*A Mini Project Synopsis on*

# Memories Website

## T.E. - I.T Engineering

## Submitted By:

| | |
|---|---|
| Hitesh Sachdev | 20104137 |
| Siddhesh Sawan | 20104044 |
| Aman Yadav | 20104050 |

**Under The Guidance Of:**

Prof. Manjusha Kashilkar

**DEPARTMENT OF INFORMATION TECHNOLOGY**

A.P. SHAH INSTITUTE OF TECHNOLOGY
G.B. Road, Kasarvadavali, Thane (W), Mumbai-400615
UNIVERSITY OF MUMBAI

**Academic year: 2022-23**

# CERTIFICATE

This to certify that the Mini Project report on “**Memories Website”** has been submitted by Hitesh Sachdev (20104137) , Siddhesh Sawan (20104044) and Aman Yadav (20104050) who are a Bonafide students of A. P. Shah Institute of Technology,Thane, Mumbai, as a partial fulfillment of the requirement for the degree in **Information Technology**, during the academic year **2022-2023** in the satisfactory manner as per the Curriculum laid down by University of Mumbai

Prof. Manjusha Kashilkar
Guide

| | |
|---|---|
| Dr. Kiran Deshpande | Dr. Uttam D. Kolekar |
| Head Department of Information Technology | Principal |

External Examiner(s)

1.

2.

Place: A.P. Shah Institute of Technology, Thane

Date:

# ACKNOWLEDGEMENT

This project would not have come to fruition without the invaluable help of our guide **Prof. Manjusha Kashilkar.** We express our gratitude towards our HOD, **DR. Kiran Deshpande**, andthe Department of Information Technology for providing us with the opportunity as well as the support required to pursue this project. We would also like to thank our peers for their helpful suggestions.

# TABLE OF CONTENTS

# Chapter 1

## ❖ Introduction

Life passes by quickly, so saving special moments on social media is a fun way to remember events and share with others. So we're working on Memories Application which allows users to post interesting events in their life and revisit those memories.
And also memories application provide information about the person as a whole, on the one hand, and how good one is at various skills on the other

### HOW MEMORIES DEFINE US?

Memories make us who we are. They create our worldview in ways we hardly realize. Like a character made of Legos, we're built of blocks of memory that all fit together to form our consciousness.

Our website provides a users to post  interesting events that are happening in their life and revisit those memories.we have a searching option through which user can search for memory and see other users uploaded memory

And Someone has said that ‘Each happiness of yesterday is a memory for tomorrow.’

## ❖ Purpose:

The purpose is to build user friendly website where people can share important events which are happening in their people life.

## ➢ Problem Statement:

- **Problem Identified :**
  One click searching of post based on different topics is not available on other websites.

- **Solution Proposed :**
  Our website provide one click searching of post based on different topics.
  Click on the topic and see post caption with that topic name.

## ➢ Objectives:

- The objectives of our project are as follows:
- To build user friendly website.
- To add memories like places they are visited or things they have enjoyed.
- To add videos of your memories with title.
- Users can share the experience through this.
- Users can search of post based on different topics.
- Users can like the post and also comment on that post.

## ➢ Scope:

It can be useful to those who wish to showcase and share their important memories in life a place for Social media enthusiast.

Can be useful for Creating and store memory with title and other information.

# Chapter 2

## ❖ Literature Review

The Swedish social networking website LunarStorm, originally called Stajlplejs, is launched in 1996. The site, founded by Rickard Eriksson, was renamed LunarStorm in 2000 and has been described as "The world's first social media on the Internet".

The social networking sites are nowadays playing an increasing role in people everyday life and the use of the social networking sites has become more and more important as a way of communicating in the last couple of years. Social networking sites like Twitter, Facebook, YouTube and MySpace etc. have gained more and more popularity. From the social networking sites especially the “big tree” services as Facebook, Twitter and YouTube have grown tremendously in the last couple of years. Politicians are now using social media and the Internet and in turn are permitting a new arena of grassroots politics. Facebook has most number of active users which is in millions nearly 2,910 million and after that youtube around 2,562 million user.

## There are some apps similar to our application

- **FACEBOOK:**

  Facebook, American online social network service that is part of the company Meta Platforms. Facebook was founded in 2004 by **Mark Zuckerberg**. Access to Facebook is free of charge, and the company earns most of its money from advertisements on the website. New users can create profiles, upload photos, join a preexisting group, and start new groups.

- **YOUTUBE**:

  YouTube, Web site for sharing videos. It was registered on February 14, 2005, by
  **Steve Chen**.it was attracting some 30,000 visitors per day. By the time YouTube was officially launched on December 15, 2005, it was serving more than two million video views each day.

- **TikTok**:
  The app was launched in 2016 by the Chinese technology company **ByteDance**.TikTok is a popular social media app that allows users to create, watch, and share 15-second videos shot on mobile devices or webcams. With its personalized feeds of quirky short videos set to music and sound effects, the app is notable for its addictive quality and high levels of engagement.

# Chapter 3

## ❖ Proposed System

In this project we are using TypeScript, ReactJs, and NextJs for frontend and Sanity for backend and here we use Google oauth for Google login in NextJs. We learned file-base routing and data fetching that allow server side rendering and static generation.

Our website provides users to post interesting events that are happening in their life and revisit those memories.we have a searching option through which user can search for memory and see other users uploaded memory.

### ➢ 3.1 Features and Functionality:

- Pagination Feature: Which allows you to only fetch a certain no. of memories at a time.
- Search and Filtering: We can search tag like places or can search title and can get all the matching Memories.
- Comments Feature: You can Comment on Users post.
- Memories Detailed Page: U can click on any memories to see more Detailed information about it.
- Recommended Memories Feature: It shows recommended memories to the user.
- One click searching of post based on different topics is available to users.

# Chapter 4

## ❖ Feasibility study

Whenever a new system (hardware or software) is to be introduced, there is a need to study every aspect or manner before working on it. The four main consideration of the study are:

- Time Feasibility: Time feasibility refers to the time management of the project. It refers to the time and process incurred during the development of the project. The project has been made with keeping time constraint and quality in mind.

- Technical Feasibility: Technical feasibility refers to technical knowledge and auxiliary devices required. Since our project is in Visual Basic 6 so we need to have a strong base of this programming language. And programming language we have used is React JS.

- Costing Feasibility: Costing Feasibility refers to the cost the project members have done toward the project since our project is tried to be made as economical as possible.

- Economical Feasibility: The hardware/software setup required is that the proposed system can be easily run on any dual core smartphone and as the software used to build system is Visual Studio code in windows 10 or we can build this in Linux also. So it does not cost high.

- Operational Feasibility: The project is user friendly as it can be used by anyone with the basic knowledge of computer.

# Chapter 5

## ❖ Hardware and Software Requirements

### ➢ Hardware Requirements:

- Standard computer with at least i3 processor Standard computer with 4GB of RAM.
- Standard computer with 100GB of free space.
- Active Internet Connectivity with good bandwidth.

### ➢ Software Requirements:

- ReactJs
- HTML , CSS , JavaScript
- Visual Studio
- NextJs
- Sanity
- API

### ➢ Operating System:

- Windows 10

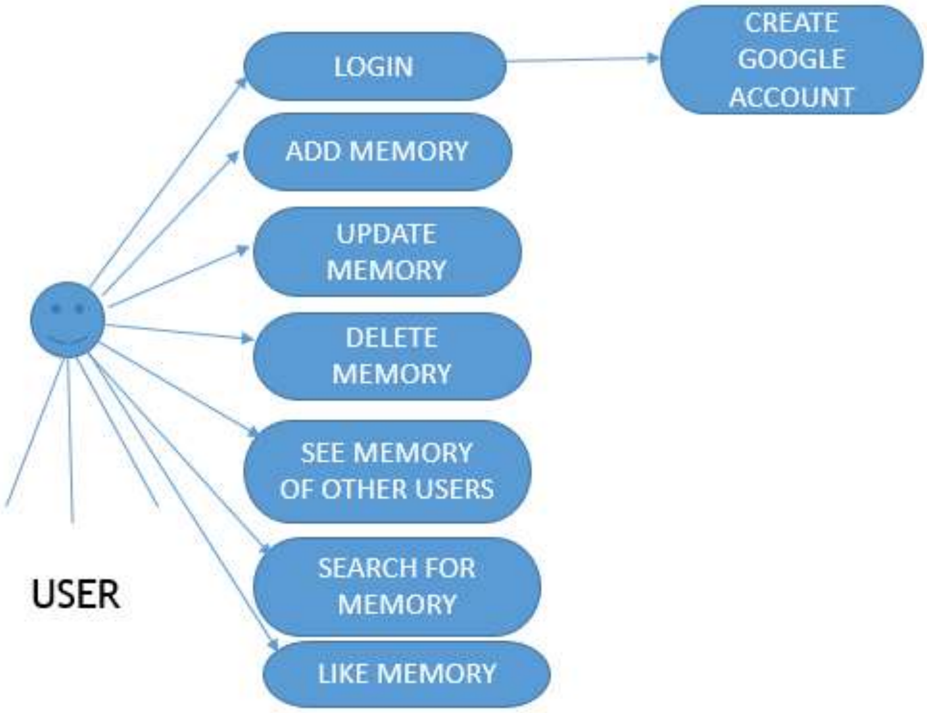
# Chapter 6

## ❖ Project Design

In this phase, a logical system is built which fulfils the given requirements. Design phase of software development deals with transforming the client's requirements into a logically working system.

### ➢ User Case Diagram:

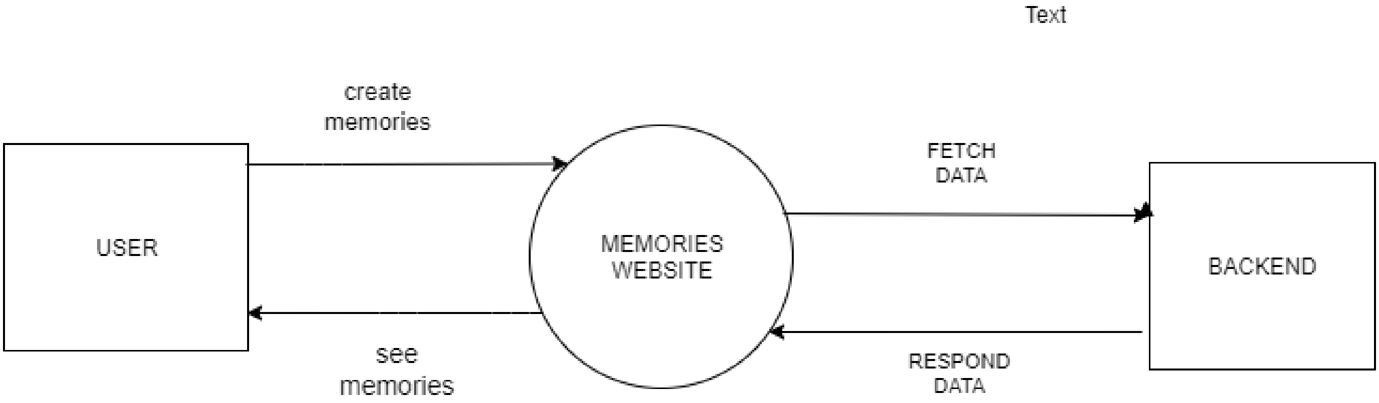
In the Unified Modeling Language (UML), a use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system. To build one, you'll use a set of specialized symbols and connectors.



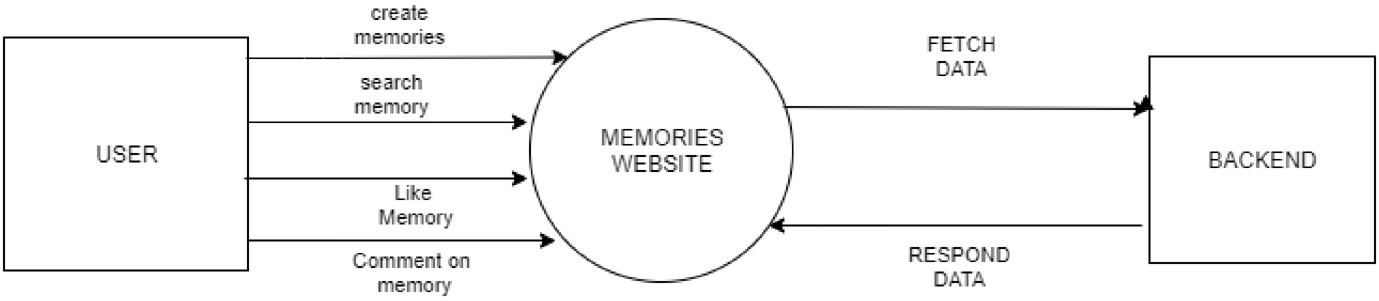
## ➢ Data Flow Diagram:

A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that dig progressively deeper into how the data is handled. They can be used to analyze an

Existing system or model a new one.



LEVEL 0



LEVEL 1

# Chapter 7

## Setup Phase

It starts from downloading the Node.js to running some commands to setup the project.
The easiest way to get started with Next.js is by using create-next-app. This CLI tool enables you to quickly start building a new Next.js application.

```
npx create-next-app@latest –ts
```

This will create a new Next.js app and also install dependencies:

```
--react
--react-dom
--next
```

The command line tool is used to set up new projects, manage datasets, import data, and much more. We'll be using it to get our project up and running.

```
npm install -g @sanity/cli
```

This command will help us setup CLI.

Sanity Studio has most of its logic and code contained in NPM packages. This makes it really easy to upgrade to new versions (sanity upgrade) and to install and ship plugins as self-contained packages.

To get started with a new project, run this command after we have installed the CLI tool:

```
sanity init
```

This will log you into Sanity, create a project, set up a dataset, and generate the files needed to run the development server locally. There are a number of options available, which we can see by running **sanity help init**.

If we get a Success! message, the studio is configured and we're ready to go.

## Run the studio

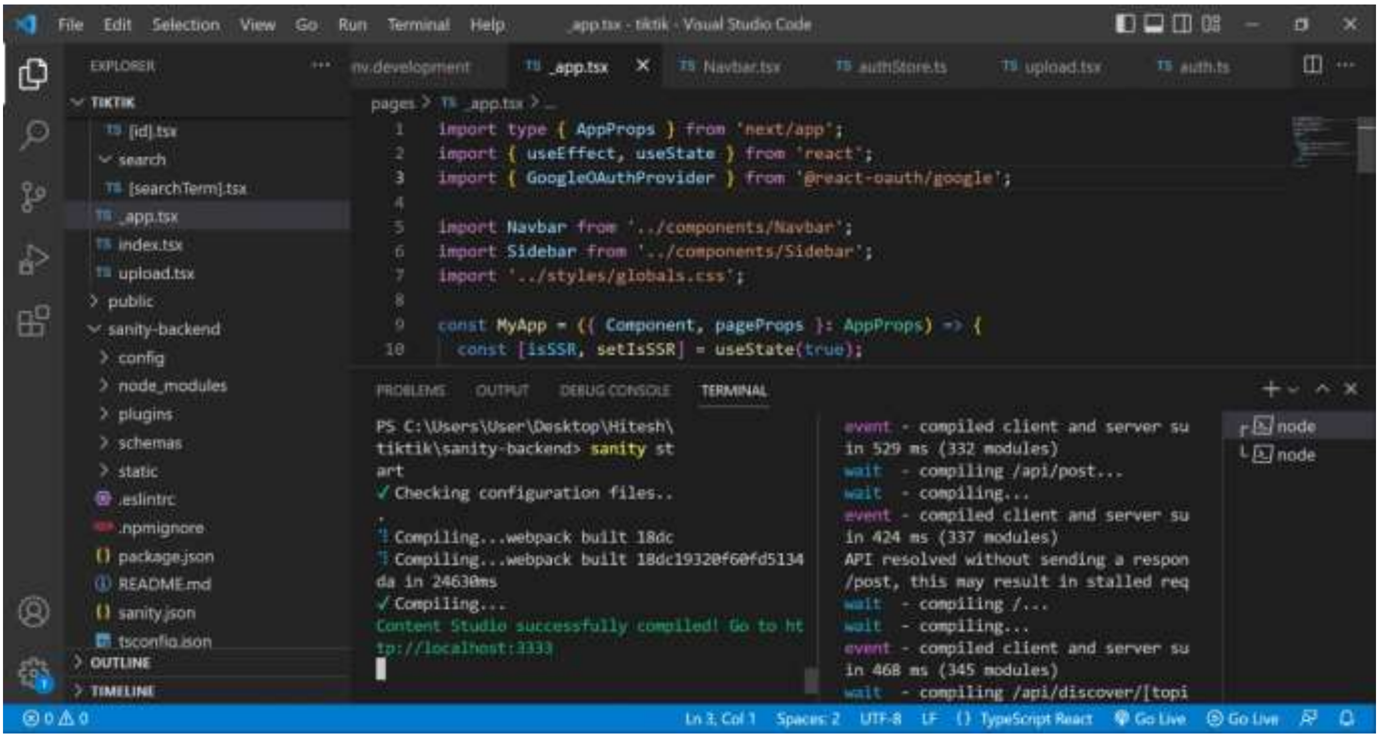
To start the development server for Sanity Stud**io, run this command within the** studio folder:

**sanity start**

This builds the initial JavaScript code required to run the studio and starts a local web server. As you modify and save the code, the server will automatically rebuild the studio and refresh the browser.

The studio is just an HTML file and some JavaScript bundles that run in your browser. It talks to the Sanity API which stores your data and lets you query it from whatever platform or front-end you want.

We can end our development server by pressing Ctrl-C in the terminal where the server is running.



```
import type { AppProps } from 'next/app';
import { useEffect, useState } from 'react';
import { GoogleOAuthProvider } from '@react-oauth/google';

import Navbar from '../components/Navbar';
import Sidebar from '../components/Sidebar';
import '../styles/globals.css';

const MyApp = ({ Component, pageProps }: AppProps) => {
  const [isSSR, setIsSSR] = useState(true);
```

```
PS C:\Users\User\Desktop\Hitesh\
tiktik\sanity-backend> sanity st
art
✓ Checking configuration files..
.
⠸ Compiling...webpack built 18dc
⠸ Compiling...webpack built 18dc19320f60fd5134
da in 24630ms
✓ Compiling...
Content Studio successfully compiled! Go to ht
tp://localhost:3333
```

```
event - compiled client and server su
in 529 ms (332 modules)
wait  - compiling /api/post...
wait  - compiling...
event - compiled client and server su
in 424 ms (337 modules)
API resolved without sending a respon
/post, this may result in stalled req
wait  - compiling /...
wait  - compiling...
event - compiled client and server su
in 468 ms (345 modules)
wait  - compiling /api/discover/[topi
```

# Setting up Google Authentication

Google has discontinued the Google Sign-in JavaScript Platform Library for web and has blocked all of the npm packages using their Google Sign-in for websites. Instead it has newly launched Google-Identity Services for web. It’s supporting and streamlining multiple types of credentials.

@react-oauth/google it is the package we need.

Google OAuth2 using the new Google Identity Services SDK for React @react-oauth/google

Installing the package using command

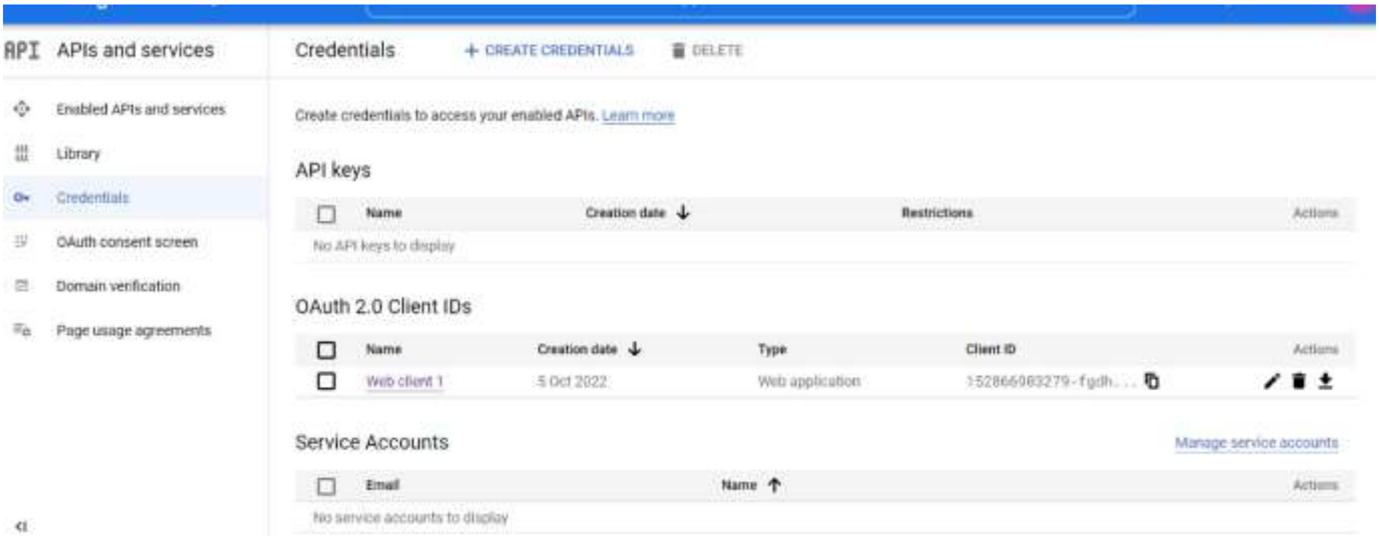
```
$ npm install @react-oauth/google@latest
```

## One-tap sign-up

Sign up new users with just one tap, without interrupting them with a sign-up screen. Users get a secure, token-based, passwordless account on your site, protected by their Google Account.

## Setting up Credentials

1. Get your Google API client ID
2. Configure your OAuth Consent Screen
3. Wrap the application with GoogleOAuthProvider



Create credentials to access your enabled APIs. Learn more

API keys

| □ | Name | Creation date ↓ | Restrictions | Actions |
|---|---|---|---|---|
| No API keys to display | | | | |

OAuth 2.0 Client IDs

| □ | Name | Creation date ↓ | Type | Client ID | Actions |
|---|---|---|---|---|---|
| □ | Web client 1 | 5 Oct 2022 | Web application | 152866983279-fgdh... | |

Service Accounts

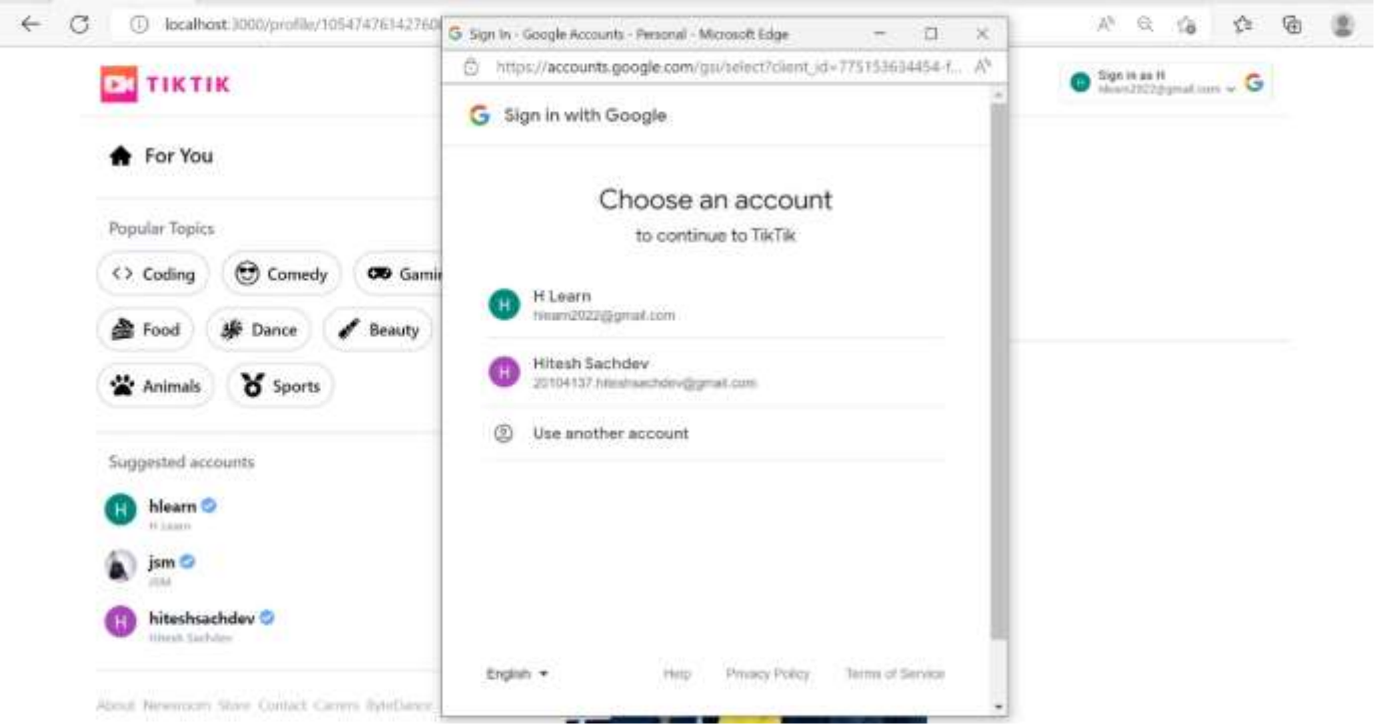
| □ | Email | Name ↑ | Actions |
|---|---|---|---|
| No service accounts to display | | | |

# Chapter 8

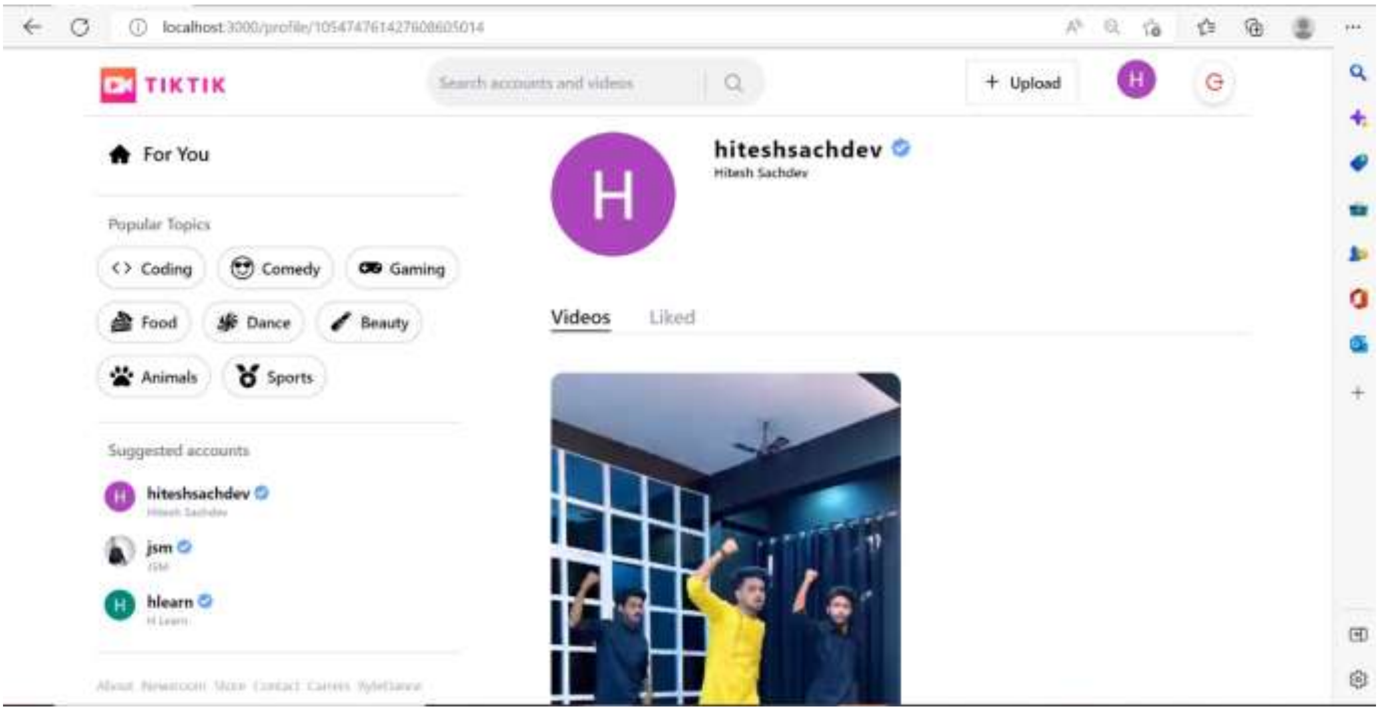
# Implementation

## _app.tsx

```
import { GoogleOAuthProvider } from '@react-oauth/google';
return (
  <GoogleOAuthProvider clientId={`${process.env.NEXT_PUBLIC_GOOGLE_API_TOKEN}`}>
   <div className='xl:w-[1200px] m-auto overflow-hidden h-[100vh]'>
    <Navbar />
    <div className='flex gap-6 md:gap-20 '>
     <div className='h-[92vh] overflow-hidden xl:hover:overflow-auto'>
      <Sidebar />
     </div>
     <div className='mt-4 flex flex-col gap-10 overflow-auto h-[88vh] videos flex-1'>
      <Component {...pageProps} />
     </div>
    </div>
   </div>
  </GoogleOAuthProvider>
 );
```



localhost:3000/profile/105474761427608
TIKTIK
For You
Popular Topics
Coding
Comedy
Gaming
Food
Dance
Beauty
Animals
Sports
Suggested accounts
hlearn
H Learn
jsm
JSM
hiteshsachdev
Hitesh Sachdev
About Newsroom Store Contact Careers ByteDance
Sign in - Google Accounts - Personal - Microsoft Edge
https://accounts.google.com/gsi/select?client_id=775153634454-f...
Sign in with Google
Choose an account
to continue to TikTik
H Learn
hlearn2022@gmail.com
Hitesh Sachdev
20104137.hiteshsachdev@gmail.com
Use another account
English
Help
Privacy Policy
Terms of Service
Sign in as H
hlearn2022@gmail.com

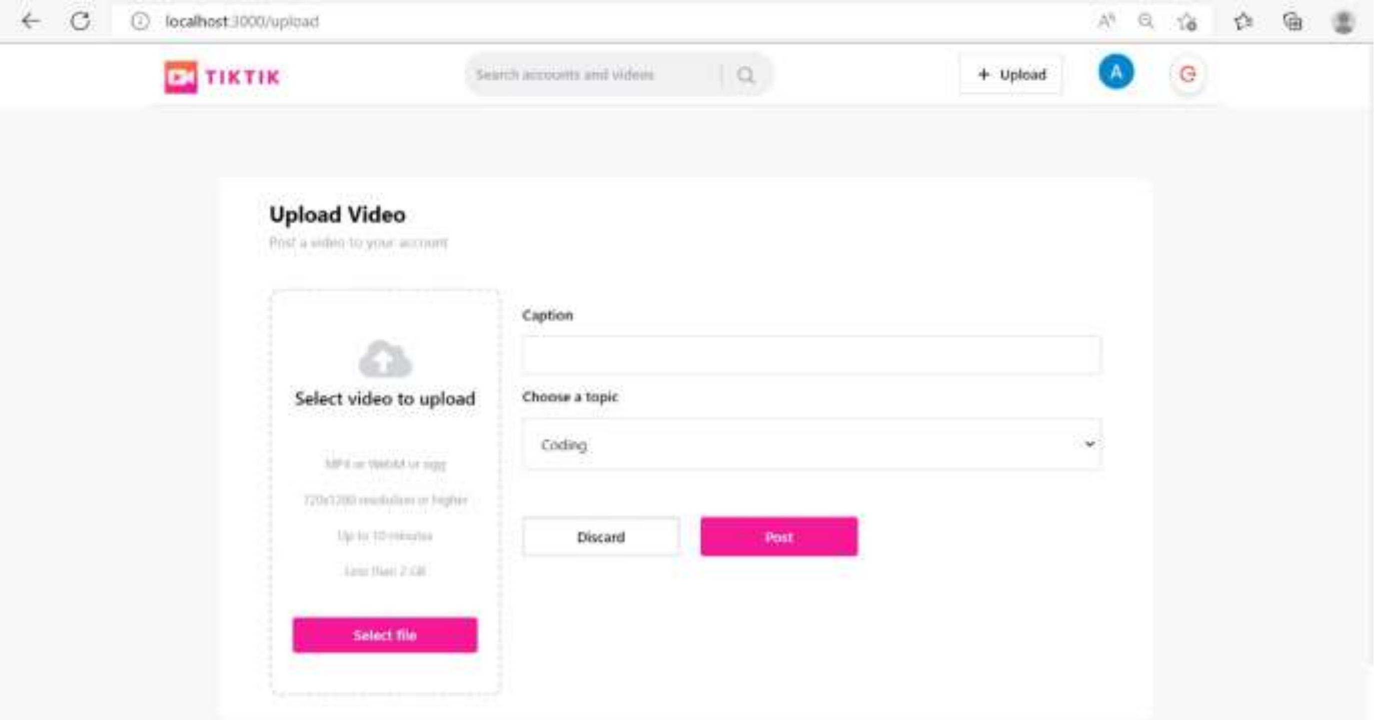


localhost:3000/profile/105474761427608605014
TIKTIK
Search accounts and videos
+ Upload
For You
Popular Topics
Coding
Comedy
Gaming
Food
Dance
Beauty
Animals
Sports
Suggested accounts
hiteshsachdev
Hitesh Sachdev
jsm
JSM
hlearn
H Learn
About Newsroom Store Contact Careers ByteDance
hiteshsachdev
Hitesh Sachdev
Videos
Liked

# Uploading Video

## Upload.tsx

```
const uploadVideo = async (e: any) => {
    const selectedFile = e.target.files[0];
    const fileTypes = ['video/mp4', 'video/webm', 'video/ogg'];

    // uploading asset to sanity
    if (fileTypes.includes(selectedFile.type)) {
      setWrongFileType(false);
      setLoading(true);

      client.assets
        .upload('file', selectedFile, {
          contentType: selectedFile.type,
          filename: selectedFile.name,
        })
        .then((data) => {
          setVideoAsset(data);
          setLoading(false);
        });
    } else {
      setLoading(false);
      setWrongFileType(true);
    }
  };

  const handlePost = async () => {
    if (caption && videoAsset?._id && topic) {
      setSavingPost(true);

      const doc = {
        _type: 'post',
        caption,
        video: {
          _type: 'file',
          asset: {
            _type: 'reference',
            _ref: videoAsset?._id,
          },
        },
        userId: userProfile?._id,
        postedBy: {
          _type: 'postedBy',
          _ref: userProfile?._id,
        },
        topic,
      };

      await axios.post(`${BASE_URL}/api/post`, doc);

      router.push('/');
    }
  };

  const handleDiscard = () => {
    setSavingPost(false);
    setVideoAsset(undefined);
    setCaption('');
    setTopic('');
  };
```



Caption

Choose a topic
Coding

# Commenting & Liking the Post

## Comment.tsx

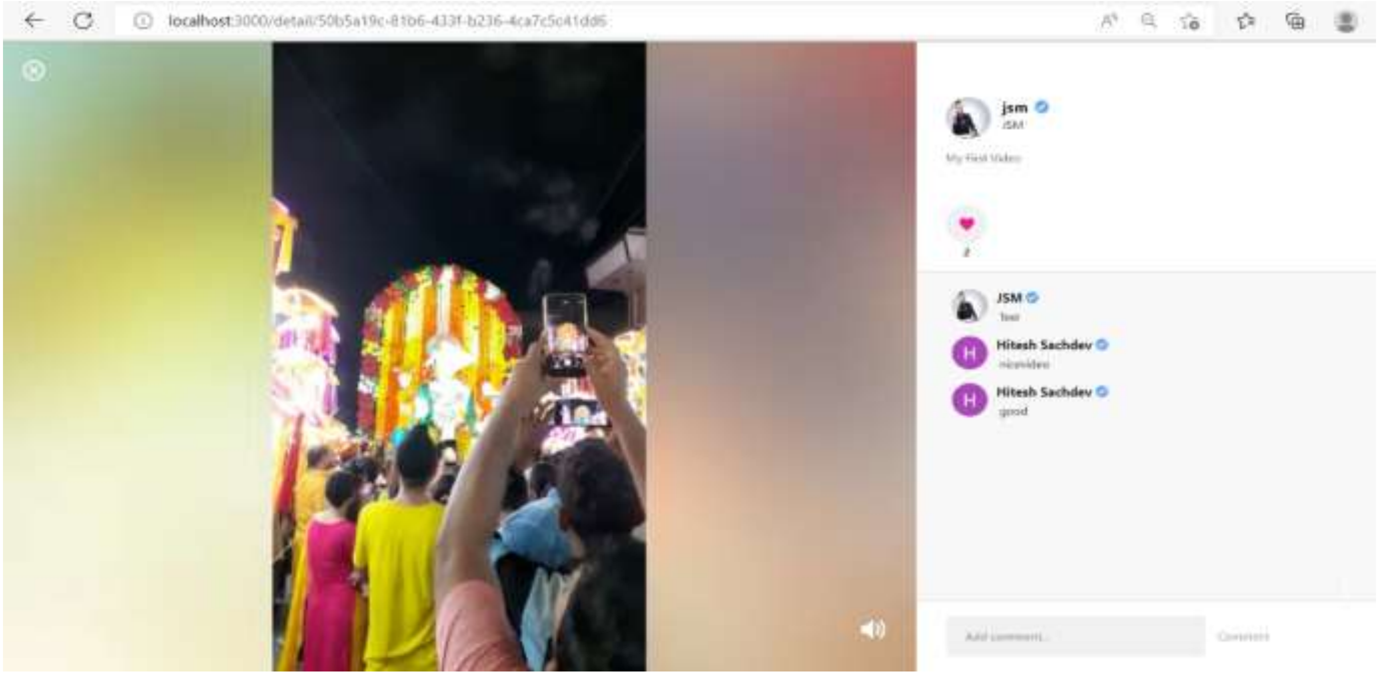
Adding it in comment.tsx file for users to be able to comment.

```
interface IProps {
  isPostingComment: Boolean;
  comment: string;
  setComment: Dispatch<SetStateAction<string>>;
  addComment: (e: React.FormEvent) => void;
  comments: IComment[];
}
interface IComment {
  comment: string;
  length?: number;
  _key: string;
  postedBy: { _ref?: string; _id?: string };
}
const Comments = ({ comment, setComment, addComment, comments, isPostingComment
}: IProps) => {
  const { allUsers, userProfile }: any = useAuthStore();
```

## LikeButton.tsx

This code helps to implement the Like Button Logic

```
interface IProps {
  likes: any;
  flex: string;
  handleLike: () => void;
  handleDislike: () => void;
}

const LikeButton: NextPage<IProps> = ({ likes, flex, handleLike, handleDislike }) => {
  const [alreadyLiked, setAlreadyLiked] = useState(false);
  const { userProfile }: any = useAuthStore();
  let filterLikes = likes?.filter((item: any) => item._ref === userProfile?._id);
 useEffect(() => {
    if (filterLikes?.length > 0) {
      setAlreadyLiked(true);
    } else {
      setAlreadyLiked(false);
    }
  }, [filterLikes, likes]);
```



localhost:3000/detail/50b5a19c-81b6-433f-b236-4ca7c5c41dd6

jsm
JSM

My First Video

2

JSM
Test

Hitesh Sachdev
nicevideo

Hitesh Sachdev
good

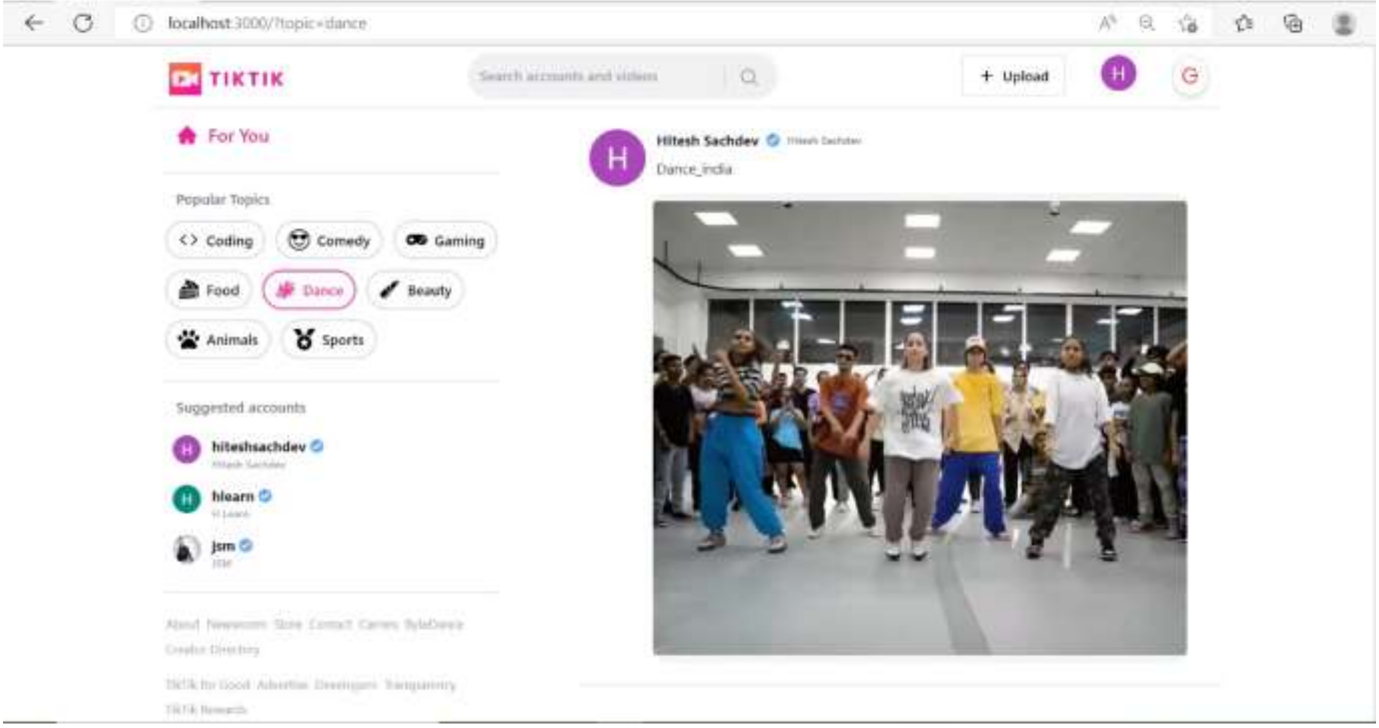
# Search based on Topics

## Constants.tsx

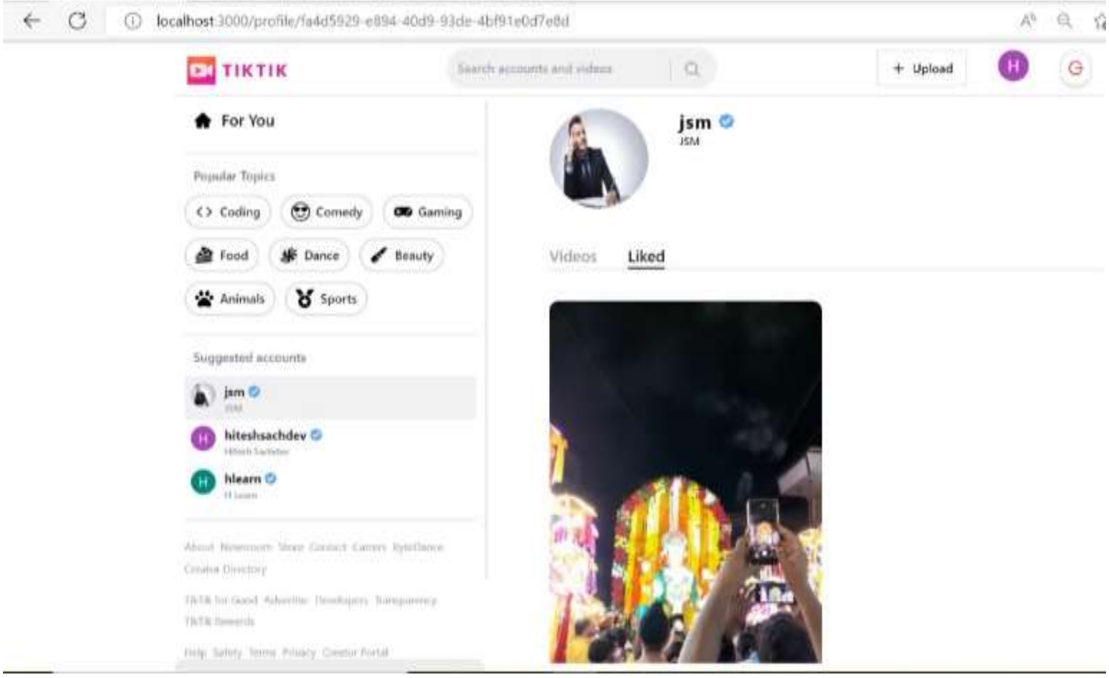
This let user search video based on particular topic

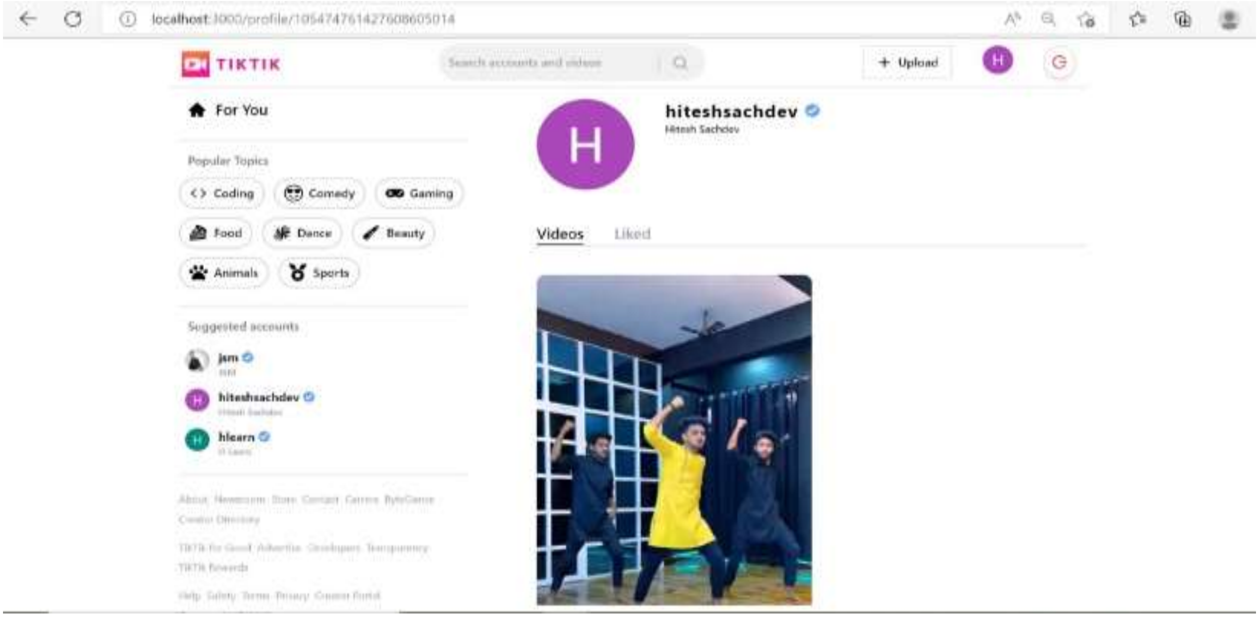
```
export const topics = [
  {
    name: 'coding',
    icon: <BsCode />,
  },
  {
    name: 'comedy',
    icon: <BsEmojiSunglasses />,
  },
  {
    name: 'gaming',
    icon: <FaGamepad />,
  },
  {
    name: 'food',
    icon: <GiCakeSlice />,
  },
  {
    name: 'dance',
    icon: <GiGalaxy />,
  },
  {
    name: 'beauty',
    icon: <GiLipstick />,
  },
  {
    name: 'animals',
    icon: <FaPaw />,
  },
  {
    name: 'sports',
    icon: <FaMedal />,
  },
];
```

For example Dance topic here

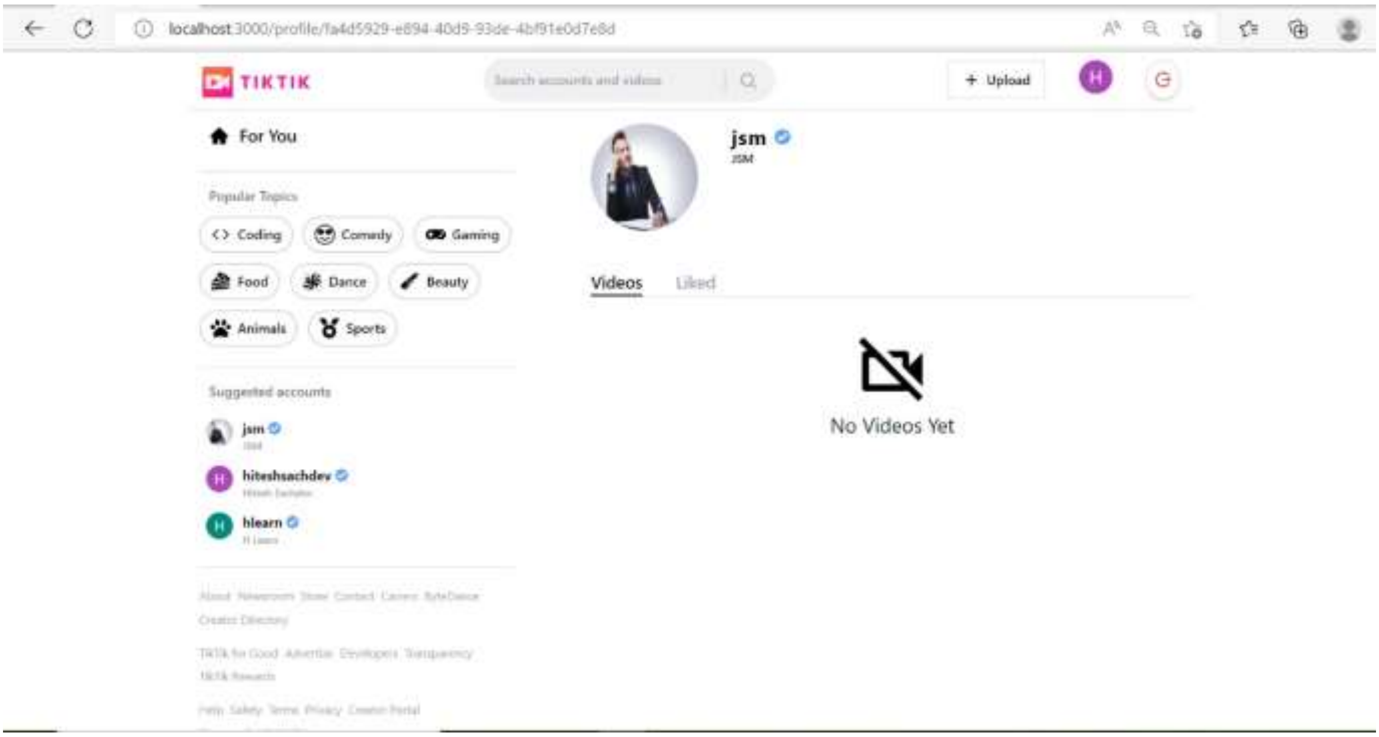


It has feature to go to user profile check what videos they’ve uploaded and liked.


localhost:3000/profile/fa4d5929-e894-40d9-93de-4bf91e0d7e8d
TIKTIK
Search accounts and videos
+ Upload
H
G
For You
Popular Topics
Coding
Comedy
Gaming
Food
Dance
Beauty
Animals
Sports
Suggested accounts
jsm
JSM
hiteshsachdev
Hitesh Sachdev
hlearn
H Learn
About Newsroom Store Contact Careers ByteDance
Creator Directory
TikTik for Good Advertise Developers Transparency
TikTik Rewards
Help Safety Terms Privacy Creator Portal
jsm
JSM
Videos
Liked



And if not video uploaded yet in that case it will show No Videos Yet.
It will also show the accounts which are using this website under Suggested accounts and one can visit their accounts and posts.



# Chapter 9

## Project Scheduling:

| Sr. No | Group Member | Work done |
|---|---|---|
| 1 | Hitesh Sachdev<br>Siddesh Sawan | Implementing 1st module/ functionality:<br>Design User Interface |
| | | Testing 1st module: Documentation |
| 2 | Hitesh Sachdev<br>Aman Yadav | Implementing 2nd module/ functionality: Implementing API (Google Sign-in) |
| 3 | Hitesh Sachdev | Implementing 3rd module/ functionality: Design User Interface and adding components as well connecting API Testing End to End Flow |

# Chapter 10

## Conclusion

- Have setup the website where user can share their precious memories and revisit whenever they want and user can also see the video they've posted or liked on their profile.  User can search memories posted on website. And to add memories like places they are visited or things they have enjoyed.

- Social media will be more integrated into personal, social and business lives. Without realizing it, these platforms will be a natural part of our lives, streamlining our everyday activities and work. Social media will have longer-term implications for individuals as a result of a life lived in public.

- Efforts put to make this website a zone for people to communicate and put their memories in place and revisit them.

## Future Scope

- In future user will able to make their profile public and private and also get option to follow other user profile.

- In future users will able to upload story

# References

- Setting up sanity with the help of sanity doc https://www.sanity.io/docs
- Google Identity Services For login https://www.npmjs.com/package/@react-oauth/google
- Learning aid https://reactjs.org/tutorial/tutorial.html