering anything in search bar, will show you the same results page.
- Channel section is made of Channel.js file which contains one **Avatar**(for your Channel Image), **Channel Name, Subscribers, No. of videos, Channel Description** and a **Verified tick**.
- The above components are taken as props in Channel.js which is to be used in Search.js.
- Avatar is taken as **120px X 120px** for better adjustment with the channel logo.

- All the styling required for the channel props to be in proper place, styling is given in the Channel.css file.
- Verified Tick Material UI Icon is also given for those channels which are verified in which verified is taken as a Boolean function if to check that channel is verified or not.

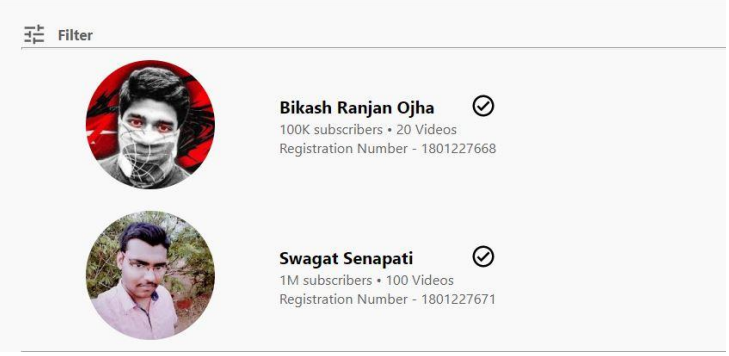

Fig: Channel Section in Search page

Videos Section:

- This section shows the videos that you searched for and these videos are shown only in row wise.
- For creating one particular row of a video file, VideosRow.js is created using props as **Channel Name, Video Thumbnail, No. of views, Subscribers, Description** of video**, Video Title** and the **Timestamp**.
- Taking these props in VideosRow function, this can be used multiple times in Search.js to use multiple videos list as **<VideosRow>** tag.
- Styling of one particular video part is done in VideosRow.css.
- The image Thumbnail is taken of size height **138px** and width **246px**.
- The subscribers count here in the video is taken in tag for better visibility with a grey background and blue text.
- In Search.js, all the functions are used. VideosRow() for list of videos and Channel() for list of Channels.
- A horizontal line <hr/> divides the channel list and videos list.

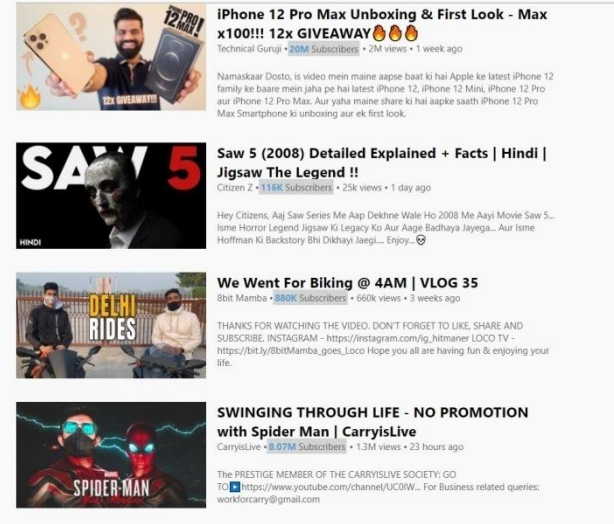

Fig: Search Videos Section

Deployment:

- After building all the pages, the building and deployment are the most important parts of a website.
- This deployment is to be done on Firebase. Firstly, we should create a project in Firebase and then we have to click on its configuration to get it's configuration json file.
- We have to create a firebase.js file and have to copy its configuration in that file to connect that react app to firebase project.
- If you are not logged in to your account with your firebase in that project, then we have to run '**firebase login**' and login to firebase with credentials.
- Next we have to type '**firebase init**' in terminal, to select for what function we were going to configure our firebase project.
- Selecting Hosting will get our firebase ready to host our website.
- Next, we have to select our project folder name and then build it. It will create a build folder with all the requirements and then it will be ready for deployment.
- Next we have to build the react app by running '**npm run build**'.
- By entering this command, a build folder created in which there is one index.html file and all other files required for our website in which it contains all the JS files we created combined into one HTML page. The

flow in which we created in our JS files are now combined into this HTML file with the same functions and same images and same designs.

- For deployment, we write '**firebase deploy**' in terminal to deploy it successfully.
- After successful deployment, it will give us a link from which we can open that same website.
- For our project, the link is https://clone-b1d83.web.app .

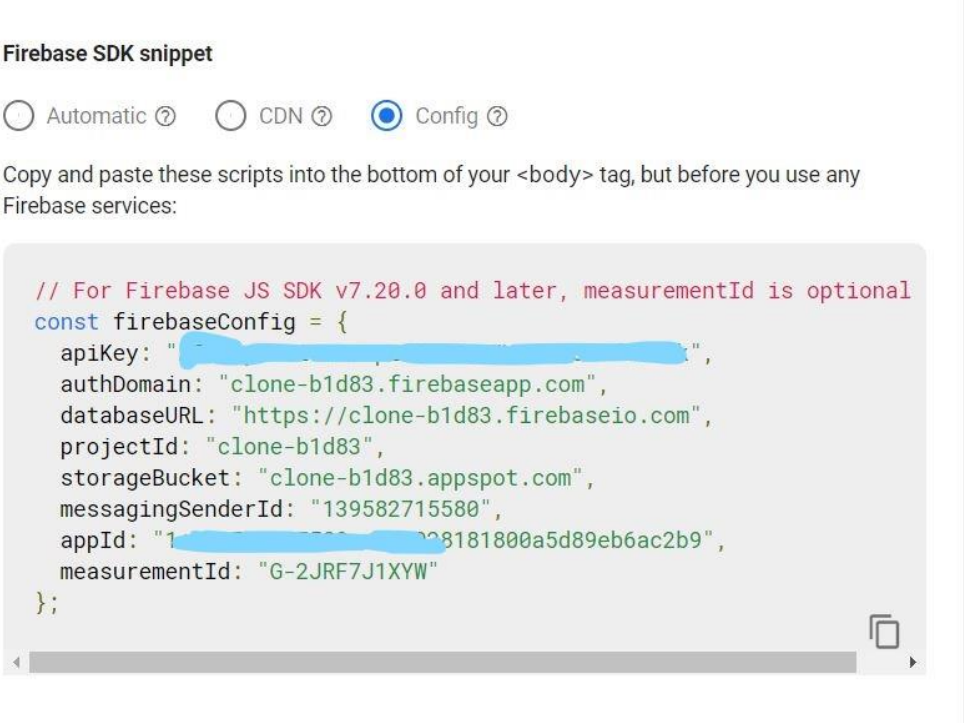

Fig: Firebase Project Config file

6 Results

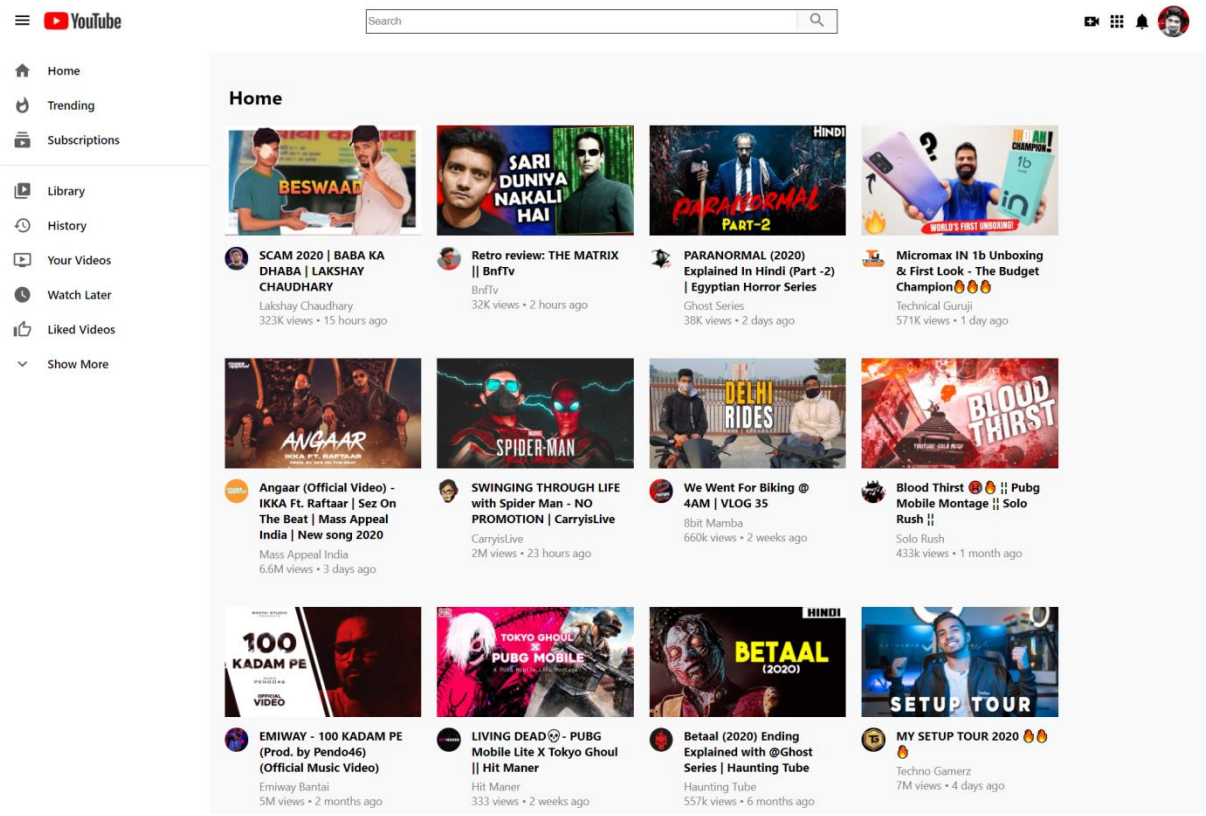

Fig: Home Page of YouTube

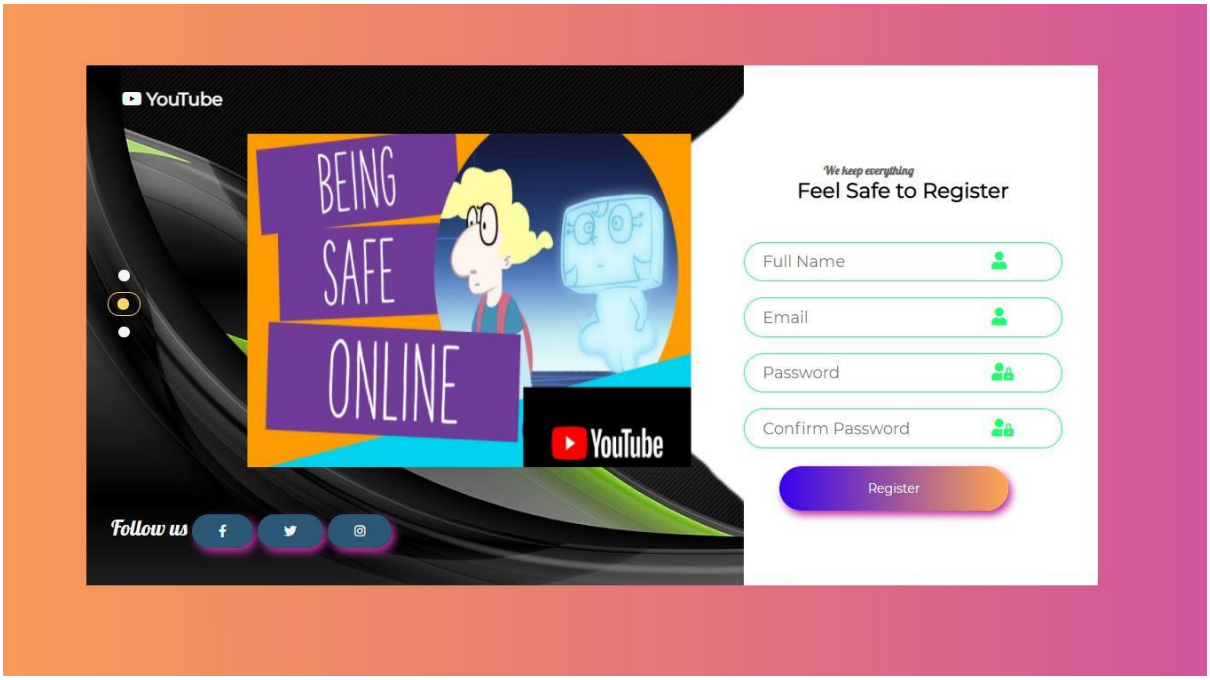

Fig: Register Page

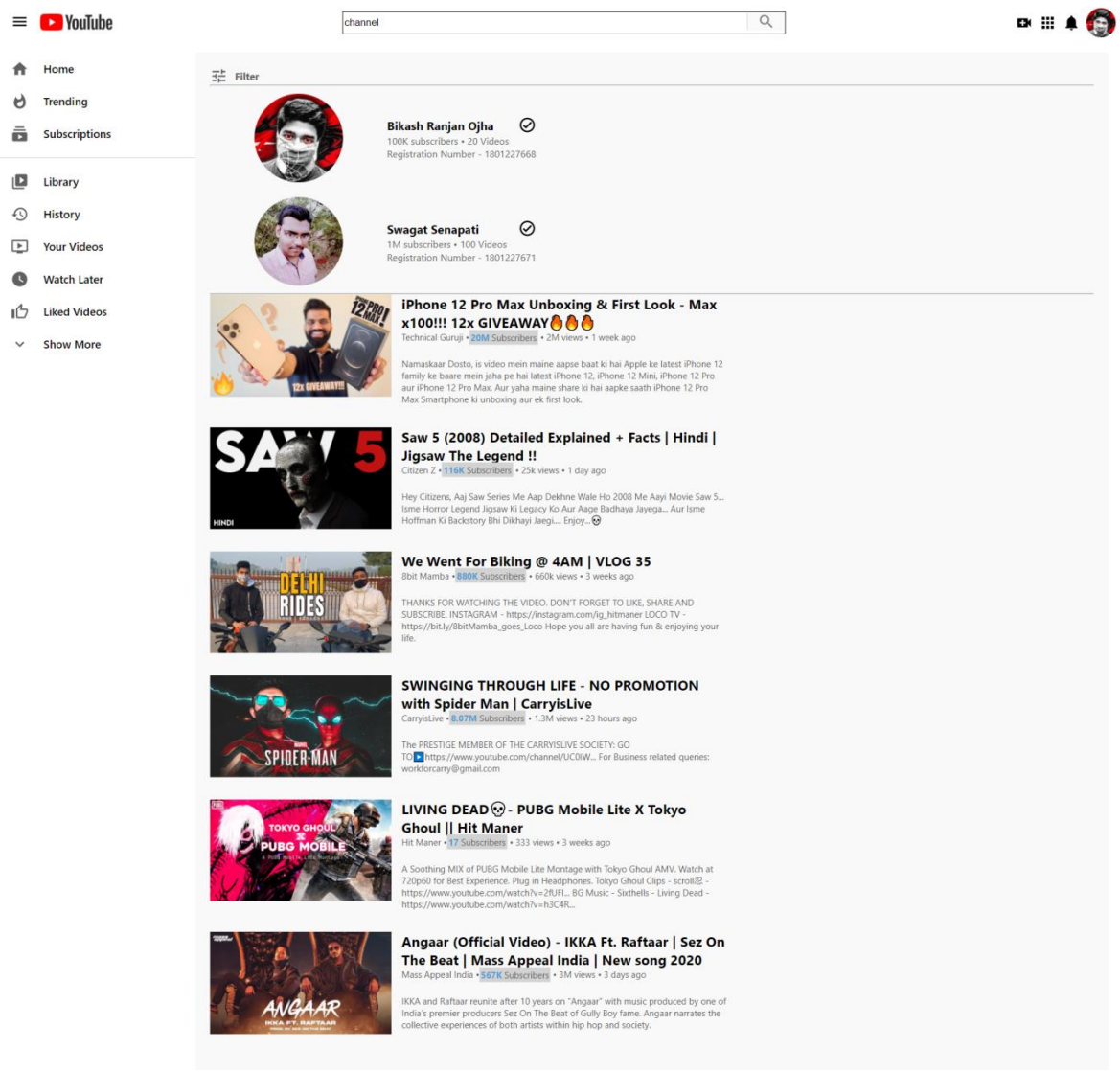

Fig: Search Page of YouTube

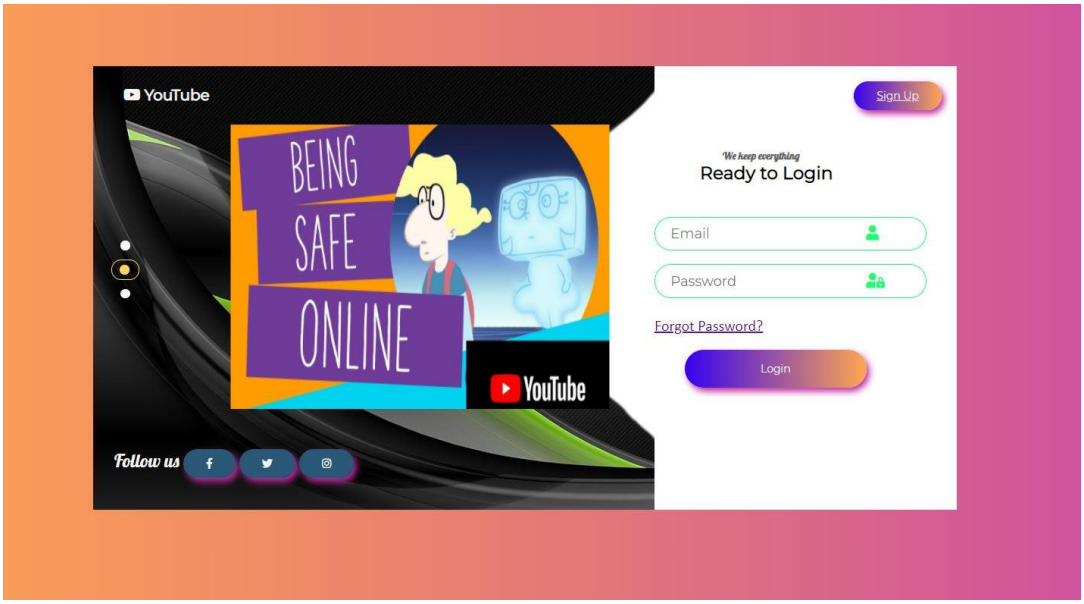

Fig: Login Page

7 Conclusion

This is a simple replication of a social media website. As YouTube is a most widely used social media in the world, we decided to replicate a similar website. This is a platform where every individual can showcase their internal talent, gain knowledge and also can earn money. In this project we basically focused on the frontend part. Later we can work on integrating advanced features to it along with that we can work on the backend part. We used simple technology to design this. We created four main pages – Home and Search pages of YouTube, Register and Signup pages. The deployment of this website is made very easy by using Firebase. Using the firebase to deploy, is a simplified way to build all he files into one index.html page and then provide a link with that page. It is completely free and safe to use.

Project Links:

1) **GitHub -** https://github.com/BikashBIOS/YouTube-Clone

2) **Firebase Deployed Link –** https://clone-b1d83.web.app

8 References

- YouTube Tutorials – React
- React Official website- https://reactjs.org/docs/getting-started.html
- Firebase - https://firebase.google.com/
- Material UI - https://material-ui.com/
- W3Schools React - https://www.w3schools.com/react/