A Thesis/Project/Dissertation Report
On
Anonymous Social Media Web Application (Thoughts Book)

Submitted in partial fulfillment of the
Requirement for the award of the degree of

Bachelor of Technology
In
Computer Science and Engineering

GALGOTIAS
UNIVERSITY
(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

Under The Supervision of
Mr. Sreenarayanam N M
Professor
Department of Computer Science and Engineering

Submitted By

Pallav Chaudhary 180210100329

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA, INDIA
DECEMBER, 2021

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the thesis/project/dissertation, entitled **"Anonymous Social Media Web Application (Thoughts Book)"** in partial fulfillment of the requirements for the award of the Bachelor Of Technology submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of July, 2021 to December, 2021 under the supervision of Mr. Shreenarayanam N M Professor, Department of Computer Science and Engineering/Computer Application and Information and Science, of School of Computing Science and Engineering , Galgotias University, Greater Noida

The matter presented in the thesis/project/dissertation has not been submitted by me for the award of any other degree of this or any other places.

Pallav Chaudhary 18021010079

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Sreenarayanam N M
Professor

CERTIFICATE

The Final Thesis/Project/ Dissertation Viva-Voice examination of **Pallav Chaudhary: 180211011329** has been held on December, 2021 and his/her work is recommended for the award **BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING.**

Signature of Examiner(s) **Signature of Supervisor(s)**

Signature of Project Coordinator **Signature of Dean**

Date: December, 2021

Place: Greater Noida

Acknowledgement

In the accomplishment of completion of my project on “Anonymous Social Media Web Application (Thoughts Book)” I would like to convey my special gratitude to Mr. Sreenarayanam N M of School of Computer Science and Engineering and as well as Dr. Munish Sabharwal of Galgotias University.

Your valuable guidance and suggestions helped me in various phases of the completion of this project. I will always be thankful to you in this regard.

I would also like to convey my gratitude towards my partner Mr. Pallav Chuadhary without his support and coordination we would not have been able to complete this project.

Pallav Chaudhary

Abstract

Social media is the group of online interchanges channels devoted to local area based input, interaction, content-sharing and joint effort. The purpose for this Project is to give detail data about the effect of social media on society. A ton of studies showed social media has both positive and adverse results. A portion of the positive results are, for instance, Socialization and Correspondence, improve learning openings and getting health related and other important data. Despondency, uneasiness, tormenting, psychological oppression, and crimes are a portion of the negative sides of web-based media on social orders. For the most part, when people groups utilize online media for fitting reason and foreordained objectives the result will be positive and the opposite is valid for antagonistic aftereffects.

This also focuses on social media application that provides anonymity and would be the best for people to come out without being insecure and express themselves openly which the current social Medias don't provide as they are more of showing people off and less of meaningful discussions.

The project uses HTML, CSS, Express JS, and Bootstrap as the Frontend Technologies. Talking about the Backend Technologies it uses Node JS, Express Js and MongoDB as the database.

As the result this application will serve the need of people to interact without being hesitant and will be able to share and views and see other people's thoughts anonymously.

Contents

List of figures –

Acronyms

B.Tech.	Bachelor of Technology
M.Tech.	Master of Technology
BCA	Bachelor of Computer Applications
MCA	Master of Computer Applications
B.Sc. (CS)	Bachelor of Science in Computer Science
M.Sc. (CS)	Master of Science in Computer Science
SCSE	School of Computing Science and Engineering

CHAPTER-1 Introduction

1.1 INTRODUCTION

The internet has become a worldwide commodity with the advent of modernization, reaching millions of individuals in their own space yet, as social media and intermediary platforms have become an integral part of our lives, their utility is becoming increasingly complex. The goal of this project is to build a more secure environment that not only fosters individuality but also fosters a more personal community for those who desire to remain anonymous on global online platforms.

When it comes to using internet platforms, privacy is crucial. This program not only protects and respects a person's privacy concerns, but it also provides an environment in which people can express themselves freely without fear of being judged, increasing morale and interpersonal communication abilities. The idea is creating an application which is all inclusive, safe, and unique and at the same time safeguarding the biggest concerns of any person while using the internet that is rights of privacy.

This Project also focuses on how the modern social media has been shaping and transforming our society in both positive and negative manner and what the consequences are of these advancements, as we have seen and known that social media affects our lives majorly now and has become a routine for most of the people.

1.2 PROJECT DESCRIPTION

This project is aimed to developing an online Social media application. The entire project has been developed keeping in view of the distributed client server computing technology, in mind. This system is developed keeping in mind the unexpressed thoughts of people which remain unheard and through this application any person who wants to express his/her feelings and thoughts can do so anonymously without the risk of being identified. Moreover this application can be a platform of discussion for not so talked about open issues.

Anyone can login or register on the platform and no personal information is required. The project has been planned to be having the view of distributed architecture, with centralized storage of the database. The application for the storage of the data has been planned. Using the constructs of MongoDB Atlas Server and all the user interfaces have been designed using the Node or (MERN) related technologies.

The database connectivity is planned using the mongoose methodology. The standards of security and data protective mechanism have been given a big choice for proper usage. The application takes care of different modules and their associated tasks, which are produced as per the applicable strategies and standards that are put forwarded by the administrative staff. The entire project has been developed keeping in view of the distributed client server computing technology, in mind. The user interfaces are browser specific to give distributed accessibility for the overall system. The internal database has been selected as MongoDB and the application is hosted on HEROKU.

The basic constructs of table spaces, clusters and indexes have been exploited to provide higher consistency and reliability for the data storage. The MongoDB database was a choice as it provides the constructs of high-level reliability and security. The total front end was dominated using the MERN technologies. At all proper levels high care was taken to check that the system manages the data consistency with proper business rules or validations.

The database connectivity was planned using the latest mongoose technology. The authentication and authorization were crosschecked at all the relevant stages. The encryption methods are DES which along with added salts gives high level authentication security.

1.3 PROBLEM FORMULATION:

The internet has become a worldwide commodity with the advent of modernization, reaching millions of individuals in their own space yet, as social media and intermediary platforms have become an integral part of our lives, their utility is becoming increasingly complex. A more secure environment that not only fosters individuality but also fosters a more personal community for those who desire to remain anonymous on global online platforms. When it comes to using internet platforms, privacy is crucial. This program not only protects and respects a person’s privacy concerns, but it also provides an environment in which people can express themselves freely without fear of being judged, increasing morale and interpersonal communication abilities. Several supporting arguments demonstrating how social media, particularly Facebook and Instagram can cause psychological issues. It is undeniable that social media has negative personal consequences for young people, allowing them to over-analyze and criticize themselves and their problems. People today and especially youngsters who use these platforms see other peoples successful and lively “online life” makes them feel less of themselves and they overthink and this can cause depression.

1.4 PURPOSE:

The purpose is to develop an online Social media application which will allow its users to be freer and will let them open up without being identified which mos of the prevailing products don't provide. This system is developed keeping in mind the unexpressed thoughts of people which remain unheard and through this application any person who wants to express his/her feelings and thoughts can do so anonymously without the risk of being identified. Moreover this application can be a platform of discussion for not so talked about open issues

1.5 SCOPE:

The application is a way for people across the world to come together at a place anonymously and have the insight of what people actually think rather than what they generally show. While developing the system, there shall be a space for further modification. There shall be a proper documentation so that further enhancement becomes easy.

As a whole, the system is focused to create a worry-free space in the times of need and an addition modification, it can also be modified for better user communication and activities which can be introduced further.

1.6 TECHNOLOGIES USED:

1.4.1 Front-end:

- HTML
- CSS
- JavaScript
- Bootstrap

2.4.1 Back-end:

- Node.js
- Express

3.4.1 Database:

- MongoDB

4.4.1 Server:

- Heroku

CHAPTER-2 Literature Survey

The system we present here is a public interface for all its users to come together at a virtual place to share and receive ideas, thoughts or things that people generally are scared to on other existing platforms because of the identity being exposed on those platforms. This application provides a safe environment for its users using the latest and most secure technologies available in the domain.

Besides these, there are ample scopes to improve this application. Some more features can be added to establish this application for a social networking application on larger grounds and can perform drastically well as per the usage required.

According to Junco et al., (2010), "social media are a collection of internet websites, services, and practices that support collaboration, community building, participation, and sharing".

As Andres (2010) referred to in Ghulam et al., 2014 depicted in his review that Social media is a bunch of web put together applications that develops with respect to the philosophical and innovative establishment of marry and that grant the plan and trade of client produced content.

So social media has definitions of various kinds from various people and can differ according to different people and how they see its advantages and disadvantages.

In the last few decades, cyberbullying has become a major issue among teenagers, as it allows bullies to humiliate their victims by posting things in front of their peers. Bullying is defined as an aggressive act committed repeatedly and over time by a group or an individual against a victim who is unable to defend himself or herself (Bannink et al., 2014). According to Campbell (2005), the use of the Internet and mobile phones has resulted in the emergence of a new form of bullying known as 'cyberbullying.' Aggression occurs via electronic means, such as the Internet and, particularly, social media, in cyberbullying (Bannink et al., 2014).

2.1 EXISTING SYSTEM

In existing there is no proper care about the people's privacy fully when it comes to "deploying their personal thoughts on the internet", for the social media world to be a place where people can express their thoughts openly is quite shallow as we can witness people being victims of cyber bullying and receive open threats from people or large groups which defeats the purpose of one's intention to post something.

Twitter for example is an app where people express their opinions but we have witnessed many cases where people’s lives and privacy have been put on line because some people don’t like what the others say or get offended and then there are consequences.

Other big Social media platforms are much about people or the users sharing their lives and kind of share information about themselves on their timelines or profiles. This all information being exposed does not give them the freedom to share openly so the existing systems are jut platforms to share and show off details rather than open platforms for discussions.

CHAPTER-3 Feasibility Study

Preliminary investigation examines project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

- Technical Feasibility
- Operation Feasibility
- Economic Feasibility

3.1 TECHNICAL FEASIBILITY

The technical issue usually raised during the feasibility stage of the investigation includes the following:

- Does the necessary technology exist to do what is suggested?
- Do the proposed equipment's have the technical capacity to hold the data required to use the new system?
- Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?
- Can the system be upgraded if developed?
- Are there technical guarantees of accuracy, reliability, ease of access and data security?

Earlier no system existed to cater to the needs of 'Secure Infrastructure Implementation System'. The current system developed is technically feasible. It is a web-based user interface for audit workflow at NIC-CSD. Thus, it provides an easy access to the users.

The database's purpose is to create, establish and maintain a workflow among various entities in order to facilitate all concerned users in their various capacities or roles. Permission to the users would be granted based on the roles specified. Therefore, it provides the technical guarantee of accuracy, reliability and security.

The software and hard requirements for the development of this project are not many and are already available in-house at NIC or are available as free as open source. The work for the project is done with the current equipment and existing software technology. Necessary bandwidth exists for providing fast feedback to the users irrespective of the number of users using the system.

3.2 OPERATIONAL FEASIBILITY

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization's operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. Some of the important issues raised are to test the operational feasibility of a project includes the following: -

- Is there sufficient support for the management from the users?
- Will the system be used and work properly if it is being developed and implemented?
- Will there be any resistance from the user that will undermine the possible application benefits?

This system is targeted to be in accordance with the above-mentioned issues. Beforehand, the management issues and user requirements have been taken into consideration. So, there is no question of resistance from the users that can undermine the possible application benefits.

The well-planned design would ensure the optimal utilization of the computer resources and would help in the improvement of performance status.

3.3 ECONOMIC FEASIBILITY

A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economic feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs.

The system is economically feasible. It does not require any addition hardware or software. Since the interface for this system is developed using the existing resources and technologies available at NIC, There is nominal expenditure and economic feasibility for certain.

CHAPTER- 4 System Analysis

4.1 SOFTWARE REQUIREMENT SPECIFICATION (SRS)

The software application is designed for working in the web from any device and can be accessed using any browser from a remote location.

4.1.1 Introduction

Purpose:

The main purpose for preparing this document is to give a general insight into the analysis and requirements of the existing system or situation and for determining the operating characteristics of the system.

Scope:

This Document plays a vital role in the software development life cycle (SDLC) and it describes the complete requirement of the system. It is meant for use by the developers and will be the basic during testing phase. Any changes made to the requirements in the future will have to go through formal change approval process.

4.1.2 Developers Responsibilities Overview:

The developer is responsible for:

- Developing the system, which meets the SRS and solving all the requirements of the system?
- Demonstrating the system and installing the system at client's location after the acceptance testing is successful.
- Submitting the required user manual describing the system interfaces to work on it and also the documents of the system.
- Conducting any user training that might be needed for using the system.
- Maintaining the system for a period of one year after installation.

4.1.3 Modules

Administration:

In this module the Administrator has the privileges to add all the Blood Groups, add donors, update contact info, manage queries and manage request for blood. He can search all the info about the Donor.

Functionality

- Creation of account.
- Logging into the account securely.
- View the thoughts or ideas of other users.
- Share thoughts or ideas.
- Edit the thoughts shred by user.
- Log out securely.

Authentication:

Authentication is the first priority which here is handled through Passport in NodeJs as it provides Auth0. Auth0 is a global leader in Identity-as-a-Service (IDaaS). Its extensible platform seamlessly authenticates and secures more than 2.5 billion logins per month, making it loved by developers and trusted by global enterprises.

4.2 HARDWARE REQUIREMENTS

- Intel i3 2.8 GHz Processor and Above
- RAM 512MB and Above
- HDD 20 GB Hard Disk Space and Above

4.3 SOFTWARE REQUIREMENTS

- WINDOWS OS (2007 and above) or MAC OS or linux OS
- Visual Studio Code 2005 Enterprise Edition
- Nodejs and NPM. And all related modules.
- MongoDB Atlas
- Herokiu Server

CHAPTER-5 Project Design

5.1 SOFTWARE DEVELOPMENT LIFE CYCLE MODEL (WATERFALL MODEL):

The waterfall model was selected as the SDLC model due to the following reasons:

- Requirements were very well documented, clear and fixed.
- Technology was adequately understood.
- Simple and easy to understand and use.
- There were no ambiguous requirements.
- Easy to manage due to the rigidity of the model. Each phase has specific deliverables and a review process.
- Easy to arrange tasks.

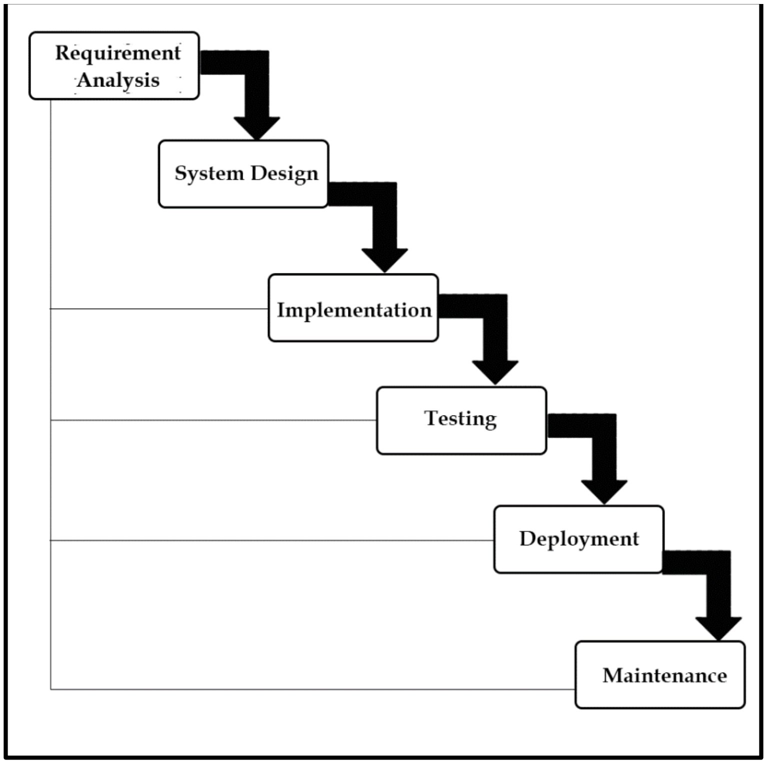


Fig 1: Waterfall Diagram

5.2 USE-CASE DIAGRAM:

Use case diagram is a platform that can provide a common understanding for the end-users, developers and the domain experts. It is used to capture the basic functionality i.e., use cases, and the users of those available functionality, i.e., actors, from a given problem statement. In this experiment, we will learn how use cases and actors can be captured and how different use cases are related in a system.

5.2.1 Use case diagrams:

Use case diagrams belong to the category of behavioral diagram of UML diagrams. Use case diagrams aim to present a graphical overview of the functionality provided by the system. It consists of a set of actions (referred to as use cases) that the concerned system can perform one or more actors, and dependencies among them.

5.2.2 Actor:

An actor can be defined as an object or set of objects, external to the system, which interacts with the system to get some meaningful work done. Actors could be human, devices, or even other systems. For example, consider the case where a customer withdraws cash from an ATM. Here, customer is a human actor. Actors can be classified as below:

- **Primary actor**: They are principal users of the system, who fulfill their goal by availing some service from the system. For example, a customer uses an ATM to withdraw cash when he needs it. A customer is the primary actor here.
- **Supporting actor**: They render some kind of service to the system. "Bank representatives", who replenishes the stock of cash, is such an example. It may be noted that replenishing stock of cash in an ATM is not the prime functionality of an ATM.

5.2.3 Use Case:

A use case is simply functionality provided by a system. Continuing with the example of the ATM, withdraw cash is a functionality that the ATM provides. Therefore, this is a use case. Other possible use cases include, check balance, change PIN, and so on. Use cases include both successful and unsuccessful scenarios of user interactions with the system. For example, authentication of a customer by the ATM would fail if he enters wrong PIN. In such case, an error message is displayed on the screen of the ATM.

5.2.4. Subject:

Subject is simply the system under consideration. Use cases apply to a subject. For example, an ATM is a subject, having multiple use cases, and multiple actors interact with it. However, one should be careful of external systems interacting with the subject as actors.

5.2.5. Guidelines for drawing Use Case diagrams:

Following general guidelines could be kept in mind while trying to draw a use case diagram:

- Determine the system boundary
- Ensure that individual actors have well-defined purpose
- Use cases identified should let some meaningful work done by the actors
- Associate the actors and use cases -- there shouldn't be any actor or use case floating without any connection
- Use include relationship to encapsulate common behavior among use cases, if any

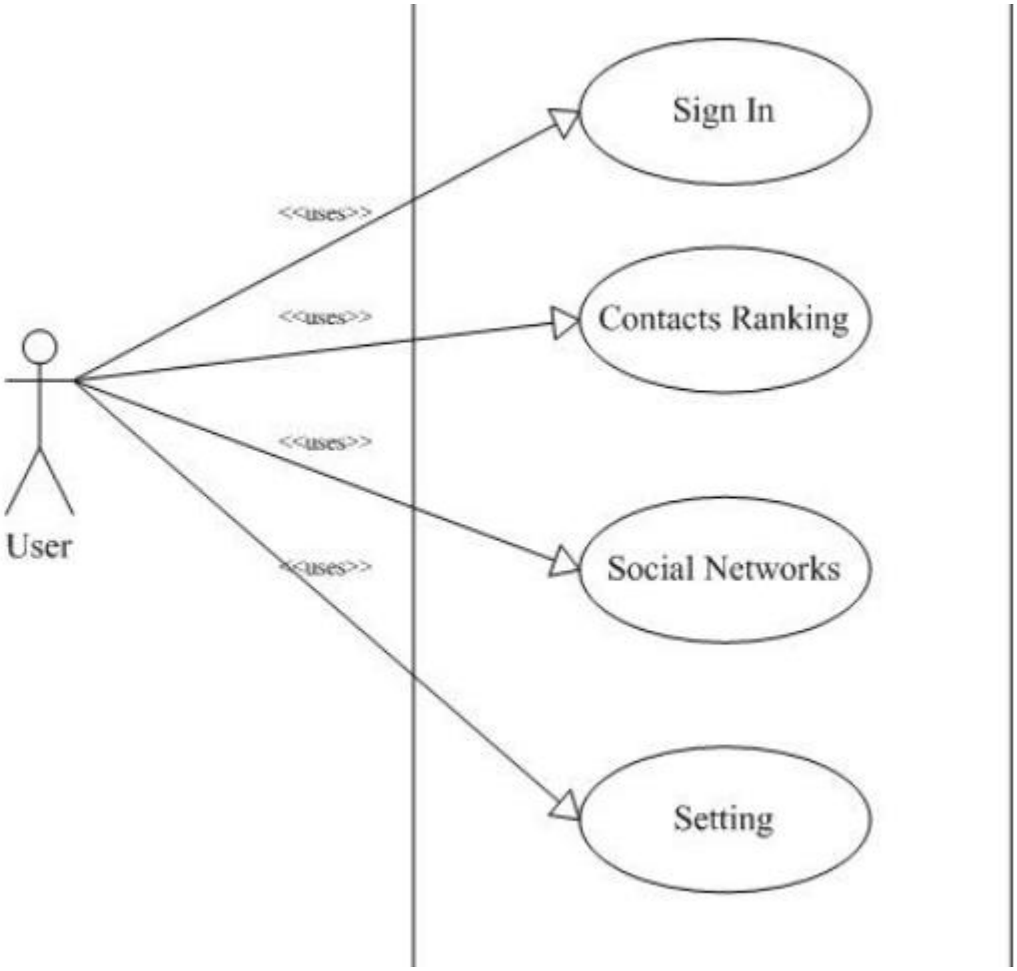


Fig 2: Use Case Diagram

5.3 DATA FLOW DIAGRAM:

A data flow diagram is graphical tool used to describe and analyze movement of data through a system. These are the central tool and the basis from which the other components are developed. The transformation of data from input to output, through processed, may be described logically and independently of physical components associated with the system. These are known as the logical data flow diagrams.

The physical data flow diagrams show the actual implements and movement of data between people, departments and workstations. A full description of a system actually consists of a set of data flow diagrams. Using two familiar notations Yourdon, Gane and Sarson notation develops the data flow diagrams. Each component in a DFD is labeled with a descriptive name. Process is further identified with a number that will be used for identification purpose.

The development of DFD'S is done in several levels. Each process in lower-level diagrams can be broken down into a more detailed DFD in the next level. The lop-level diagram is often called context diagram. It consists a single process bit, which plays vital role in studying the current system. The process in the context level diagram is exploded into other process at the first level DFD.

The idea behind the explosion of a process into more process is that understanding at one level of detail is exploded into greater detail at the next level. This is done until further explosion is necessary and an adequate amount of detail is described for analyst to understand the process.

A DFD is also known as a "bubble Chart" has the purpose of clarifying system requirements and identifying major transformations that will become programs in system design. So, it is the starting point of the design to the lowest level of detail. A DFD consists of a series of bubbles joined by data flows in the system.

5.3.1 DFD Symbols

In the DFD, there are four symbols

- A square defines a source(originator) or destination of system data

- An arrow identifies data flow. It is the pipeline through which the information flows
- A circle or a bubble represents a process that transforms incoming data flow into outgoing data flows.
- An open rectangle is a data store, data at rest or a temporary repository of data

-------------------->

Process that transforms data flow.

▭ Source or Destination of data

→ Data flow

5.3.2 Constructing A DFD

Several rules of thumb are used in drawing DFD'S:

- Process should be named and numbered for an easy reference. Each name should be representative of the process.
- The direction of flow is from top to bottom and from left to right. Data traditionally flow from source to the destination although they may flow back to the source. One way to indicate this

is to draw long flow line back to a source. An alternative way is to repeat the source symbol as a destination. Since it is used more than once in the DFD it is marked with a short diagonal.

- When a process is exploded into lower-level details, they are numbered.
- The names of data stores and destinations are written in capital letters. Process and dataflow names have the first letter of each work capitalized

A DFD typically shows the minimum contents of data store. Each data store should contain all the data elements that flow in and out.

Questionnaires should contain all the data elements that flow in and out. Missing interfaces redundancies and like is then accounted for often through interviews.

5.3.3 Salient Features Of DFD's

- The DFD shows flow of data, not of control loops and decision are controlled considerations do not appear on a DFD.
- The DFD does not indicate the time factor involved in any process whether the dataflow take place daily, weekly, monthly or yearly.
- The sequence of events is not brought out on the DFD.

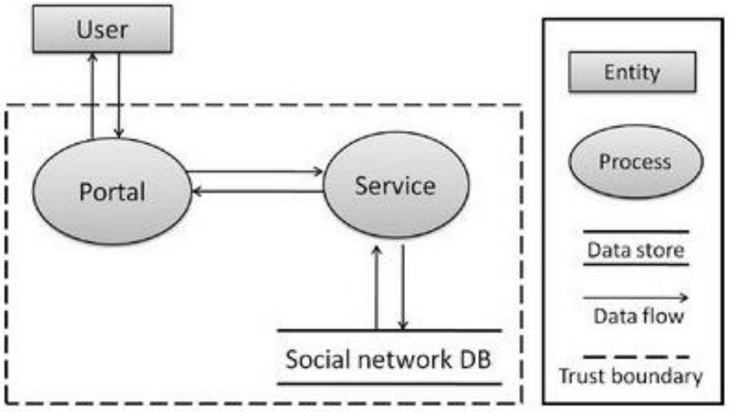


Fig. 3 Data Flow Diagram

CHAPTER-6 Software Development Environment

6.1 UI INTERFACE DESIGN

The user interface (UI) is the point at which human users interact with a computer, website or application. The goal of effective UI is to make the user's experience easy and intuitive, requiring minimum effort on the user's part to receive maximum desired outcome. UI is created in layers of interaction that appeal to the human senses (sight, touch, auditory and more). They include both input devices like keyboard, mouse, trackpad, microphone, touch screen, fingerprint scanner, e-pen and camera and output devices like monitors, speakers and printers. Devices that interact with multiple senses are called "multimedia user interfaces". For example, everyday UI uses a combination of tactile input (keyboard and mouse) and a visual and auditory output (monitor and speakers).

User interface design is concerned with the dialogue between user and computer. It is concerned with everything from starting the system or logging into the system to eventually presentation of desired inputs and outputs. The overall flow of screens messages is called a dialogue.

6.1.1 Guidelines for an effective UI

Everything stems from knowing your users, including understanding their goals, skills, preferences, and tendencies. Once you know about your user, make sure to consider the following when designing the interface:

- **Keep the interface simple:**
  The best interfaces are almost invisible to the user. They avoid unnecessary elements and are clear in the language they use on labels and in messaging.
- **Create consistency and use common UI elements:**
  By using common elements in your UI, users feel more comfortable and are able to get things done more quickly. It is also important to create patterns in language, layout and design throughout the site to help facilitate efficiency. Once a user learns how to do something, they should be able to transfer that skill to other parts of the site.

- **Be purposeful in page layout:**
  Consider the spatial relationships between items on the page and structure the page based on importance. Careful placement of items can help draw attention to the most important pieces of information and can aid scanning and readability.
- **Strategically use color and texture:**
  You can direct attention toward or redirect attention away from items using color, light, contrast, and texture to your advantage.
- **Use typography to create hierarchy and clarity:**
  Carefully consider how you use typeface. Different sizes, fonts, and arrangement of the text to help increase ability to scan, legibility and readability.
- **Make sure that the system communicates what's happening:**
  Always inform your users of location, actions, changes in state, or errors. The use of various UI elements to communicate status and, if necessary, next steps can reduce frustration for your user.
- **Think about the defaults:**
  By carefully thinking about and anticipating the goals people bring to your site, you can create defaults that reduce the burden on the user. This becomes particularly important when it comes to form design where you might have an opportunity to have some fields pre-chosen or filled out.

6.2 Introduction to NodeJS

Node.js is an open-source and cross-platform JavaScript runtime environment. It is a popular tool for almost any kind of project!

Node.js runs the V8 JavaScript engine, the core of Google Chrome, outside of the browser. This allows Node.js to be very performant.

A Node.js app runs in a single process, without creating a new thread for every request. Node.js provides a set of asynchronous I/O primitives in its standard library that prevent JavaScript code from blocking and generally, libraries in Node.js are written using non-blocking paradigms, making blocking behavior the exception rather than the norm.

When Node.js performs an I/O operation, like reading from the network, accessing a database or the filesystem, instead of blocking the thread and wasting CPU cycles waiting, Node.js will resume the operations when the response comes back.

This allows Node.js to handle thousands of concurrent connections with a single server without introducing the burden of managing thread concurrency, which could be a significant source of bugs.

Node.js has a unique advantage because millions of frontend developers that write JavaScript for the browser are now able to write the server-side code in addition to the client-side code without the need to learn a completely different language.

In Node.js the new ECMAScript standards can be used without problems, as you don't have to wait for all your users to update their browsers - you are in charge of deciding which ECMAScript version to use by changing the Node.js version, and you can also enable specific experimental features by running Node.js with flags.

Express: It provides one of the simplest yet powerful ways to create a web server. Its minimalist approach, un-opinionated, focused on the core features of a server, is key to its success.

6.2.1 Web Development

PHP is widely used in web development nowadays. PHP can develop dynamic websites easily. But you must have the basic the knowledge of following technologies for web development as well.

- HTML
- CSS
- JavaScript
- Bootstrap
- XML and JSON
- jQuery

6.2.2 Prerequisite

Before learning Node, you must have the basic knowledge of HTML, CSS, and JavaScript**.** So, learn these technologies for better implementation of NodeJS.

- **HTML -** HTML is used to design static webpage.
- **CSS -** CSS helps to make the webpage content more effective and attractive.
- **JavaScript -** JavaScript is used to design an interactive website.

6.3 CLIENT APPLICATION DEVELOPMENT

Client applications are the closest to a traditional style of application in Windows-based programming. These are the types of applications that display windows or forms on the desktop, enabling a user to perform a task. Client applications include applications such as word processors and spreadsheets, as well as custom business applications such as data-entry tools, reporting tools, and so on. Client applications usually employ windows, menus, buttons, and other GUI elements, and they likely access local resources such as the file system and peripherals such as printers.

Another kind of client application is the traditional ActiveX control (now replaced by the managed Windows Forms control) deployed over the Internet as a Web page. This application is much like other client applications: it is executed natively, has access to local resources, and includes graphical elements.

Express: It provides one of the simplest yet powerful ways to create a web server. Its minimalist approach, un-opinionated, focused on the core features of a server, is key to its success.

6.3.1 Introduction to DBMS

Welcome to the MongoDB 5.0 Manual! MongoDB is a document database designed for ease of development and scaling. The Manual introduces key concepts in MongoDB, presents the query language, and provides operational and administrative considerations and procedures as well as a comprehensive reference section.

- MongoDB offers both local and cloud-hosted deployment options:

- For locally hosted deployments, MongoDB offers both a Community and an Enterprise version of the database:
- MongoDB Community is the source available and free to use edition of MongoDB.
  MongoDB Enterprise is available as part of the MongoDB Enterprise Advanced subscription

and includes comprehensive support for your MongoDB deployment. MongoDB Enterprise also adds enterprise-focused features such as LDAP and Kerberos support, on-disk encryption, and auditing.

- MongoDB Atlas is a hosted MongoDB Enterprise service option in the cloud which requires no installation overhead and offers a free tier to get started.
- Document Database
- A record in MongoDB is a document, which is a data structure composed of field and value pairs. MongoDB documents are similar to JSON objects. The values of fields may include other documents, arrays, and arrays of documents.

6.3.2 How MongoDb works?

Key Features

High Performance

MongoDB provides high performance data persistence. In particular,

Support for embedded data models reduces I/O activity on database system.

Indexes support faster queries and can include keys from embedded documents and arrays.

Rich Query Language

MongoDB supports a rich query language to support read and write operations (CRUD) as well as:

Data Aggregation

Text Search and Geospatial Queries.

SQL to MongoDB Mapping Chart

SQL to Aggregation Mapping Chart

Learn about the latest query language features with the MongoDB Query Language: What's New presentation from MongoDB.live 2020.

High Availability

MongoDB's replication facility, called replica set, provides: Automatic failover data redundancy.

A replica set is a group of MongoDB servers that maintain the same data set, providing redundancy and increasing data availability.

Horizontal Scalability

MongoDB provides horizontal scalability as part of its core functionality:

Sharding distributes data across a cluster of machines.

Starting in 3.4, MongoDB supports creating zones of data based on the shard key. In a balanced cluster, MongoDB directs reads and writes covered by a zone only to those shards inside the zone. See the Zones manual page for more information.

CHAPTER-7 Proposed System

7.1 ***SYSTEM FUNCTIONALITIES***

- The application is a web based application which is made on Node.js and MongoDB Atlas serves as the database for storing both the user data and their credentials.

- A safe registration and login which allows users to make an account or if they already have log in securely and the data they provide here is encrypted using the advanced encryption technique AES and then thereafter a salt is added too so it can in no way be possible to decrypt which ensures the privacy of users at the application level.

- Users once logged in will be able to see the thoughts put in by other users and can now share their own thoughts too.

- This application uses Express JS along with HTML and CSS for frontend and is managed and powered by Node.js in the backend.

- express: a minimalist framework for building web applications

- bodyparser: for parsing data in the body of requests sent to the server

- mongoose: A DRM (Data relation management) for managing mongodb database

- AES encryption: AES-256 uses a 256-bit key length to encrypt and decrypt a block of messages.

- Each cipher encrypts and decrypts data in blocks of bits using cryptographic keys of 256 bits, Symmetric, also known as secret key, ciphers use the same key for encrypting and decrypting. The sender and the receiver must both know -- and use -- the same secret key.

- The addition of salts to data which are long values which further encrypt the data and make it more complex to break.

All this combines to make a web application which is safe and secure and its users can openly express and share thoughts they were not able to on the conventional platforms using the modern technologies in the application development field and also making use of the secure encryption standards.

CHAPTER-8 Testing

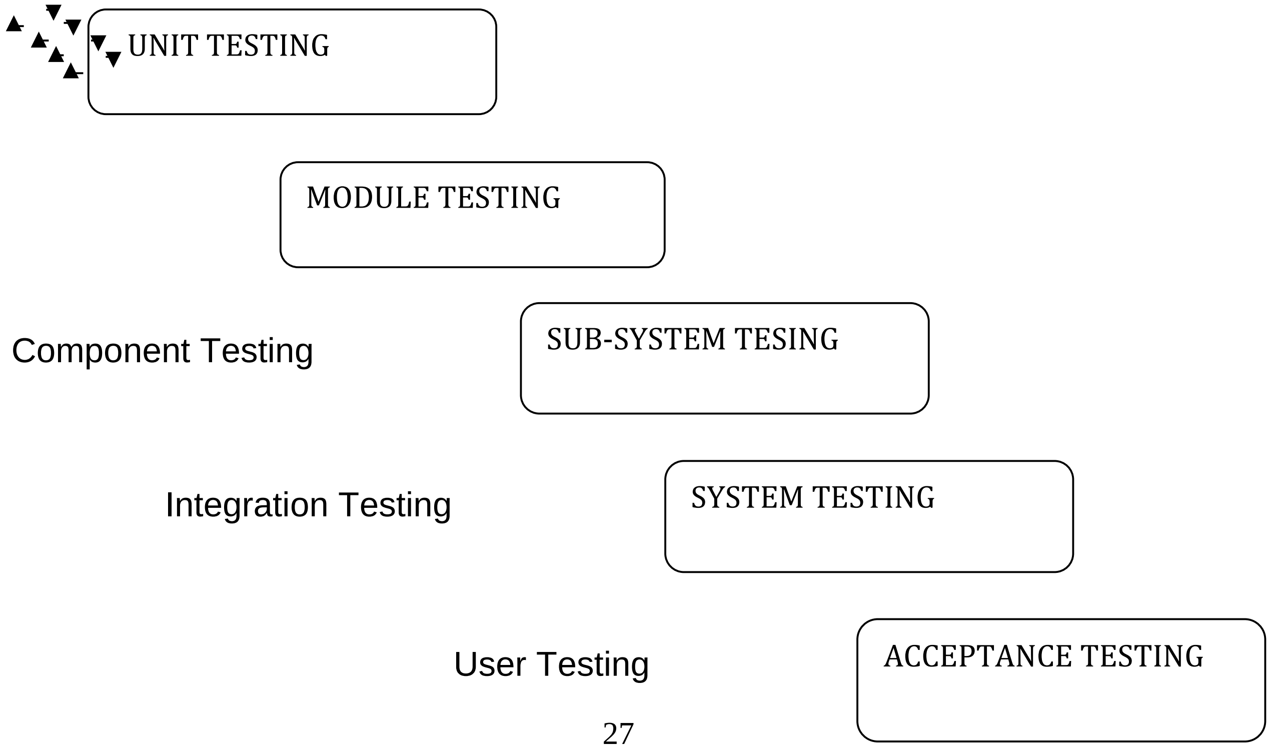
8.1 INTRODUCTION

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. In fact, testing is the one step in the software engineering process that could be viewed as destructive rather than constructive.

A strategy for software testing integrates software test case design methods into a well-planned series of steps that result in the successful construction of software. Testing is the set of activities that can be planned in advance and conducted systematically. The underlying motivation of program testing is to affirm software quality with methods that can economically and effectively apply to both strategic to both large and small-scale systems.

8.2 STRATEGIC APPROACH TO SOFTWARE TESTING

The software engineering process can be viewed as a spiral. Initially system engineering defines the role of software and leads to software requirement analysis where the information domain, functions, behavior, performance, constraints and validation criteria for software are established. Moving inward along the spiral, we come to design and finally to coding. To develop computer software, we spiral in along streamlines that decrease the level of abstraction on each turn.



8.3 UNIT TESTING

Unit testing focuses verification effort on the smallest unit of software design, the module. The unit testing, we have is white box oriented and some modules the steps are conducted in parallel.

8.3.1 White Box Testing

This type of testing ensures that

- All independent paths have been exercised at least once
- All logical decisions have been exercised on their true and false sides
- All loops are executed at their boundaries and within their operational bounds
- All internal data structures have been exercised to assure their validity.

To follow the concept of white box testing we have tested each form .we have created independently to verify that Data flow is correct, All conditions are exercised to check their validity, All loops are executed on their boundaries.

8.3.2 Conditional Testing

In this part of the testing each of the conditions were tested to both true and false aspects. And all the resulting paths were tested. So that each path that may be generate on particular condition is traced to uncover any possible errors.

8.3.3 Data Flow Testing

This type of testing selects the path of the program according to the location of definition and use of variables. This kind of testing was used only when some local variables were declared. The definition-use chain method was used in this type of testing. These were particularly useful in nested statements.

8.3.4 Loop Testing

In this type of testing all the loops are tested to all the limits possible. The following exercise was adopted for all loops:

- All the loops were tested at their limits, just above them and just below them.

- All the loops were skipped at least once.
- For nested loops test the inner most loop first and then work outwards.
- For concatenated loops the values of dependent loops were set with the help of connected loop.
- Unstructured loops were resolved into nested loops or concatenated loops and tested as above.

Each unit has been separately tested by the development team itself and all the input have been validated.

CHAPTER-9 Conclusion and Future Works

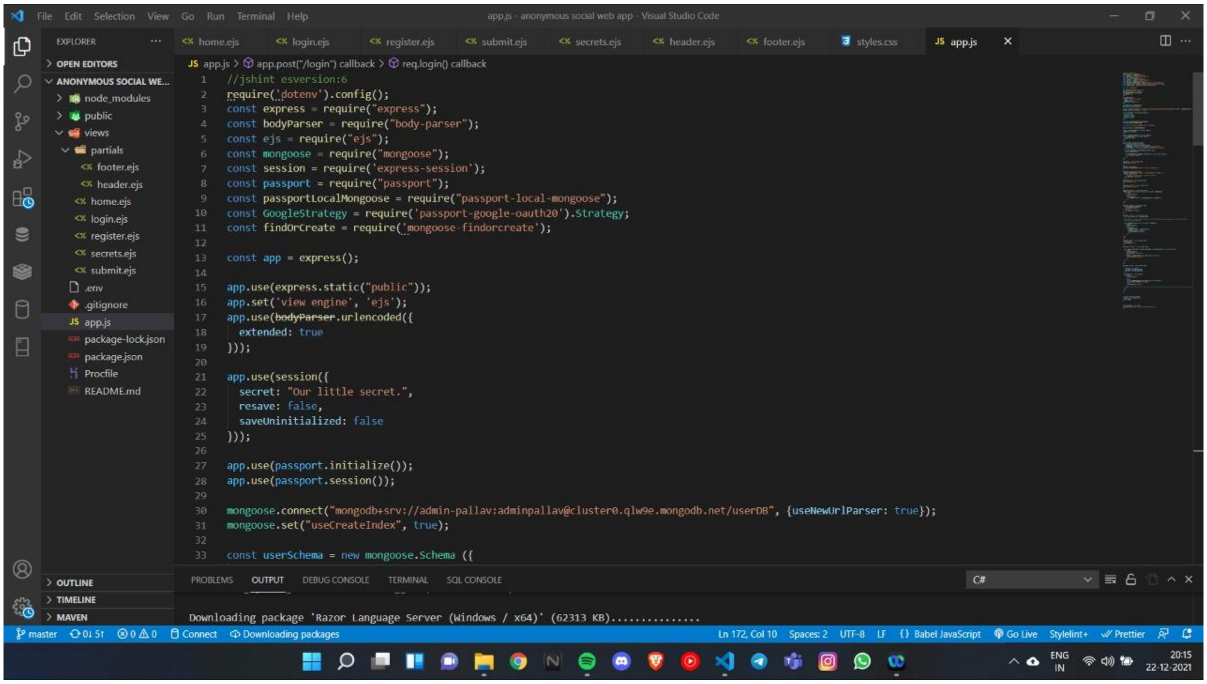
9.1 Conclusion

As the result the project shows what social media is and how it has affected the lives of people in the modern human society and what are its positive and negative impacts and provides a proposed solution as an anonymous social media application which is a project focused on creation of an anonymous platform for users to share their thoughts and see other peoples thoughts which is not there in the social media applications as of now. Like any other project this is also a first level prototype which can further be developed and can have many other functionalities like functions where users can interact directly interact and share their views whilst being anonymous. The end result is a web application which functions on good quality security and authentication for the usage required.

After observing the growth, usefulness and the user-acceptance of the application, we would be expanding the functionalities of the application and launching more features which will allow further enhancements of the application and make it more immense and user friendly.

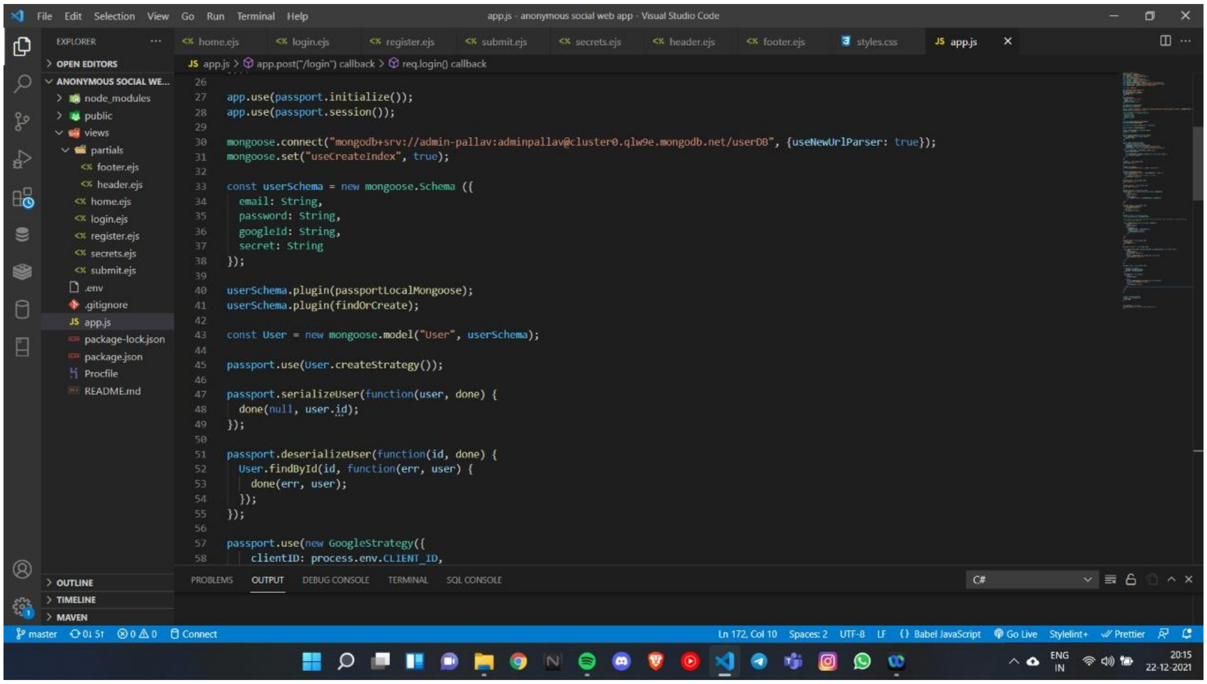
CHAPTER-10 Source Code and Product Output

10.1 Source code-

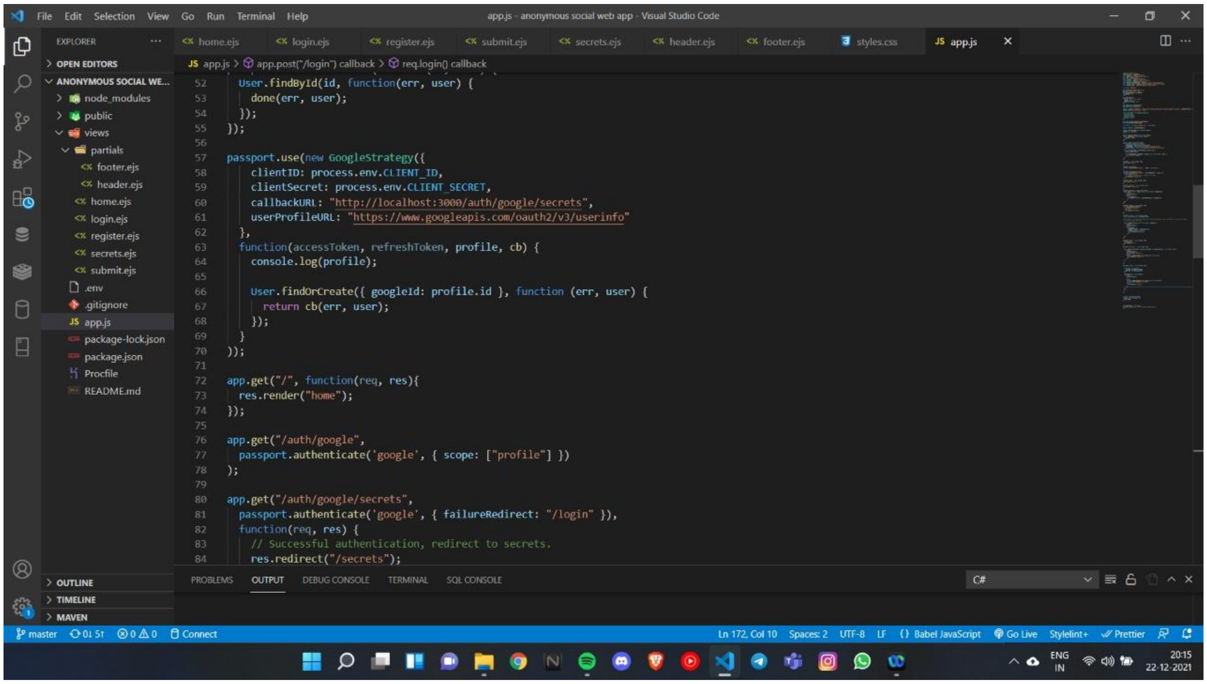


```javascript
//jshint esversion:6
require('dotenv').config();
const express = require("express");
const bodyParser = require("body-parser");
const ejs = require("ejs");
const mongoose = require("mongoose");
const session = require('express-session');
const passport = require("passport");
const passportLocalMongoose = require("passport-local-mongoose");
const GoogleStrategy = require('passport-google-oauth20').Strategy;
const findOrCreate = require('mongoose-findorcreate');

const app = express();

app.use(express.static("public"));
app.set('view engine', 'ejs');
app.use(bodyParser.urlencoded({
  extended: true
}));

app.use(session({
  secret: "Our little secret.",
  resave: false,
  saveUninitialized: false
}));

app.use(passport.initialize());
app.use(passport.session());

mongoose.connect("mongodb+srv://admin-pallav:adminpallav@cluster0.qlw9e.mongodb.net/userDB", {useNewUrlParser: true});
mongoose.set("useCreateIndex", true);

const userSchema = new mongoose.Schema ({
```

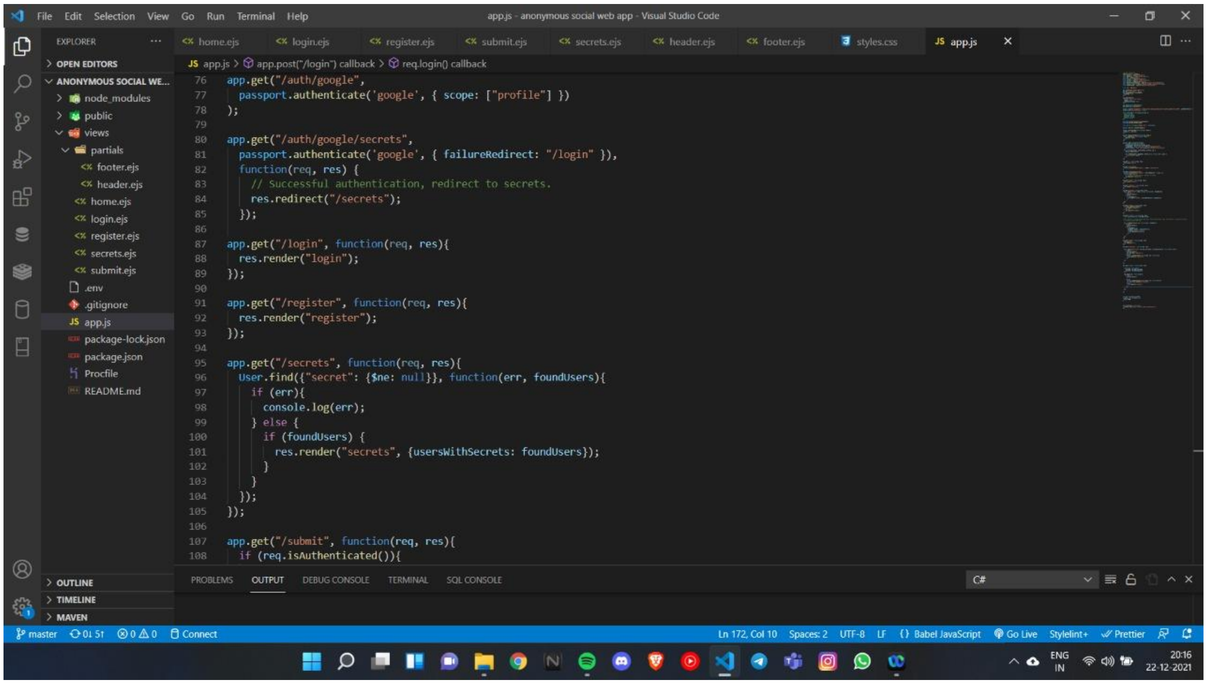
```
Downloading package 'Razor Language Server (Windows / x64)' (62313 KB)..............
```



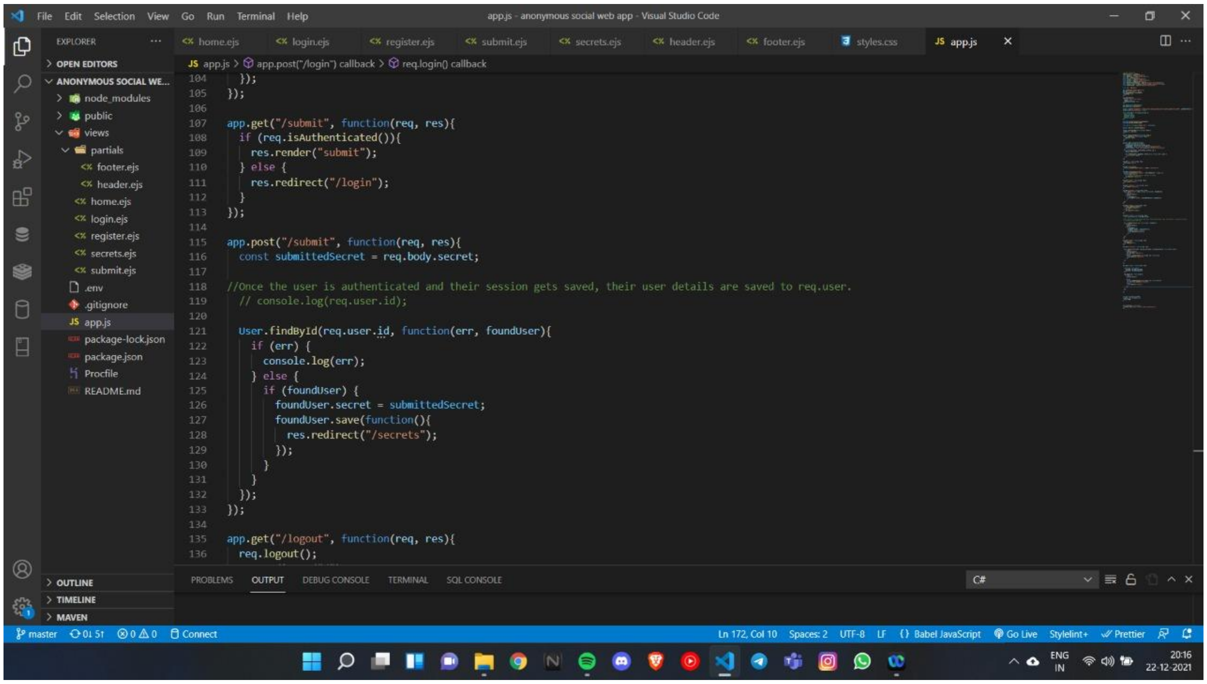
```
app.use(passport.initialize());
app.use(passport.session());

mongoose.connect("mongodb+srv://admin-pallav:adminpallav@cluster0.qlw9e.mongodb.net/userDB", {useNewUrlParser: true});
mongoose.set("useCreateIndex", true);

const userSchema = new mongoose.Schema ({
  email: String,
  password: String,
  googleId: String,
  secret: String
});

userSchema.plugin(passportLocalMongoose);
userSchema.plugin(findOrCreate);

const User = new mongoose.model("User", userSchema);

passport.use(User.createStrategy());

passport.serializeUser(function(user, done) {
  done(null, user.id);
});

passport.deserializeUser(function(id, done) {
  User.findById(id, function(err, user) {
    done(err, user);
  });
});

passport.use(new GoogleStrategy({
    clientID: process.env.CLIENT_ID,
```



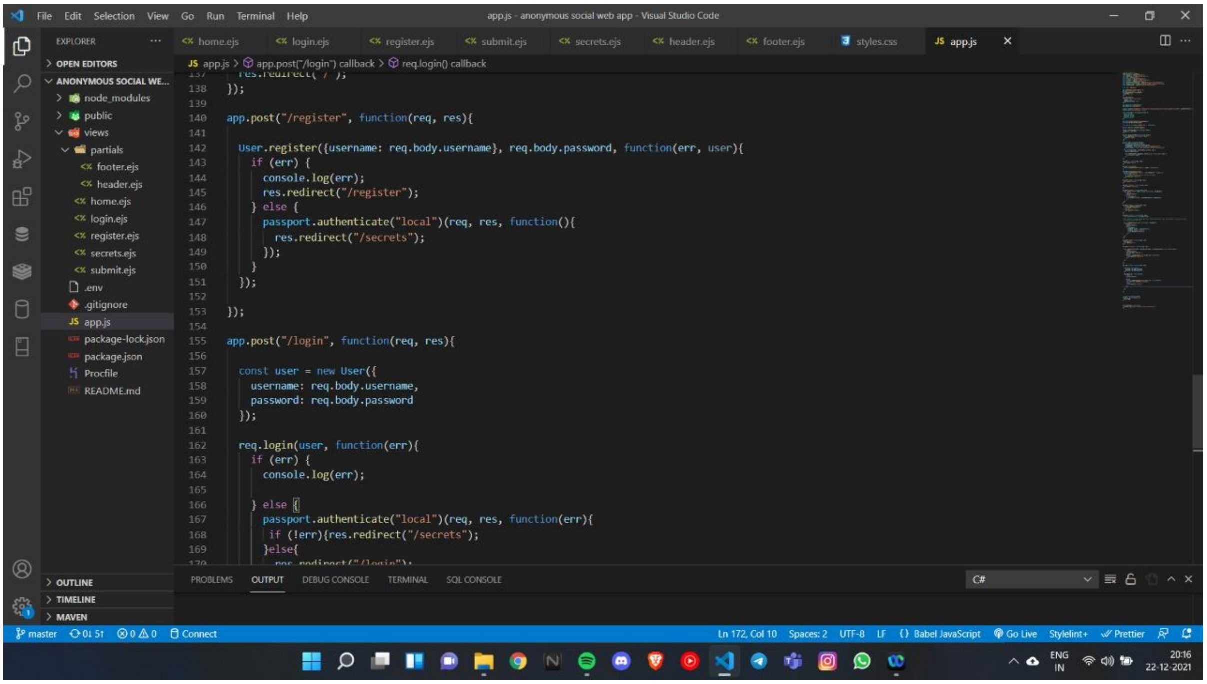
```
  User.findById(id, function(err, user) {
    done(err, user);
  });
});

passport.use(new GoogleStrategy({
    clientID: process.env.CLIENT_ID,
    clientSecret: process.env.CLIENT_SECRET,
    callbackURL: "http://localhost:3000/auth/google/secrets",
    userProfileURL: "https://www.googleapis.com/oauth2/v3/userinfo"
  },
  function(accessToken, refreshToken, profile, cb) {
    console.log(profile);

    User.findOrCreate({ googleId: profile.id }, function (err, user) {
      return cb(err, user);
    });
  }
));

app.get("/", function(req, res){
  res.render("home");
});

app.get("/auth/google",
  passport.authenticate('google', { scope: ["profile"] })
);

app.get("/auth/google/secrets",
  passport.authenticate('google', { failureRedirect: "/login" }),
  function(req, res) {
    // Successful authentication, redirect to secrets.
    res.redirect("/secrets");
```


app.js - anonymous social web app - Visual Studio Code
File Edit Selection View Go Run Terminal Help
home.ejs login.ejs register.ejs submit.ejs secrets.ejs header.ejs footer.ejs styles.css app.js
app.js > app.post("/login") callback > req.login() callback
EXPLORER
OPEN EDITORS
ANONYMOUS SOCIAL WE...
node_modules
public
views
partials
footer.ejs
header.ejs
home.ejs
login.ejs
register.ejs
secrets.ejs
submit.ejs
.env
.gitignore
app.js
package-lock.json
package.json
Procfile
README.md
OUTLINE
TIMELINE
MAVEN
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL SQL CONSOLE
C#
master 0↓ 5↑ 0 0 Connect
Ln 172, Col 10 Spaces: 2 UTF-8 LF {} Babel JavaScript Go Live Stylelint+ Prettier
ENG IN 20:16 22-12-2021

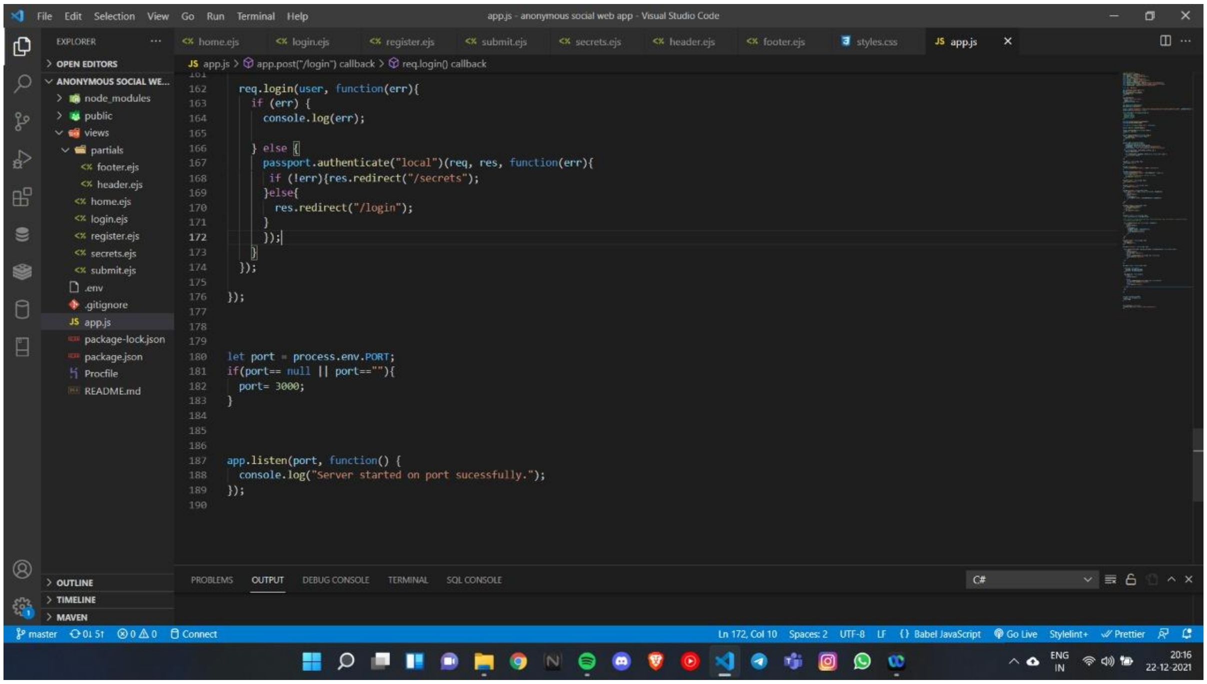
```
app.get("/auth/google",
  passport.authenticate('google', { scope: ["profile"] })
);

app.get("/auth/google/secrets",
  passport.authenticate('google', { failureRedirect: "/login" }),
  function(req, res) {
    // Successful authentication, redirect to secrets.
    res.redirect("/secrets");
  });

app.get("/login", function(req, res){
  res.render("login");
});

app.get("/register", function(req, res){
  res.render("register");
});

app.get("/secrets", function(req, res){
  User.find({"secret": {$ne: null}}, function(err, foundUsers){
    if (err){
      console.log(err);
    } else {
      if (foundUsers) {
        res.render("secrets", {usersWithSecrets: foundUsers});
      }
    }
  });
});

app.get("/submit", function(req, res){
  if (req.isAuthenticated()){
```


app.js - anonymous social web app - Visual Studio Code
File Edit Selection View Go Run Terminal Help
home.ejs login.ejs register.ejs submit.ejs secrets.ejs header.ejs footer.ejs styles.css app.js
app.js > app.post("/login") callback > req.login() callback
EXPLORER
OPEN EDITORS
ANONYMOUS SOCIAL WE...
node_modules
public
views
partials
footer.ejs
header.ejs
home.ejs
login.ejs
register.ejs
secrets.ejs
submit.ejs
.env
.gitignore
app.js
package-lock.json
package.json
Procfile
README.md
OUTLINE
TIMELINE
MAVEN
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL SQL CONSOLE
C#
master 0↓ 5↑ 0 0 Connect
Ln 172, Col 10 Spaces: 2 UTF-8 LF {} Babel JavaScript Go Live Stylelint+ Prettier
ENG IN 20:16 22-12-2021

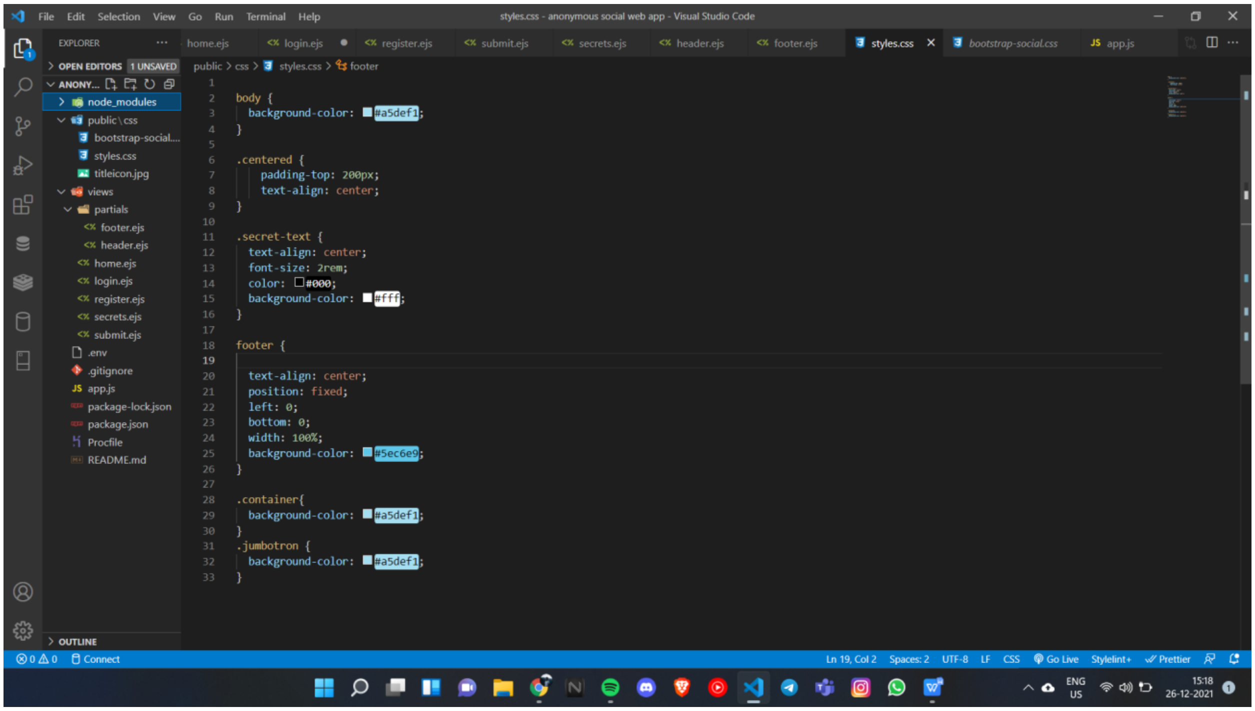
```
  });
});

app.get("/submit", function(req, res){
  if (req.isAuthenticated()){
    res.render("submit");
  } else {
    res.redirect("/login");
  }
});

app.post("/submit", function(req, res){
  const submittedSecret = req.body.secret;

//Once the user is authenticated and their session gets saved, their user details are saved to req.user.
  // console.log(req.user.id);

  User.findById(req.user.id, function(err, foundUser){
    if (err) {
      console.log(err);
    } else {
      if (foundUser) {
        foundUser.secret = submittedSecret;
        foundUser.save(function(){
          res.redirect("/secrets");
        });
      }
    }
  });
});

app.get("/logout", function(req, res){
  req.logout();
```



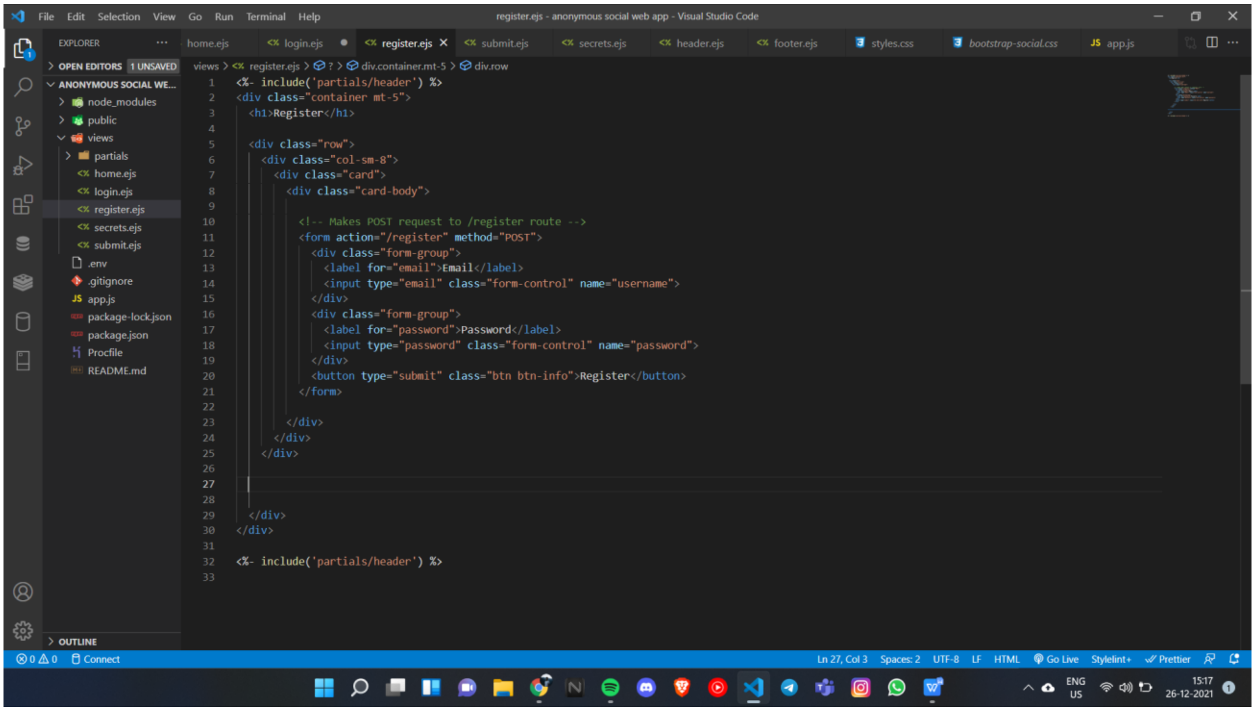
```
    res.redirect("/");
});

app.post("/register", function(req, res){

  User.register({username: req.body.username}, req.body.password, function(err, user){
    if (err) {
      console.log(err);
      res.redirect("/register");
    } else {
      passport.authenticate("local")(req, res, function(){
        res.redirect("/secrets");
      });
    }
  });

});

app.post("/login", function(req, res){

  const user = new User({
    username: req.body.username,
    password: req.body.password
  });

  req.login(user, function(err){
    if (err) {
      console.log(err);

    } else {
      passport.authenticate("local")(req, res, function(err){
       if (!err){res.redirect("/secrets");
      }else{
        res.redirect("/login");
```



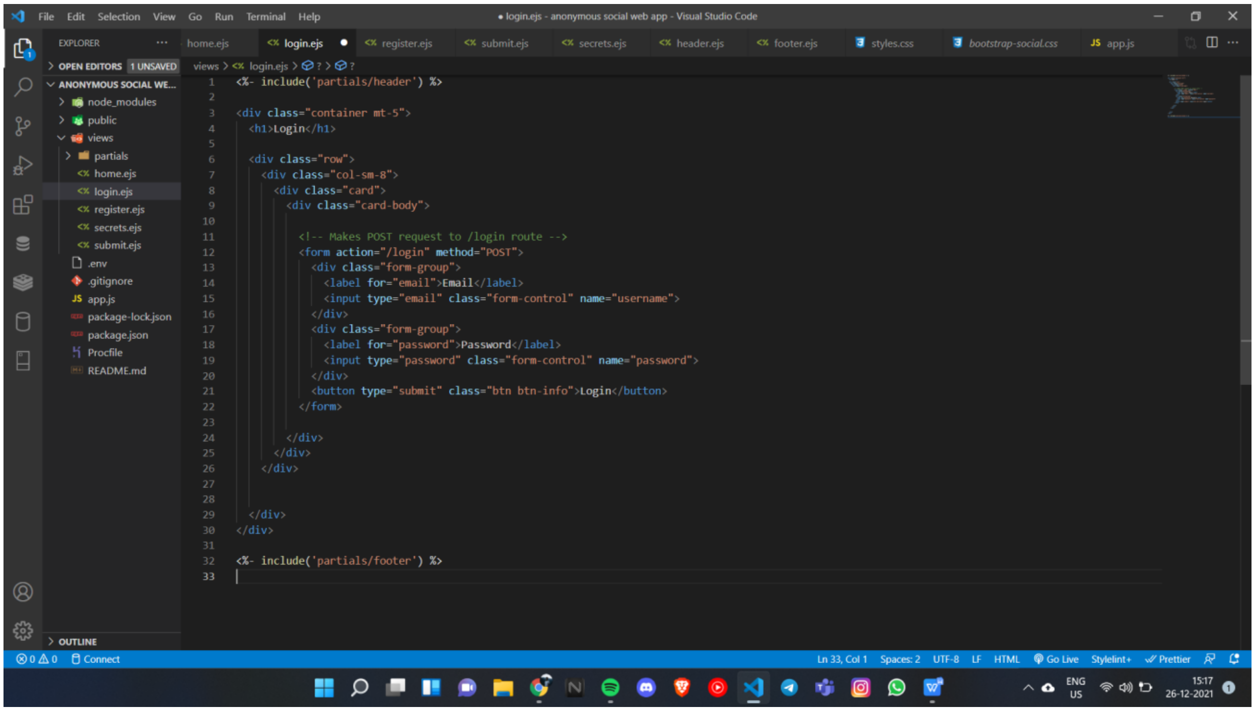
```
  req.login(user, function(err){
    if (err) {
      console.log(err);

    } else {
      passport.authenticate("local")(req, res, function(err){
       if (!err){res.redirect("/secrets");
      }else{
        res.redirect("/login");
      }
      });
    }
  });

});



let port = process.env.PORT;
if(port== null || port==""){
  port= 3000;
}



app.listen(port, function() {
  console.log("Server started on port sucessfully.");
});
```



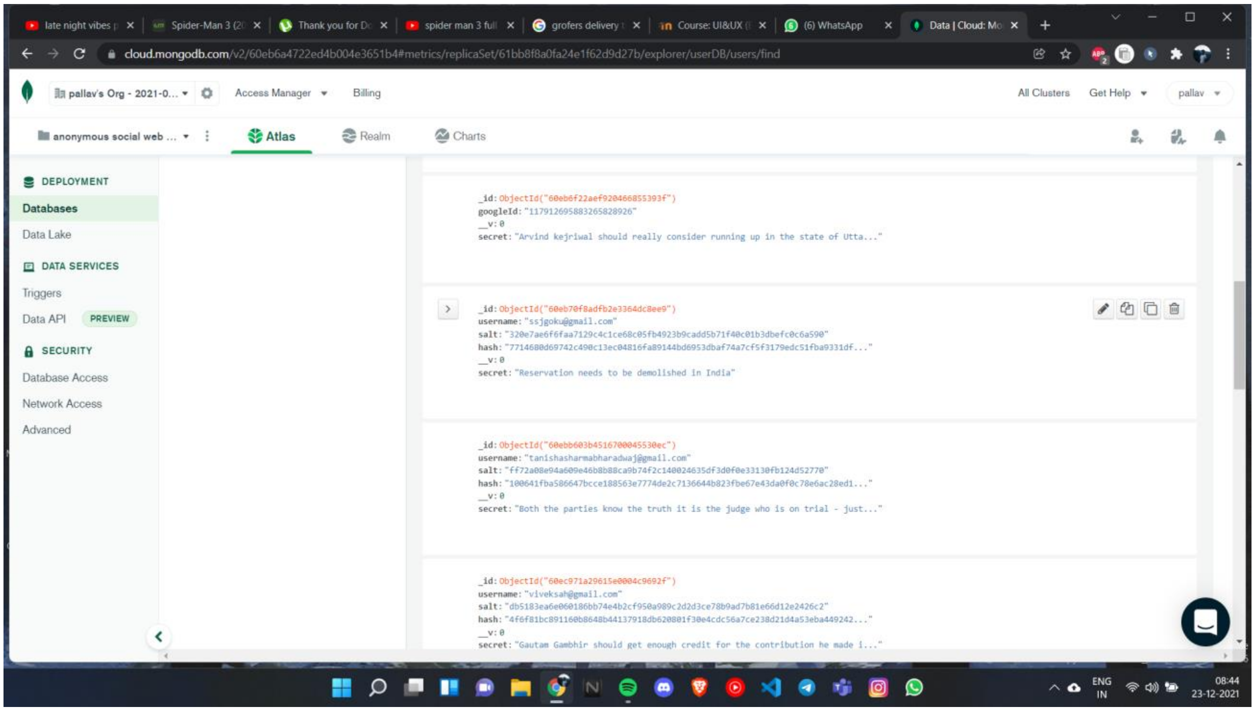
```css

body {
  background-color: #a5def1;
}

.centered {
    padding-top: 200px;
    text-align: center;
}

.secret-text {
  text-align: center;
  font-size: 2rem;
  color: #000;
  background-color: #fff;
}

footer {

  text-align: center;
  position: fixed;
  left: 0;
  bottom: 0;
  width: 100%;
  background-color: #5ec6e9;
}

.container{
  background-color: #a5def1;
}
.jumbotron {
  background-color: #a5def1;
}
```



```html
<%- include('partials/header') %>
<div class="container mt-5">
  <h1>Register</h1>

  <div class="row">
    <div class="col-sm-8">
      <div class="card">
        <div class="card-body">

          <!-- Makes POST request to /register route -->
          <form action="/register" method="POST">
            <div class="form-group">
              <label for="email">Email</label>
              <input type="email" class="form-control" name="username">
            </div>
            <div class="form-group">
              <label for="password">Password</label>
              <input type="password" class="form-control" name="password">
            </div>
            <button type="submit" class="btn btn-info">Register</button>
          </form>

        </div>
      </div>
    </div>



  </div>
</div>

<%- include('partials/header') %>

```

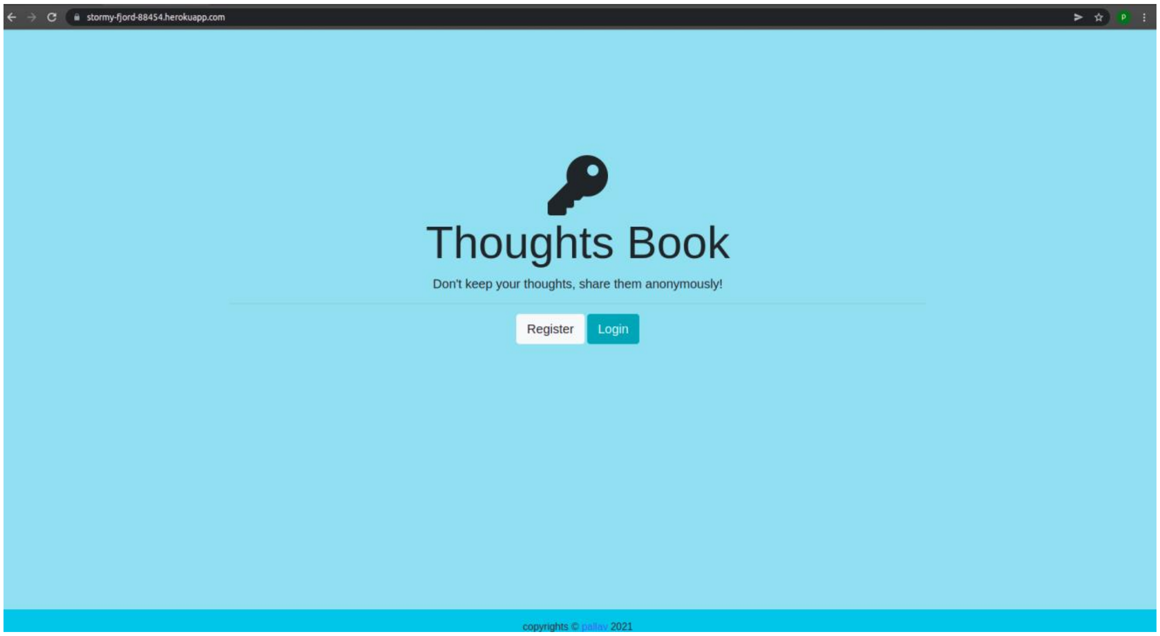


```
<%- include('partials/header') %>

<div class="container mt-5">
  <h1>Login</h1>

  <div class="row">
    <div class="col-sm-8">
      <div class="card">
        <div class="card-body">

          <!-- Makes POST request to /login route -->
          <form action="/login" method="POST">
            <div class="form-group">
              <label for="email">Email</label>
              <input type="email" class="form-control" name="username">
            </div>
            <div class="form-group">
              <label for="password">Password</label>
              <input type="password" class="form-control" name="password">
            </div>
            <button type="submit" class="btn btn-info">Login</button>
          </form>

        </div>
      </div>
    </div>


  </div>
</div>

<%- include('partials/footer') %>

```



```
_id: ObjectId("60eb6f22aef920466855393f")
googleId: "117912695883265828926"
__v: 0
secret: "Arvind kejriwal should really consider running up in the state of Utta..."

_id: ObjectId("60eb70f8adfb2e3364dc8ee9")
username: "ssjgoku@gmail.com"
salt: "320e7ae6f6faa7129c4c1ce68c05fb4923b9cadd5b71f40c01b3dbefc0c6a590"
hash: "7714680d69742c490c13ec04816fa89144bd6953dbaf74a7cf5f3179edc51fba9331df..."
__v: 0
secret: "Reservation needs to be demolished in India"

_id: ObjectId("60ebb603b4516700045530ec")
username: "tanishasharmabharadwaj@gmail.com"
salt: "ff72a08e94a609e46b8b88ca9b74f2c140024635df3d0f0e33130fb124d52770"
hash: "100641fba586647bcce188563e7774de2c7136644b823fbe67e43da0f0c78e6ac28ed1..."
__v: 0
secret: "Both the parties know the truth it is the judge who is on trial - just..."

_id: ObjectId("60ec971a29615e0004c9692f")
username: "viveksah@gmail.com"
salt: "db5183ea6e060186bb74e4b2cf950a989c2d2d3ce78b9ad7b81e66d12e2426c2"
hash: "4f6f81bc891160b8648b44137918db620801f30e4cdc56a7ce238d21d4a53eba449242..."
__v: 0
secret: "Gautam Gambhir should get enough credit for the contribution he made i..."
```

10.2 Product Output

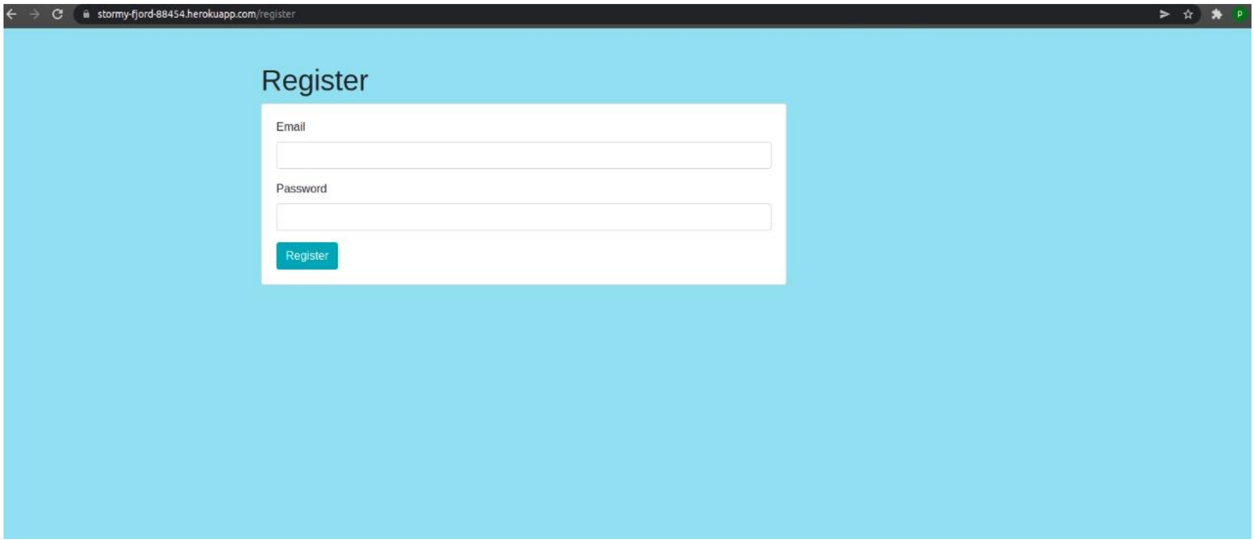


stormy-fjord-88454.herokuapp.com

Thoughts Book

Don't keep your thoughts, share them anonymously!

copyrights © pallav 2021

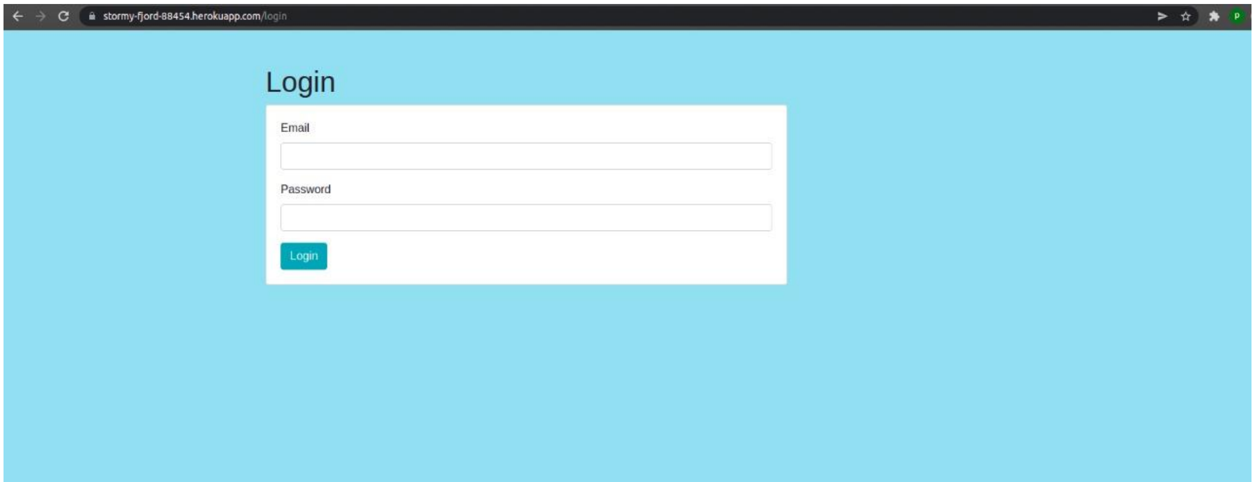


stormy-fjord-88454.herokuapp.com/register

Register

Email

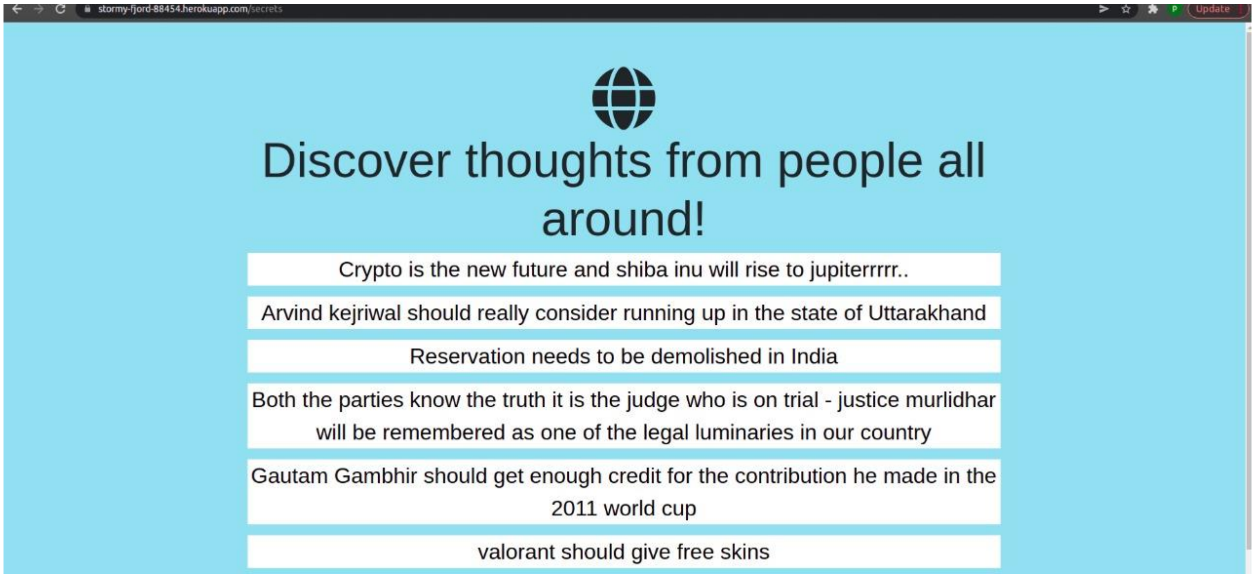
Password



Login

Email

Password



Discover thoughts from people all around!

Crypto is the new future and shiba inu will rise to jupiterrrrr..

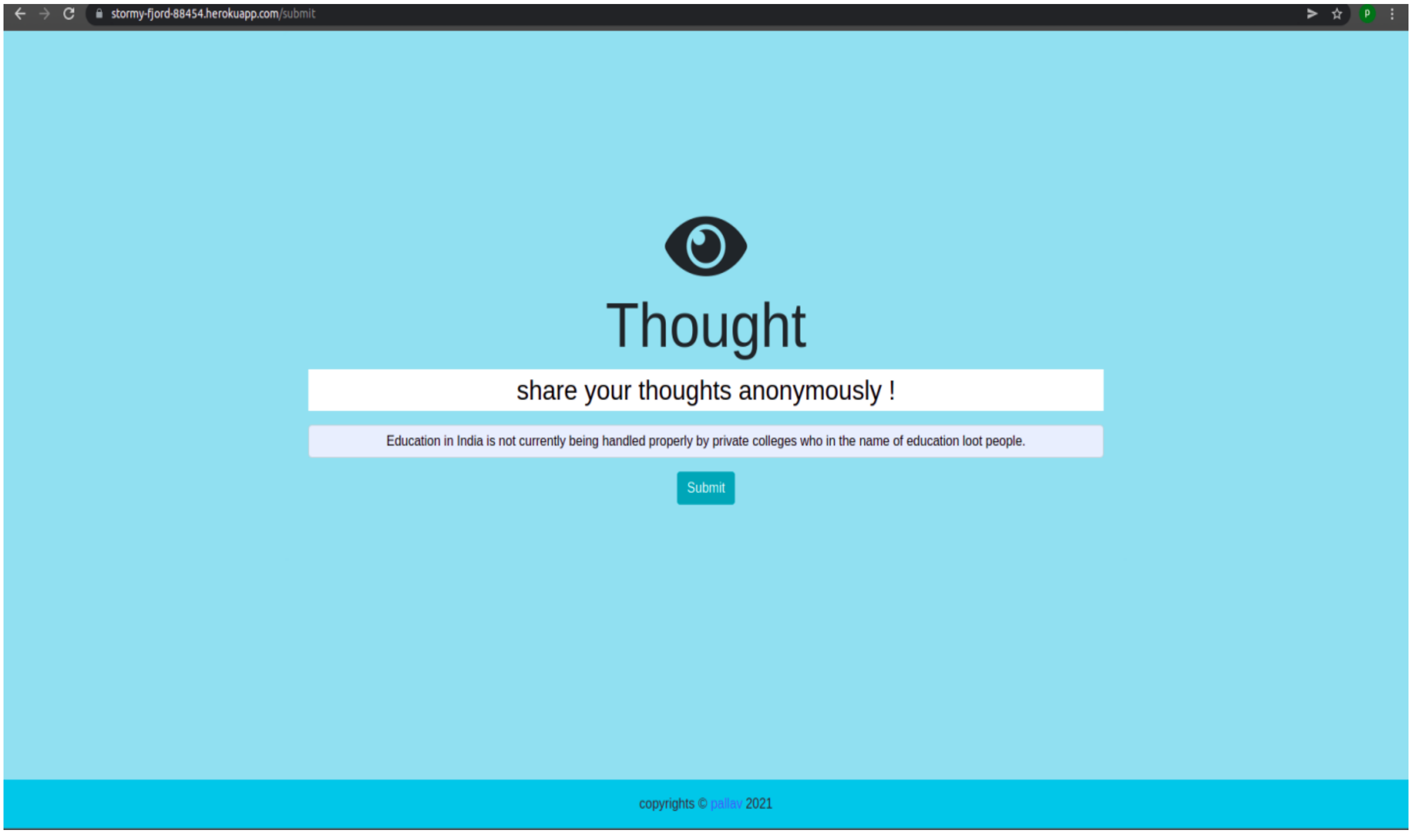
Arvind kejriwal should really consider running up in the state of Uttarakhand

Reservation needs to be demolished in India

Both the parties know the truth it is the judge who is on trial - justice murlidhar will be remembered as one of the legal luminaries in our country

Gautam Gambhir should get enough credit for the contribution he made in the 2011 world cup

valorant should give free skins



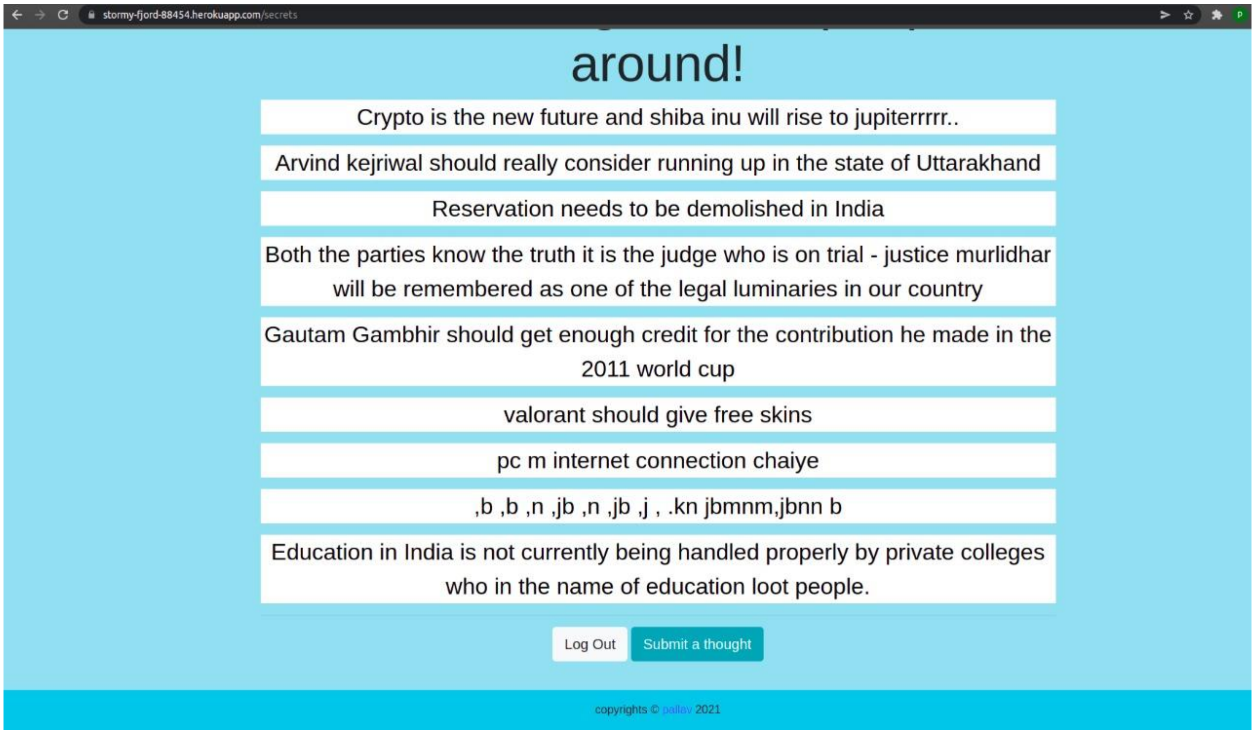
stormy-fjord-88454.herokuapp.com/submit

Thought

share your thoughts anonymously !

Education in India is not currently being handled properly by private colleges who in the name of education loot people.

copyrights © pallav 2021



around!

Crypto is the new future and shiba inu will rise to jupiterrrrr..

Arvind kejriwal should really consider running up in the state of Uttarakhand

Reservation needs to be demolished in India

Both the parties know the truth it is the judge who is on trial - justice murlidhar will be remembered as one of the legal luminaries in our country

Gautam Gambhir should get enough credit for the contribution he made in the 2011 world cup

valorant should give free skins

pc m internet connection chaiye

,b ,b ,n ,jb ,n ,jb ,j , .kn jbmnm,jbnn b

Education in India is not currently being handled properly by private colleges who in the name of education loot people.

copyrights © pallav 2021

Reference

[1] Boyd D. Taken Out of Context: American Teen Sociality in Networked Publics Berkeley, CA: University of California; (2008).

[2] Junco, R., Heibergert, G. & Loken, E. (2010). The Effect of Twitter on college students Engagement and Grades, Journal of Computer Assisted Learning

[3] Krishna S, Boren SA, Balas EA. Healthcare via cell phones: a systematic review. Telemed E Health. (2009).

Lenhart A. Cyberbullying. Washington, DC: Pew Research Center; (2007).

https://stormy-fjord-88454.herokuapp.com
Mentor Mr. Shreenarayanam N M