A Project Report on

An Integrated Platform for Knowledge Sharing

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

Bachelor of Engineering

in

Information Technology

by

Shankarlal Sharma (15104018)
Gaurav Babar (15104033)
Dharmraj Yadav (15104021)
Satyajeet Yadav (15104025)

Under the Guidance of

Guide: Prof. Apeksha Mohite
Co-Guide: Prof. Neha Deshmukh

Department of Information Technology
A.P. Shah Institute of Technology
G.B.Road,Kasarvadavli, Thane(W), Mumbai-400615
UNIVERSITY OF MUMBAI

Academic Year 2018-2019

Approval Sheet

This Project Report entitled ***"An Integrated Platform for knowledge Sharing"*** Submitted by ***"Shankarlal Sharma" (15104018)***, ***"Gaurav Babar" (15104033)***, **"Dharmraj Yadav" (15104021), "Satyajeet Yadav" (15104025)** is approved for the partial fulfillment of the requirenment for the award of the degree of ***Bachelor of Engineering*** in ***Information Technology*** from ***University of Mumbai***.

Prof. Neha Deshmukh
Co-Guide

Prof. Apeksha Mohite
Guide

Prof. Kiran Deshpande
Head Department of Information Technology

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

CERTIFICATE

This is to certify that the project entitled ***"An Integrated Platform for Knowledge Sharing"*** submitted by ***"Shankarlal Sharma" (15104018),"Gaurav Babar" (15104033),"Dharmraj Yadav" (15104021),"Satyajeet Yadav" (15104025)*** for the partial fulfillment of the requirement for award of a degree ***Bachelor of Engineering*** in ***Information Technology***,to the University of Mumbai,is a bonafide work carried out during academic year 2018-2019.

Prof. Neha Deshmukh Co-Guide	Prof. Apeksha Mohite Guide
Prof. Kiran Deshpande Head Department of Information Technology	Dr. Uttam D.Kolekar Principal

External Examiner(s)

1.

2.

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

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, We have adequately cited and referenced the original sources. We also declare that We have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Signatures:

Shankarlal Sharma
15104018

Gaurav Babar
15104033

Dharmraj Yadav
15104021

Satyajeet Yadav
15104025

Date:

Acknowledgement

We have great pleasure in presenting the report on **An Integrated Platform for Knowledge Sharing.** We take this opportunity to express our sincere thanks towards our guide **Prof. Apeksha Mohite** & Co-Guide **Prof. Neha Deshmukh** Department of IT, APSIT thane for providing the technical guidelines and suggestions regarding line of work. We would like to express our gratitude towards his constant encouragement, support and guidance through the development of project.

We thank **Prof. Kiran B. Deshpande** Head of Department,IT, APSIT for his encouragement during progress meeting and providing guidelines to write this report.

We thank **Prof. Vishal S. Badgujar** BE project co-ordinator, Department of IT, APSIT for being encouraging throughout the course and for guidance.

We also thank the entire staff of APSIT for their invaluable help rendered during the course of this work. We wish to express our deep gratitude towards all our colleagues of APSIT for their encouragement.

Shankarlal Sharma
15104018

Gaurav Babar
15104033

Dharmraj Yadav
15104021

Satyajeet Yadav
15104025

Abstract

The issue of what content to be written in exams is generally faced by most of the students. It may happen that a student understands the concept but isnt able to represent it in a way that examiner can understand. Therefore, we intend to build a community source platform that will assist students in preparing for their exams and also for learning a proper skillset. Academics are an important part of student life and getting high scores in exams is equally important as it is considered the primary criteria for hiring of a candidate by majority of organizations. In order to score high, studying from quality study material is a must. But it is observed that not all students get access to such quality study material while preparing for their exams and finally they tend to go for local publications available in market and rote learn them to anyhow clear the exams without taking care about whether the content they are referring is trustworthy. With the help of this project, we aim to provide a solution to this problem by building a web-based platform where the students sharing the same academic domain can come together, share the study material, collaborate with each other, help other students to expand their knowledge and at the same time gain the benefits from content already provided by other users which they were searching for. This platform will have a Rich User Interface and easy navigation where the study material will be categorized into branch, subject, module and finally into topics. Such a platform can help a large number of students by guiding them to prepare for their exams with confidence

Contents

List of Figures

List of Abbreviations

JSON:	JavaScript Object Notation
JWT:	JSON Web Tokens
JSX:	JavaScript XML
DOM:	Document Object Model
PHP:	Hypertext Preprocessor
API:	Application programming interface
HTTP:	HyperText Transfer Protocol
UX:	User Experience
HMAC:	Hashed Message Authentication Code
ECDSA:	Elliptic Curve Digital Signature Algorithm

Chapter 1

Introduction

Examinations play an important role in students lives and so do the scores achieved in those exams. In order to score high grades, students require quality study material by referring which they can be confident of performing well in their exams. The problem that arises is not all students have access to such quality study material and so they decide to refer local publications which are easily available in market and rote learn them to somehow clear this exam without thinking about whether the material they are reading is of relevance and is trustworthy. During our research, we searched for resources that are already available to solve this problem, and found that there are several websites that have took steps to reduce this but most of them are limited to providing solutions to questions that are previously asked in exams but what about some important topics that are not appeared in exams but is important and also there are no resources available that provides material according to the syllabus prescribed by university. With the help of our project, we intend to provide a solution to this problem in the form of a web-based platform where the users sharing the same domain can come together and collaborate on study material. This platform will have the content categorized into branch, subject, modules and topics in accordance to the syllabus prescribed by the university, which will make it easier for the user to navigate through the platform and quickly access the topic they were searching for. The platform will have a Rich User Interface with other interactive features to help students prepare better for their exams.

1.1 Problem Definition

The problem that arises is that there is no single book, portal, website where students of a university can get relevant content for learning the courses of that particular university. This causes the students to read from various unreliable sources and understanding of the subject is unclear at the end. This leads to less understanding and low score in examinations. Also, for professors who curate proper resources for their content there is no platform available currently to share their content outside their colleges, because of this only few privileged and lucky students get access to the valuable resources.

1.2 Objectives

- Our main objective is to create a platform for students and teachers where they can upload and view self made or collaborative content shared by others and prepare themselves for their academics and enhance their knowledge.
- To maintain a collection of educational content filtered according to topics, modules and subjects.
- To reduce the gap between the students and new modern technology.

1.3 Operating Environment

1.3.1 Hardware Requirements:

Client-end: This software needs a processor having at least 2 Cores, with a 512MB memory and a 512Kbps internet connection.

Server-end: We need a Hex-core processor, or better, with a Gigabit F/O and Ethernet connection, with 12GB or more memory.

1.3.2 Software requirements:

Client End: OS: Windows XP or later. Linux 14.10 or later. Smartphones: Android 4.0 or later. Web Browsers supported: Google Chrome (prefer updated version), Mozilla Firefox (prefer updated), Internet Explorer 9 or later.

Server-End: OS: Windows Server 2012 R2 or better, or Linux-kernel based OS.

Chapter 2

Literature Review

In this section a number of research manuscripts were reviewed concentrating on web-based learning, e-learning and internet based learning including summaries about the technology used to create online resources and the technology used to present and access online information.

[1] Blanka Frydrychova Klimova, Petra Poulova, Michal Slama, eLeaming study materials and students preferences , 2014 Information Technology Based Higher Education and Training(ITHET) , 2014.
A survey of 2449 respondents 1867 students was undertaken on online courses, As this research showed, a majority of respondents welcomed a possibility of having their study materials online. Students are not used to exploring the resources because they are not made available in an online medium.

[2] Ivana Simonova, Petra Poulava, Study Materials in Online Courses, Analysis Reflecting Individual Learning Styles, 2014 IEEE Global Engineering Education Conference(EDUCON), 2014.
Students were provided various types of study materials and their satisfaction with the process was monitored, The participants in the online course expressed their positive approach and satisfaction with the course of study. Despite the contribution of the learning style theory to the online learning process was not proved within this project

[3] Dr.P Pabitha, M.Mohana, S.Suganthi, B.Sivanandhini, Automatic Question Generation System, 2014 International Conference on Recent Trends in Information Technology.
The conceptual architecture is proposed which focuses on adaptive e-Learning application As this research showed, a majority of respondents welcomed a possibility of having their study materials online.

[4] Sucheta Kolekar, Radhika M. Pai, Manohara Pai M.M., Adaptive User Interface for Elearning Applications based on Learning Styles using Web Logs Analysis: A Hybrid Cloud Architecture,TENCON 2015 - 2015 IEEE Region 10 Conference, 2015.
Misamis University Online Learning Environment used to rate the UX on web services It is expected that with this approach, future E-Learning applications will concentrate on learners learning styles and adaptive interface for personalization. The novelty aspect earned very low. This might due to the students or users do not consider it as a contributing factor for

them to have a good user experience.

[5] Rolysent K. Paredes, Alexander A. Hernandez, Measuring the Quality of User Experience on Web Services: A Case of University in the Philippines, 2017 IEEE 9th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM), 2017.
Case study of Massive Online Open Course (MOOC) Students were able to discover more requirements regarding the app features and mobile device requirements. Thus, students are not used to exploiting the resources because they are not made available in an online medium.

[6] Mahesh S. Patil, Meenaxi M Raikar, Padmashree Desai, Vijayalakshmi M, Shivalingappa Battur, Parikshit H, G.H Joshi,Leveraging Student Project through MOOC on UX:Case Study 2016 IEEE 4th International Conference on MOOCs, Innovation and Technology in Education (MITE).
Question generation using Supervised learning proposed system has so far implemented the stemming part using the porter algorithm. Extracting all nouns hinder the system and generate unnecessary questions.

[7] Sriramya, P and Karthika, R.A.. (2015). Providing password security by salted password hashing using Bcrypt algorithm. ARPN Journal of Engineering and Applied Sciences. 10. 5551-5556.
This paper discusses different methods to thwart these attacks: (1) the use of a strong password to reduce the probability of it existing in a dictionary, (2) using salts, (3) key stretching and iteration hashing to make the bcrypt computation slower, (4) chaining method, where the output of one iteration is used in the input of the next iteration and the use of a different initialization vector for each password. This paper also addresses the Salted Password Hashing Technique using bcrypt algorithm for providing the users privacy when shopping online.

[8] Kumar N., Chaudhary P. (2018) Password Security Using Bcrypt with AES Encryption Algorithm. In: Satapathy S., Bhateja V., Das S. (eds) Smart Computing and Informatics. Smart Innovation, Systems and Technologies, vol 77. Springer, Singapore
In this paper, authers have scheduled a technique utilizing Bcrypt hashing technique with AES encryption for securing an online account and reducing cyber criminal activity.

[9] F. Wiemer and R. Zimmermann, "High-speed implementation of bcrypt password search using special-purpose hardware," 2014 International Conference on ReConFigurable Computing and FPGAs (ReConFig14), Cancun, 2014, pp. 1-6.
In this paper, auther have presented a novel, flexible, high-speed implementation of a bcrypt password search system on a low-power Xilinx Zynq 7020 FPGA. The design consisted of 40 parallel bcrypt cores running at 100 MHz. This implementation outperformed all currently available implementations and improved password attacks on the same platform by at least 42%, computing 6,511 passwords per second for a cost parameter of 5.

[10] C. Gyrdi, R. Gyrdi, G. Pecherle and A. Olah, "A comparative study: MongoDB vs.

MySQL," 2015 13th International Conference on Engineering of Modern Electric Systems (EMES), Oradea, 2015, pp. 1-6.
In this paper, author has presented a comparative study of non-relational databases and relational databases. We mainly focus our presentation on one implementation of the NoSQL database technology, namely MongoDB, and make a comparison with another implementation of relational databases, namely MySQL, and thus justifying why MongoDB is more efficient than MySQL. Author have also presented the advantages of using a non-relational database compared to a relational database, integrated in a forum in the field of personal and professional development.
The findings of this paper were as follows: MongoDB provided lower execution times than MySQL in all four basic operations, which is essential when an application should provide support to thousands of users simultaneously. We can choose MongoDB instead of MySQL if the application is data intensive and stores many data and queries lots of data.

[11] M. H. Mohamed, M. H. Khafagy and M. H. Ibrahim, "Recommender Systems Challenges and Solutions Survey," 2019 International Conference on Innovative Trends in Computer Engineering (ITCE), Aswan, Egypt, 2019, pp. 149-155.
In this paper, Author discusses four techniques in recommender systems and list advantages and disadvantages for everyone. Also this paper discusses recommender system challenges: like, cold start, scalability, privacy, gray sheep, Shilling attack, and novelty, Sparsity, Diversity, over specialization problem.It also discusses some research topics solutions used to overcome challenges and their advantages.

[12] Martins P., Abbasi M., S F. (2019) A Study over NoSQL Performance. In: Rocha ., Adeli H., Reis L., Costanzo S. (eds) New Knowledge in Information Systems and Technologies. WorldCIST'19 2019. Advances in Intelligent Systems and Computing, vol 930. Springer, Cham
This paper provides a comparative study of different NoSQL databases performance. First, the most popular NoSQL databases, like, Cassandra and HBase, which are column-family based. Second, MongoDB and OrientDB, which are document-store based. Finally, Redis and Memcached, are a key-value in- memory database.
In order to test the performance of the database, YCSB benchmark was used with several workloads. Results show that the in-memory database, Redis and Memcached, are extremely fast, loading, updating and reading data. Nevertheless, a big limitation must be accounted, and the used memory is volatile, more expensive and relatively limited. Voldemort combines key-valued approach with in-memory and file-system for data persistence, however, it comes with a performance cost. With the worst performance are the document store databases, OrientDB. The use of virtual machine environment, regarding computing resources, is more demanding, therefore implies larger overheads. HBase and Cassandra, are optimized for sequential reading and writing, reducing the accesses to disk, as well as, number of operations by temporally storing records in memory.

Chapter 3

Existing System Architecture

The problem that arises is that there is no single book, portal, website where students of a university can get relevant content for learning the courses of that particular university. This causes the students to read from various unreliable sources and understanding of the subject is unclear at the end. Students just read the solutions of those questions which have high possibility of appearing in exams. The conceptual knowledge is not gained and students tend to forget what they tried to recite. This leads to less understanding and low score in examinations. Also, for professors who curate proper resources for their content there is no platform available currently to share their content outside their colleges, because of this only few privileged and lucky students get access to the valuable resources.

The competitors are like StupidSid and Ques10. StupidSid is providing paid service for answers solutions and Ques10 only focuses on Ques-Ans which have appeared in University examination, it doesnt focus on providing content for all the topics in the syllabus.

Chapter 4

Proposed System

4.1 Proposed System Architecture

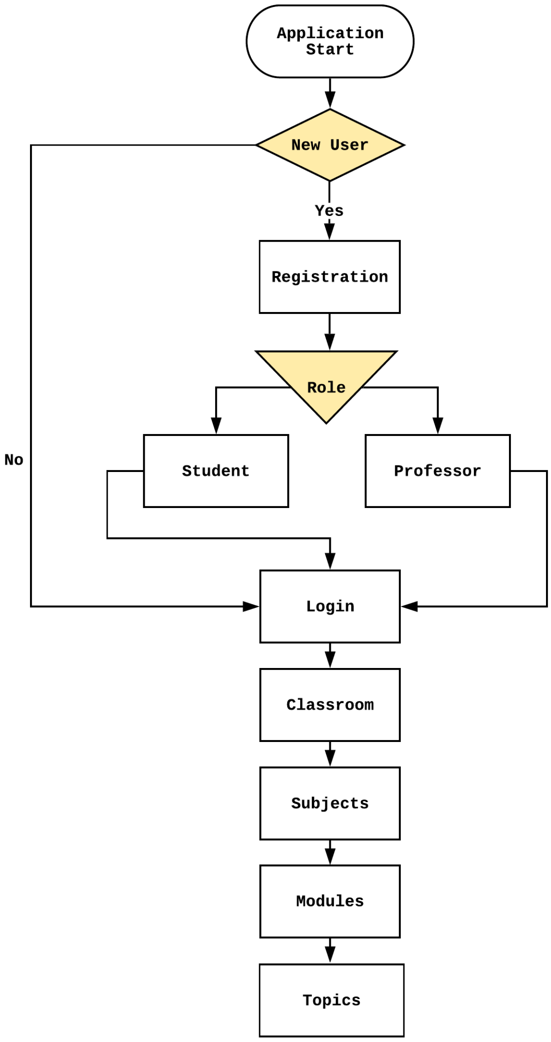


Figure 4.1: System Architecture

After identifying the drawbacks of the current system, we decided to tackle the issues by creating a system which is crowdsourced and encourages the users to collaborate together and curate their own content. When the user uploads the content about a specific topic, it gets stored in the repository. Like this, content from various topics will integrate to form a module. When we group up all the modules, we can have the notes of whole subject! This will help students to understand the concept rather than just reading the solution to previously appeared questions. Another advantage of creating this system is that, nowadays a greater population of the younger generation depends upon the social media to gain information. Creating this platform will help in building an e-learning social network which gets its information through collaborative learning. Thats why our motto is For the student, Of the student, By the student.

In this system if the user is new to the web application then he/she needs to register,enter their details and login. If the user is already registered then he/she will directly login to the web application. After login if he/she wants to read a module or a particular topic,the user can navigate through the platform and quickly access the topic they were searching for. Users can also create,upload and share the study materials for others.

4.1.1 Purpose of this model

What:
The motive behind this is to build a network that will help teachers and students of a university to collaborate together to share their knowledge and learn with each other.
Why:
The problem that comes is there is no single book, portal, website where students of a university can get relevant content for learning of the courses in that particular university. This causes the students to read from various unreliable sources and understanding of the subject is unclear at the end and that leads to less understanding and low score in examinations.
How:
The process is simple with the following four steps to make it easier to use for its users.

- **Step 1:**
  Register: The user will first have to register to the site to create his/her own identity to browse through the entire site

- **Step 2:**
  Learn: After the sign in, the user will be able to start the process of acquiring new, or modifying existing, knowledge, behaviors, skills, values

- **Step 3:**
  Teach: If the user has better skills, he can showenthusiasm towards the course materials and students can create a positive learning experience.

- **Step 4:**
  Grow: Keep exploring the content to enhance your skill set and boost your knowledge through flexible and collaborative learning

4.2 React (JavaScript library)

React (also known as **React.js** or **ReactJS**) is a JavaScript library for building user interfaces. It is maintained by Facebook and a community of individual developers and companies. React can be used as a base in the development of single-page or mobile applications. Complex React applications usually require the use of additional libraries for state management, routing, and interaction with an API

4.2.1 Features of ReactJs

- **One-way data binding with props**
  Properties (commonly, props) are passed to a component from the parent component. Components receive props as a single set of immutable values (a JavaScript object).

- **Stateful components**
  States hold values throughout the component and can be passed to child components through props:

- **Virtual DOM**
  Another notable feature is the use of a ”virtual Document Object Model”, or ”virtual DOM”. React creates an in-memory data-structure cache, computes the resulting differences, and then updates the browser’s displayed DOM efficiently.
  This allows the programmer to write code as if the entire page is rendered on each change, while the React libraries only render subcomponents that actually change.

- **JSX**
  JSX (JavaScript XML) is an extension to the JavaScript language syntax. Similar in appearance to HTML, JSX provides a way to structure component rendering using syntax familiar to many developers. React components are typically written using JSX, although they do not have to be (components may also be written in pure JavaScript). JSX is similar to another extension syntax created by Facebook for PHP, XHP.

4.2.2 Why do internet giants use ReactJS?

- **Facebook**
  Although partially, Facebook is making use of ReactJS. Their webpage is built with React, as the script that is blended in the application code. The mobile app is also build with a version of React called React Native which is similar, although responsible for displaying the iOS and Android native components instead of the DOM elements. Interestingly, Facebook was the place where the ReactJS library was initially created, which is why it is obvious for the app to use it. Facebook currently opened a beta of its completely rewritten ReactJS, called React Fiber.

- **Instagram**
  The use of ReactJS within Instagram is huge. A proof for that are the numerous features including the geolocations, Google Maps APIs, search engine accuracy as well as tags that pop out without hashtags. IT is all there in the API of the app and is really impressive.

Instagram is completely based on the ReactJS library and has let fans fully adapt to its amazing features.

- **Netflix**
  The React version works with Netflix too specifically on their platform called Gibbon which is used for low-performance TV devices instead of the DOM used in web browsers. Netflix has even published an official blog post explaining how the ReactJS library helps their startup speed, runtime performance, modularity and various other advantages.
  As the UI engineers at Netflix state in the blog post:
  Our decision to adopt React was influenced by a number of factors, most notably: 1) startup speed, 2) runtime performance, and 3) modularity.

4.3 MongoDB

4.3.1 What is MongoDB

- MongoDB stores data in flexible, JSON-like documents, meaning fields can vary from document to document and data structure can be changed over time
- The document model maps to the objects in your application code, making data easy to work with
- Ad hoc queries, indexing, and real time aggregation provide powerful ways to access and analyze your data
- Ad hoc queries, indexing, and real time aggregation provide powerful ways to access and analyze your data
- MongoDB is free to use.

4.3.2 Why Other Databases Fall Short

Most databases make you chose between a flexible data model, low latency at scale, and powerful access. But increasingly you need all three at the same time.

- **Rigid Schemas.** You should be able to analyze unstructured, semi-structured, and polymorphic data. And it should be easy to add new data. But this data doesnt belong in relational rows and columns. Plus, relational schemas are hard to change incrementally, especially without impacting performance or taking the database offline.
- **Scaling Problems.** Relational databases were designed for single-server configurations, not for horizontal scale-out. They were meant to serve 100s of ops per second, not 100,000s of ops per second. Even with a lot of engineering hours, custom sharding layers, and caches, scaling an RDBMS is hard at best and impossible at worst.
- **Takes Too Long.** Analyzing data in real time requires a break from the familiar ETL and data warehouse approach. You dont have time for lengthy load schedules, or to build new query models. You need to run aggregation queries against variably structured data. And you should be able to do so in place, in real time.

4.3.3 How MongoDB Makes it Easy

Organizations are using MongoDB for analytics because it lets them store any kind of data, analyze it in real time, and change the schema as they go.

- **New Data.** MongoDBs document model enables you to store and process data of any structure: events, time series data, geospatial coordinates, text and binary data, and anything else. You can adapt the structure of a documents schema just by adding new fields, making it simple to bring in new data as it becomes available.

- **Horizontal Scalability.** MongoDBs automatic sharding distributes data across fleets of commodity servers, with complete application transparency. With multiple options for scaling including range-based, hash-based and location-aware sharding MongoDB can support thousands of nodes, petabytes of data, and hundreds of thousands of ops per second without requiring you to build custom partitioning and caching layers.

- **Powerful Analytics,**In Place, In Real Time. With rich index and query support including secondary, geospatial and text search indexes as well as the aggregation framework and native MapReduce, MongoDB can run complex ad-hoc analytics and reporting in place.

4.4 Bcrypt Encrypt

bcrypt is a password hashing function designed by Niels Provos and David Mazires, based on the Blowfish cipher, and presented at USENIX in 1999. Besides incorporating a salt to protect against rainbow table attacks, bcrypt is an adaptive function: over time, the iteration count can be increased to make it slower, so it remains resistant to brute-force search attacks even with increasing computation power.
The bcrypt function is the default password hash algorithm for OpenBSD and other systems including some Linux distributions such as SUSE Linux.
There are implementations of bcrypt for C, C++, C#, Go, Java, JavaScript, Elixir, Perl, PHP, Python, Ruby and other languages.

4.4.1 How bcrypt is used

The prefix ”$2a$” or ”$2b$” (or ”$2y$”) in a hash string in a shadow password file indicates that hash string is a bcrypt hash in modular crypt format. The rest of the hash string includes the cost parameter, a 128-bit salt (Radix-64 encoded as 22 characters), and 184 bits of the resulting hash value (Radix-64 encoded as 31 characters). The Radix-64 encoding uses the unix/crypt alphabet, and is not ’standard’ Base-64. The cost parameter specifies a key expansion iteration count as a power of two, which is an input to the crypt algorithm.
For example, the shadow password record,
$2$10$N9qo8uLOickgx2ZMRZoMyeIjZAgcfl7p92ldGxad68LJZdL17lhWy specifies a cost parameter of 10, indicating 210 key expansion rounds. The salt is N9qo8uLOickgx2ZMRZoMye and the resulting hash is IjZAgcfl7p92ldGxad68LJZdL17lhWy. Per standard practice, the user’s password itself is not stored.

4.4.2 Bcrypt password hash format

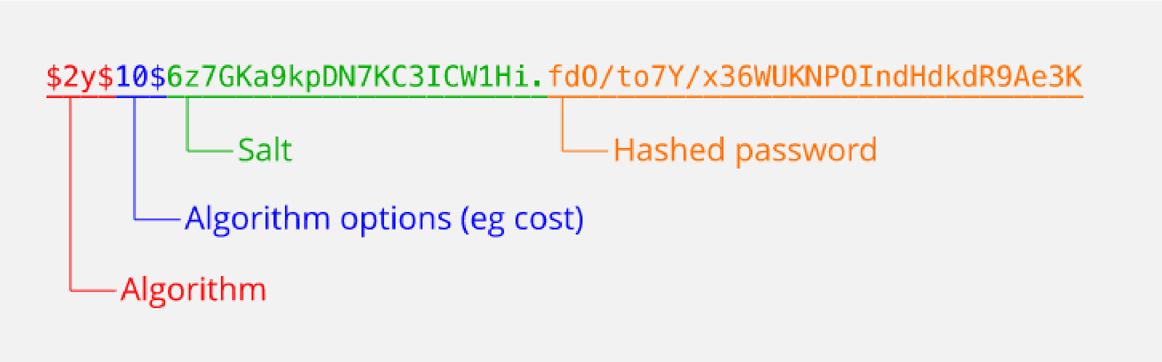

$2y$10$6z7GKa9kpDN7KC3ICW1Hi.fd0/to7Y/x36WUKNPOIndHdkdR9Ae3K
Salt
Hashed password
Algorithm options (eg cost)
Algorithm

Figure 4.2: Bcrypt password hash format

4.5 JSON Web Token (JWT)

4.5.1 What is JSON Web Token?

JSON Web Token (JWT) is an open standard (RFC 7519) that defines a compact and self-contained way for securely transmitting information between parties as a JSON object. This information can be verified and trusted because it is digitally signed. JWTs can be signed using a secret (with the **HMAC** algorithm) or a public/private key pair using **RSA** or **ECDSA**. Although JWTs can be encrypted to also provide secrecy between parties, we will focus on signed tokens. Signed tokens can verify the integrity of the claims contained within it, while encrypted tokens hide those claims from other parties. When tokens are signed using public/private key pairs, the signature also certifies that only the party holding the private key is the one that signed it.

4.5.2 When should you use JSON Web Tokens

Here are some scenarios where JSON Web Tokens are useful:

- **Authorization:** This is the most common scenario for using JWT. Once the user is logged in, each subsequent request will include the JWT, allowing the user to access routes, services, and resources that are permitted with that token. Single Sign On is a feature that widely uses JWT nowadays, because of its small overhead and its ability to be easily used across different domains.

- **Information Exchange:** JSON Web Tokens are a good way of securely transmitting information between parties. Because JWTs can be signedfor example, using public/private key pairsyou can be sure the senders are who they say they are. Additionally, as the signature is calculated using the header and the payload, you can also verify that the content hasn't been tampered with.

4.5.3 JSON Web Token structure

In its compact form, JSON Web Tokens consist of three parts separated by dots (.), which are:

- Header
- Payload
- Signature

Therefore, a JWT typically looks like the following.
xxxxx.yyyyy.zzzzz

4.5.4 Authentication Process using JWT

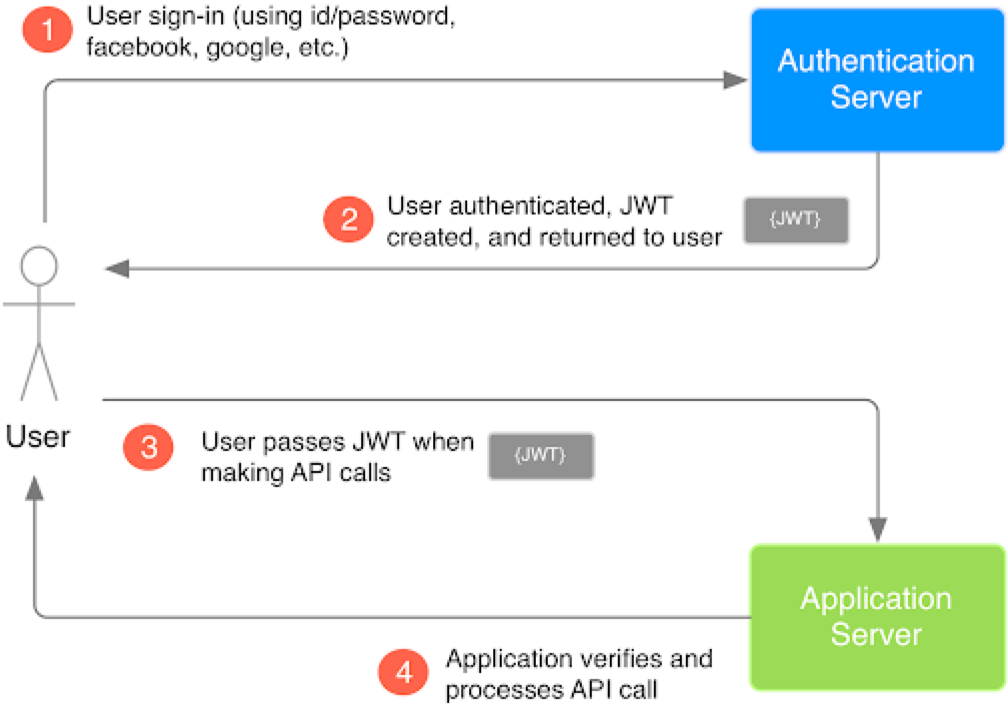

1
User sign-in (using id/password, facebook, google, etc.)
Authentication Server
2
User authenticated, JWT created, and returned to user
{JWT}
User
3
User passes JWT when making API calls
{JWT}
Application Server
4
Application verifies and processes API call

Figure 4.3: Authentication using JWT

4.5.5 Axios:

Axios is promise-based and thus we can take advantage of async and await for more readable asynchronous code. We can also intercept and cancel requests, and theres built-in client side protection against cross site request forgery. But the best part about Axios The easy to use API for fetching HTTP client.

4.6 Redux

Redux is a predictable state container for JavaScript applications. It helps you write applications that behave consistently, run in different environments (client, server, and native), and are easy to test.

Simply put, Redux is a state management tool. While its mostly used with React, it can be used with any other JavaScript framework or library. It is lightweight at 2KB (including dependencies), so you dont have to worry about it making your applications asset size bigger. With Redux, the state of your application is kept in a store and each component can access any state that it needs from application server. Lets dive a little deeper to see why you might need a state management tool.

4.6.1 Need of Redux

Most libraries like React, Angular, etc. are built with a way for components to internally manage their state without any need for an external library or tool. It does well for applications with few components but as the application grows bigger, managing states shared across components becomes a chore.

In an app where data is shared among components, it might be confusing to actually know where a state should live. Ideally, the data in a component should live in just one component. So sharing data among sibling components becomes difficult.

For instance, in React, to share data among siblings, a state has to live in the parent component. A method for updating this state is provided by this parent component and passed as props to these sibling components.

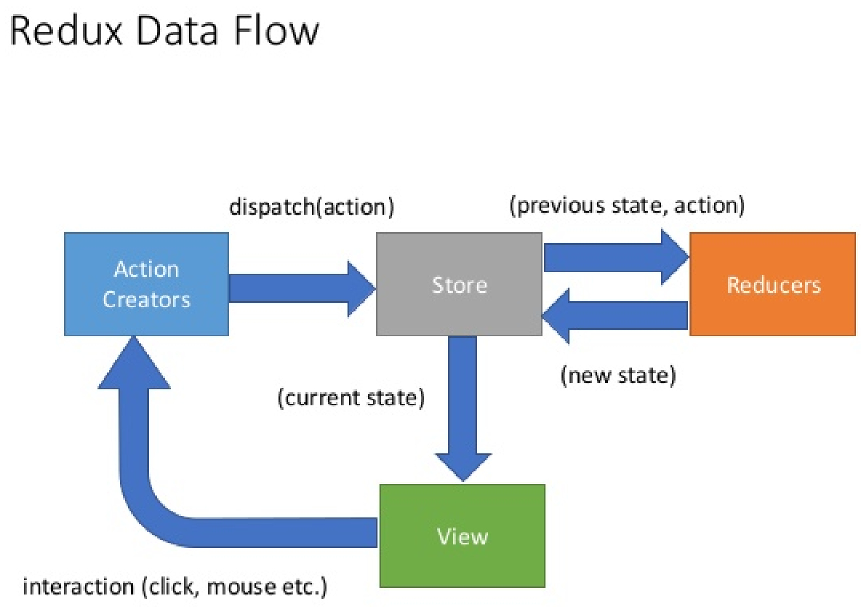


Figure 4.4: Redux data flow

Chapter 5

Project Design

5.1 UML Diagrams

5.1.1 Usecase Diagram

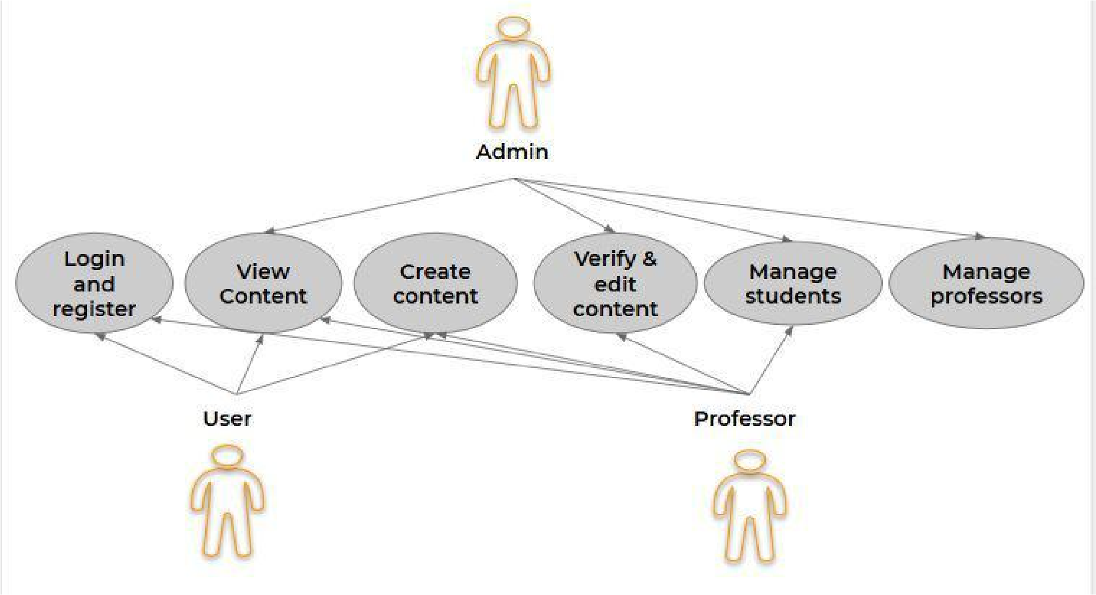


Figure 5.1: Usecase Diagram

The proposed system consists of category of users as follows:
1)Admin
2) Professor
3)Student/ Enduser
Admin has the permission to view control, manage students and manage professors.
The professor has permission to login, view content, create content, verify and edit.
Students can login and register, view content, create content. Apaer from this students can like, share and comment the content.

5.1.2 Activity Diagram

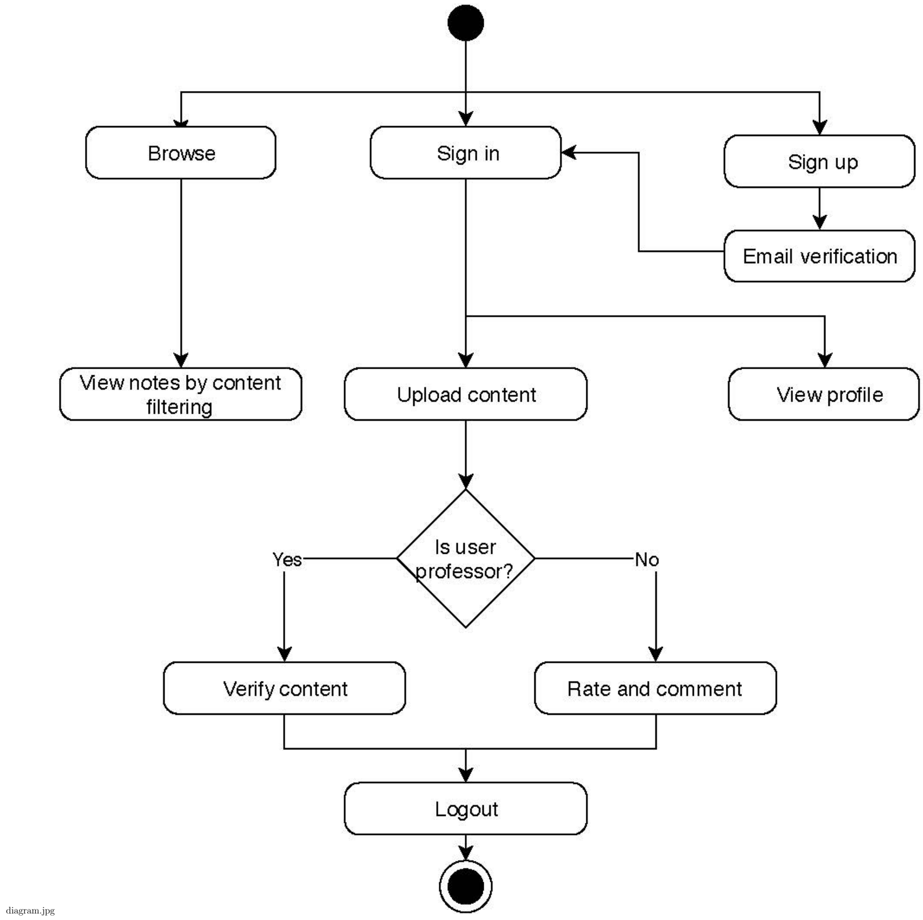


Figure 5.2: Activity Diagram

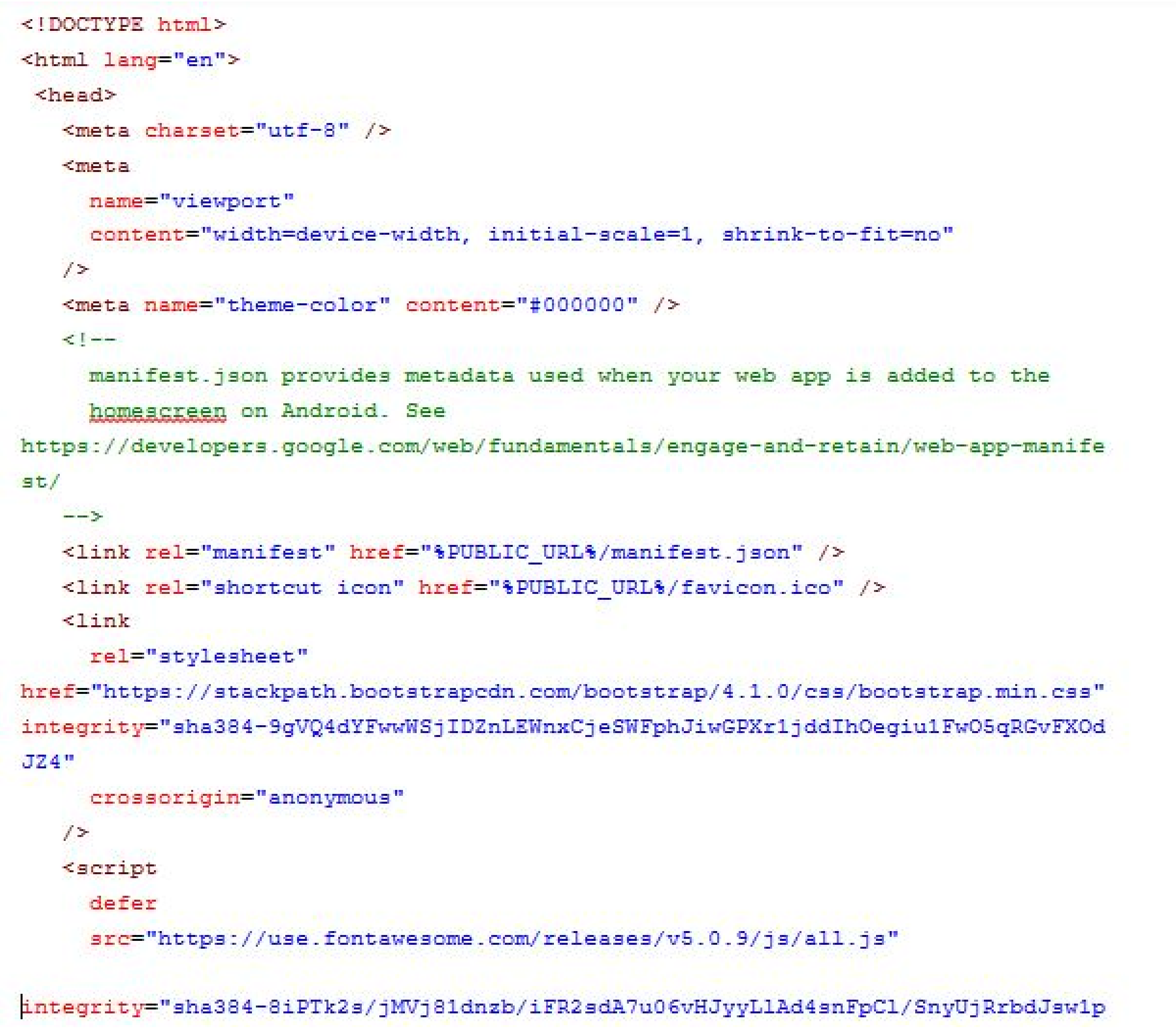
The above given activity diagram works as follows:
1)A user is able to browse the website and view notes even if they have not logged themselves in.
2)If the user wants more interaction with the website then they must log themselves in.
3)When user signs in for the first time, they need to verify email by clicking on link sent on mail.
4)After verifying their email, the user can sign in and they can perform actions like sign in and retieve and upload the content on this platform.
5) A check is made whether the user is a professor or not.
6) User can log in whenever they want.

Chapter 6

Implementation

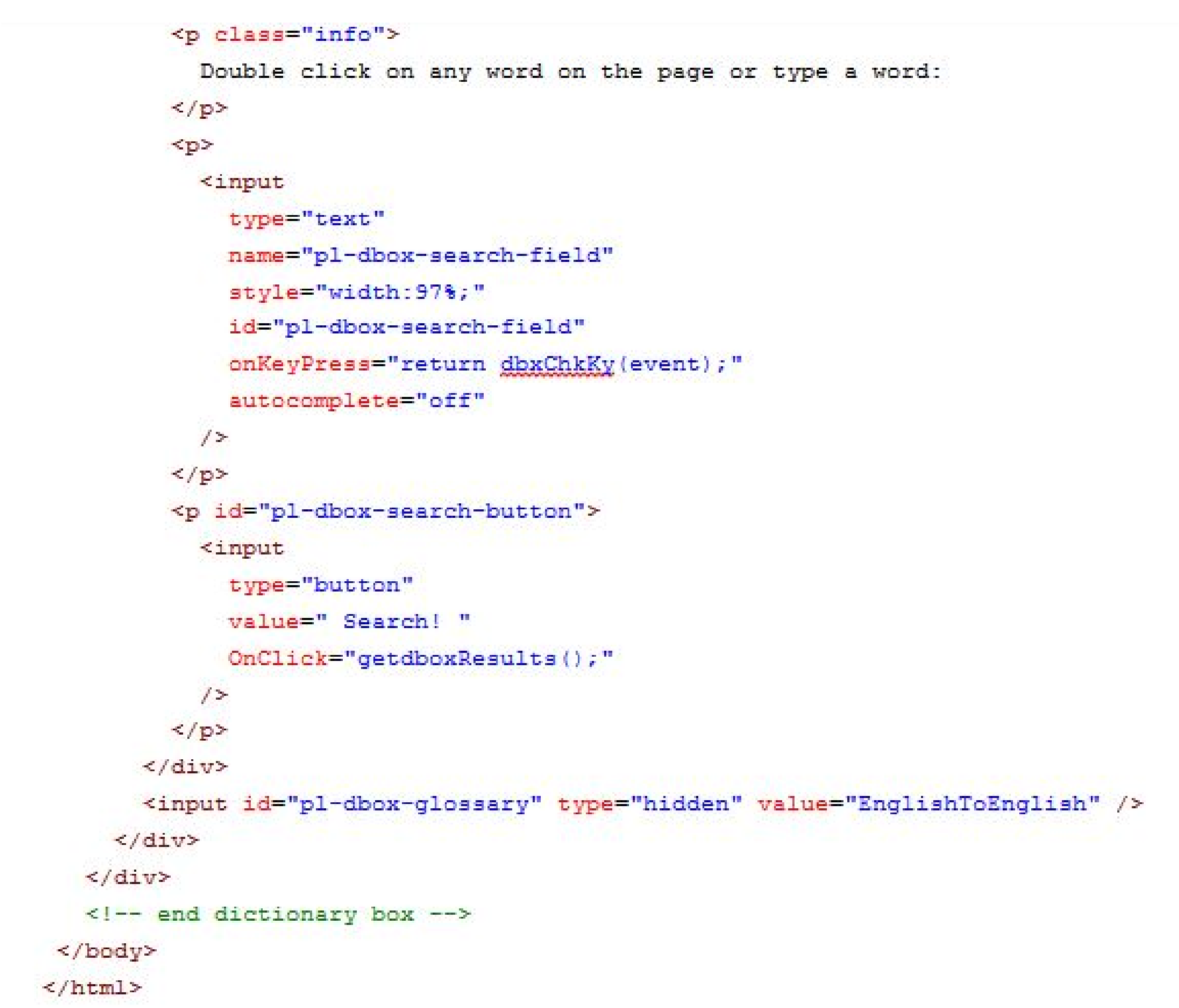
6.1 Reactjs implementation code

index.html



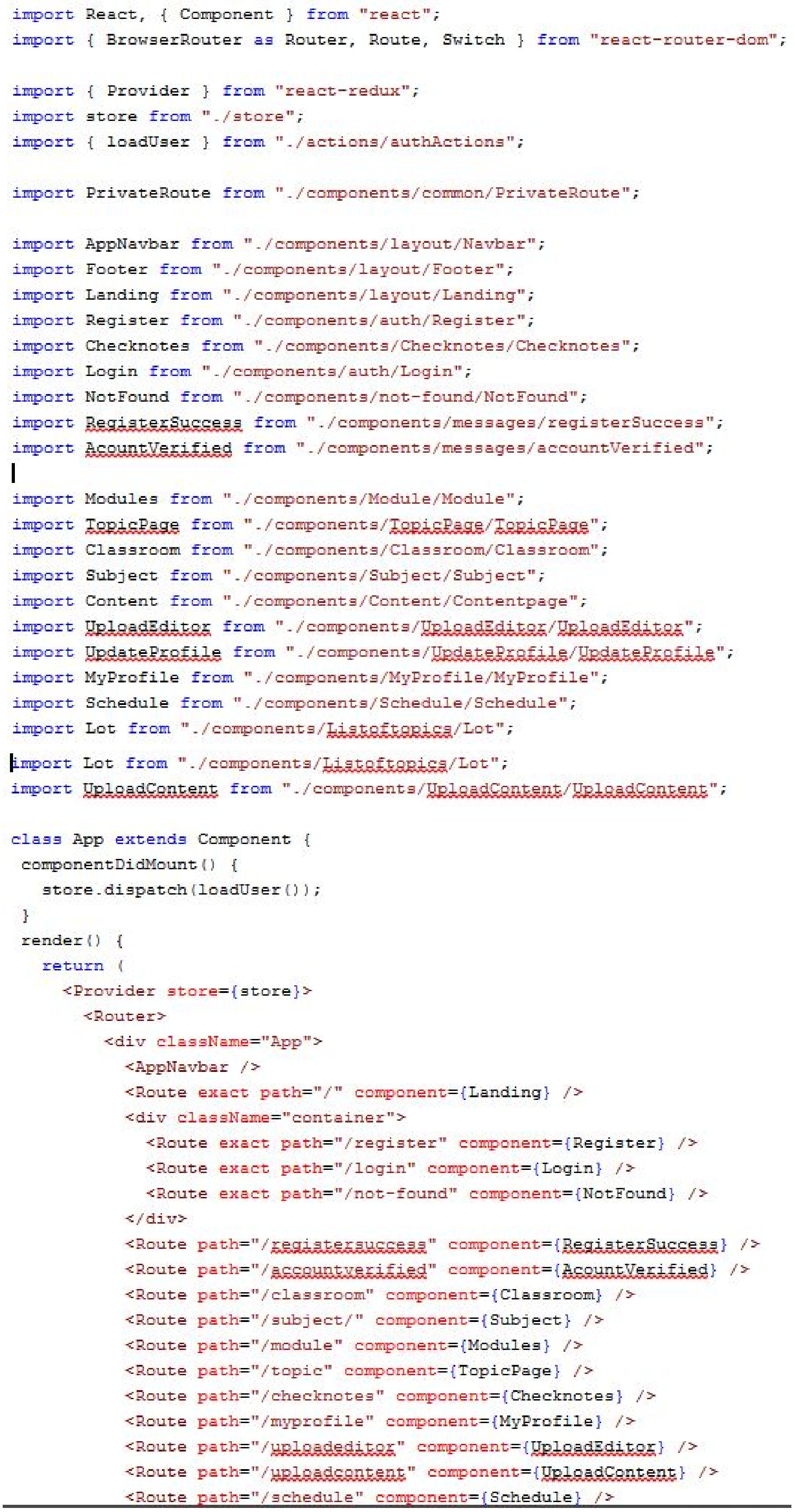
```
<!DOCTYPE html>
<html lang="en">
 <head>
   <meta charset="utf-8" />
   <meta
     name="viewport"
     content="width=device-width, initial-scale=1, shrink-to-fit=no"
   />
   <meta name="theme-color" content="#000000" />
   <!--
     manifest.json provides metadata used when your web app is added to the
     homescreen on Android. See
https://developers.google.com/web/fundamentals/engage-and-retain/web-app-manife
st/
   -->
   <link rel="manifest" href="%PUBLIC_URL%/manifest.json" />
   <link rel="shortcut icon" href="%PUBLIC_URL%/favicon.ico" />
   <link
     rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.0/css/bootstrap.min.css"
integrity="sha384-9gVQ4dYFwwWSjIDZnLEWnxCjeSWFphJiwGPXr1jddIhOegiu1FwO5qRGvFXOd
JZ4"
     crossorigin="anonymous"
   />
   <script
     defer
     src="https://use.fontawesome.com/releases/v5.0.9/js/all.js"

integrity="sha384-8iPTk2s/jMVj81dnzb/iFR2sdA7u06vHJyyLlAd4snFpCl/SnyUjRrbdJsw1p
```

```
     crossorigin="anonymous"
   ></script>

   <title>WikiNotes</title>
 </head>

 <body>
   <noscript>
     You need to enable JavaScript to run this app.
   </noscript>
   <div id="root"></div>

   <script
     src="https://code.jquery.com/jquery-3.3.1.slim.min.js"

integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6j
izo"
     crossorigin="anonymous"
   ></script>
   <script

src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.0/umd/popper.min.js"

integrity="sha384-cs/chFZiN24E4KMATLdqdvsezGxaGsi4hLGOzlXwp5UZB1LY//20VyM2taTB4
QvJ"
     crossorigin="anonymous"
   ></script>
   <script

src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.0/js/bootstrap.min.js"
integrity="sha384-uefMccjFJAIv6A+rW+L4AHf99KvxDjWSu1z9VI8SKNVmz4sk7buKt/6v9KI65
qnm"
     crossorigin="anonymous"
   ></script>
   <!-- begin dictionary box -->
   <link
     rel="stylesheet"
     type="text/css"
     media="all"
     href="https://dictionarybox.com/dic_box_v3.css"
   />
   <script
     type="text/javascript"
     src="https://ajax.googleapis.com/ajax/libs/jquery/1.6.4/jquery.min.js"
   ></script>
   <script
     type="text/javascript"
     src="https://dictionarybox.com/dic_box.js"
   ></script>
   <div id="pl-dbox">
     <a class="pl-dbox-title" href="javascript:void(0);dboxCursorLoc();"
       ><span id="pl-dbox-title-text">Dictionary</span></a
     >
     <div id="pl-dbox-content">
       <div id="pl-dboxFlags">
         <ul>
           <li>
             <img
```

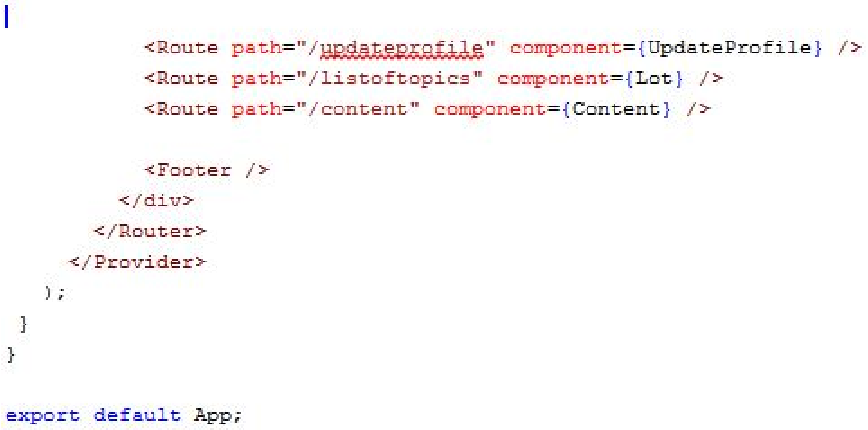


```
         <p class="info">
           Double click on any word on the page or type a word:
         </p>
         <p>
           <input
             type="text"
             name="pl-dbox-search-field"
             style="width:97%;"
             id="pl-dbox-search-field"
             onKeyPress="return dbxChkKy(event);"
             autocomplete="off"
           />
         </p>
         <p id="pl-dbox-search-button">
           <input
             type="button"
             value=" Search! "
             OnClick="getdboxResults();"
           />
         </p>
       </div>
       <input id="pl-dbox-glossary" type="hidden" value="EnglishToEnglish" />
     </div>
   </div>
   <!-- end dictionary box -->
 </body>
</html>
```

App.js for client

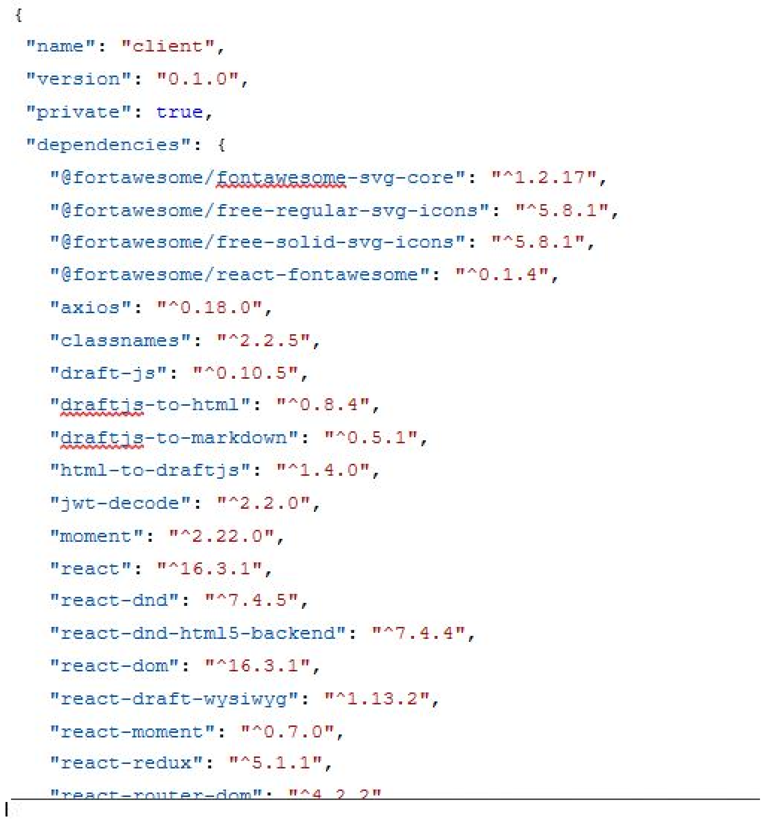


```
import React, { Component } from "react";
import { BrowserRouter as Router, Route, Switch } from "react-router-dom";

import { Provider } from "react-redux";
import store from "./store";
import { loadUser } from "./actions/authActions";

import PrivateRoute from "./components/common/PrivateRoute";

import AppNavbar from "./components/layout/Navbar";
import Footer from "./components/layout/Footer";
import Landing from "./components/layout/Landing";
import Register from "./components/auth/Register";
import Checknotes from "./components/Checknotes/Checknotes";
import Login from "./components/auth/Login";
import NotFound from "./components/not-found/NotFound";
import RegisterSuccess from "./components/messages/registerSuccess";
import AcountVerified from "./components/messages/accountVerified";

import Modules from "./components/Module/Module";
import TopicPage from "./components/TopicPage/TopicPage";
import Classroom from "./components/Classroom/Classroom";
import Subject from "./components/Subject/Subject";
import Content from "./components/Content/Contentpage";
import UploadEditor from "./components/UploadEditor/UploadEditor";
import UpdateProfile from "./components/UpdateProfile/UpdateProfile";
import MyProfile from "./components/MyProfile/MyProfile";
import Schedule from "./components/Schedule/Schedule";
import Lot from "./components/Listoftopics/Lot";
import Lot from "./components/Listoftopics/Lot";
import UploadContent from "./components/UploadContent/UploadContent";

class App extends Component {
 componentDidMount() {
   store.dispatch(loadUser());
 }
 render() {
   return (
     <Provider store={store}>
       <Router>
         <div className="App">
           <AppNavbar />
           <Route exact path="/" component={Landing} />
           <div className="container">
             <Route exact path="/register" component={Register} />
             <Route exact path="/login" component={Login} />
             <Route exact path="/not-found" component={NotFound} />
           </div>
           <Route path="/registersuccess" component={RegisterSuccess} />
           <Route path="/accountverified" component={AcountVerified} />
           <Route path="/classroom" component={Classroom} />
           <Route path="/subject/" component={Subject} />
           <Route path="/module" component={Modules} />
           <Route path="/topic" component={TopicPage} />
           <Route path="/checknotes" component={Checknotes} />
           <Route path="/myprofile" component={MyProfile} />
           <Route path="/uploadeditor" component={UploadEditor} />
           <Route path="/uploadcontent" component={UploadContent} />
           <Route path="/schedule" component={Schedule} />
```

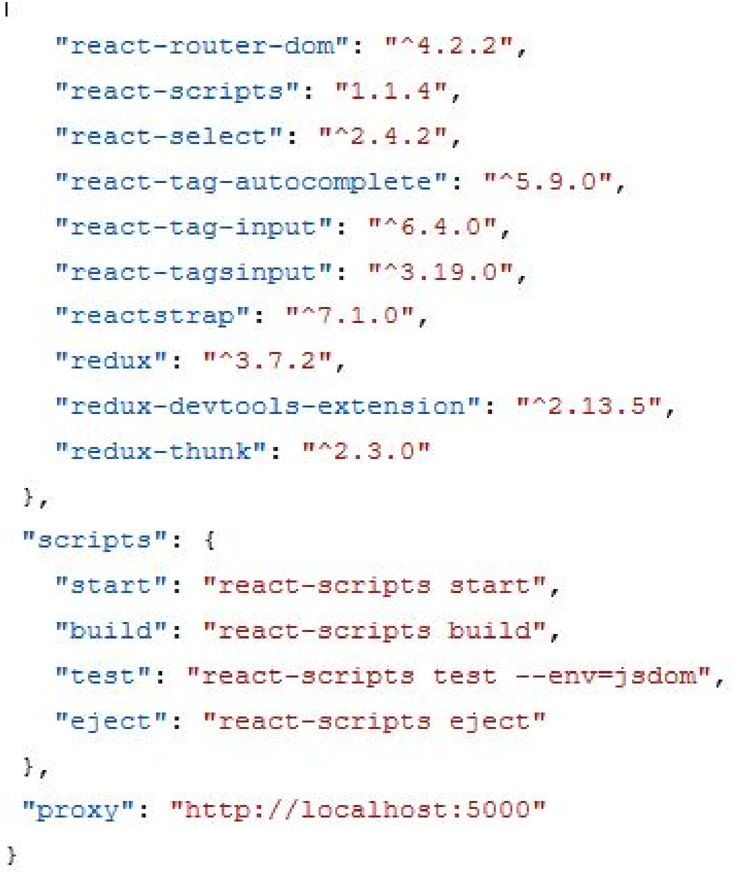


```
           <Route path="/updateprofile" component={UpdateProfile} />
           <Route path="/listoftopics" component={Lot} />
           <Route path="/content" component={Content} />

           <Footer />
         </div>
       </Router>
     </Provider>
   );
 }
}

export default App;
```

Package.json for client

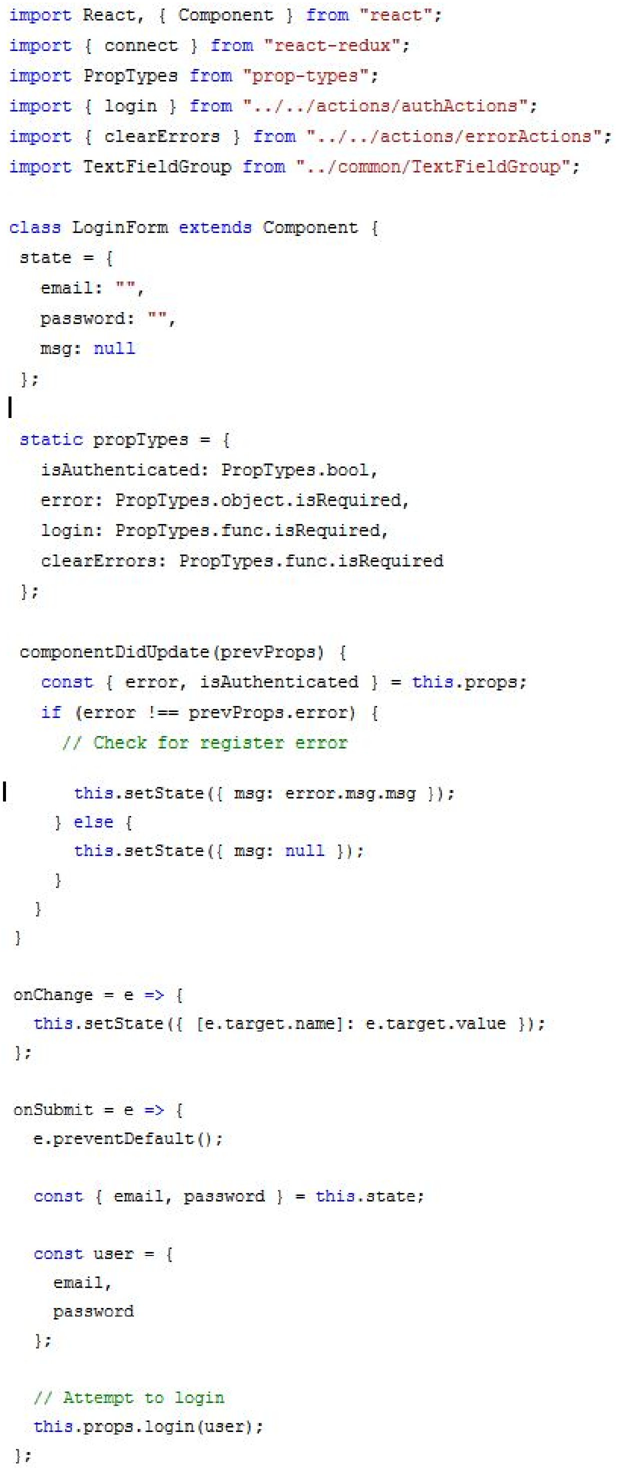


```
{
 "name": "client",
 "version": "0.1.0",
 "private": true,
 "dependencies": {
   "@fortawesome/fontawesome-svg-core": "^1.2.17",
   "@fortawesome/free-regular-svg-icons": "^5.8.1",
   "@fortawesome/free-solid-svg-icons": "^5.8.1",
   "@fortawesome/react-fontawesome": "^0.1.4",
   "axios": "^0.18.0",
   "classnames": "^2.2.5",
   "draft-js": "^0.10.5",
   "draftjs-to-html": "^0.8.4",
   "draftjs-to-markdown": "^0.5.1",
   "html-to-draftjs": "^1.4.0",
   "jwt-decode": "^2.2.0",
   "moment": "^2.22.0",
   "react": "^16.3.1",
   "react-dnd": "^7.4.5",
   "react-dnd-html5-backend": "^7.4.4",
   "react-dom": "^16.3.1",
   "react-draft-wysiwyg": "^1.13.2",
   "react-moment": "^0.7.0",
   "react-redux": "^5.1.1",
```



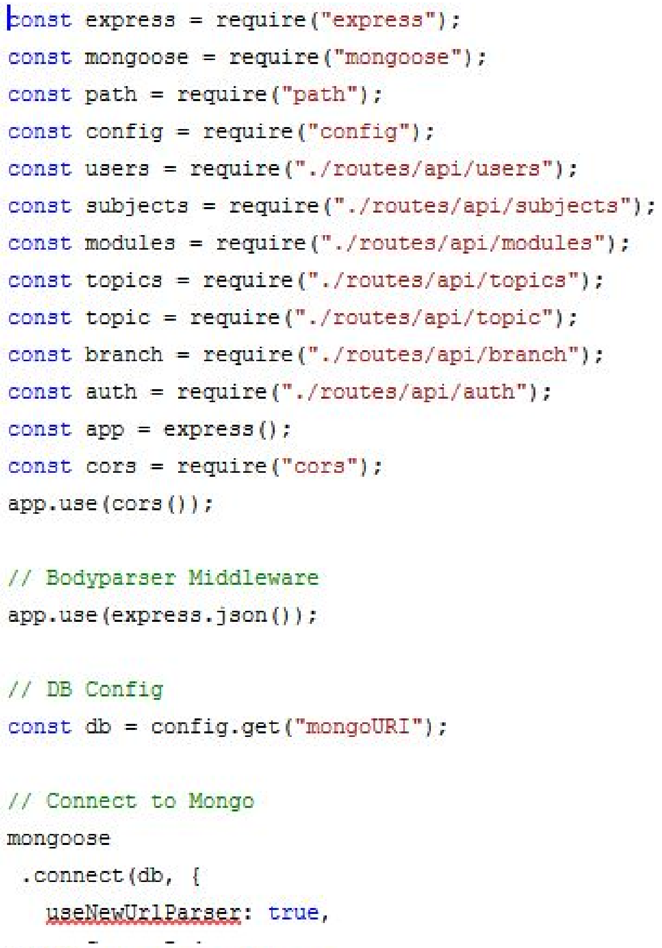
```
   "react-router-dom": "^4.2.2",
   "react-scripts": "1.1.4",
   "react-select": "^2.4.2",
   "react-tag-autocomplete": "^5.9.0",
   "react-tag-input": "^6.4.0",
   "react-tagsinput": "^3.19.0",
   "reactstrap": "^7.1.0",
   "redux": "^3.7.2",
   "redux-devtools-extension": "^2.13.5",
   "redux-thunk": "^2.3.0"
 },
 "scripts": {
   "start": "react-scripts start",
   "build": "react-scripts build",
   "test": "react-scripts test --env=jsdom",
   "eject": "react-scripts eject"
 },
 "proxy": "http://localhost:5000"
}
```

login.js

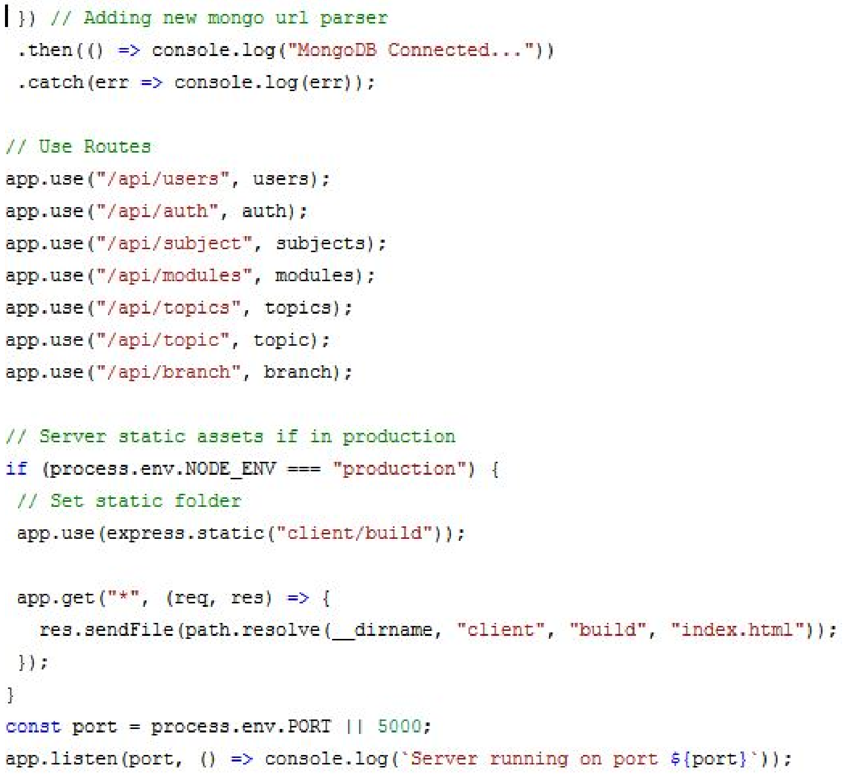


```
import React, { Component } from "react";
import { connect } from "react-redux";
import PropTypes from "prop-types";
import { login } from "../../actions/authActions";
import { clearErrors } from "../../actions/errorActions";
import TextFieldGroup from "../common/TextFieldGroup";

class LoginForm extends Component {
 state = {
   email: "",
   password: "",
   msg: null
 };

 static propTypes = {
   isAuthenticated: PropTypes.bool,
   error: PropTypes.object.isRequired,
   login: PropTypes.func.isRequired,
   clearErrors: PropTypes.func.isRequired
 };

 componentDidUpdate(prevProps) {
   const { error, isAuthenticated } = this.props;
   if (error !== prevProps.error) {
     // Check for register error

      this.setState({ msg: error.msg.msg });
    } else {
      this.setState({ msg: null });
    }
  }
}

onChange = e => {
  this.setState({ [e.target.name]: e.target.value });
};

onSubmit = e => {
  e.preventDefault();

  const { email, password } = this.state;

  const user = {
    email,
    password
  };

  // Attempt to login
  this.props.login(user);
};
```

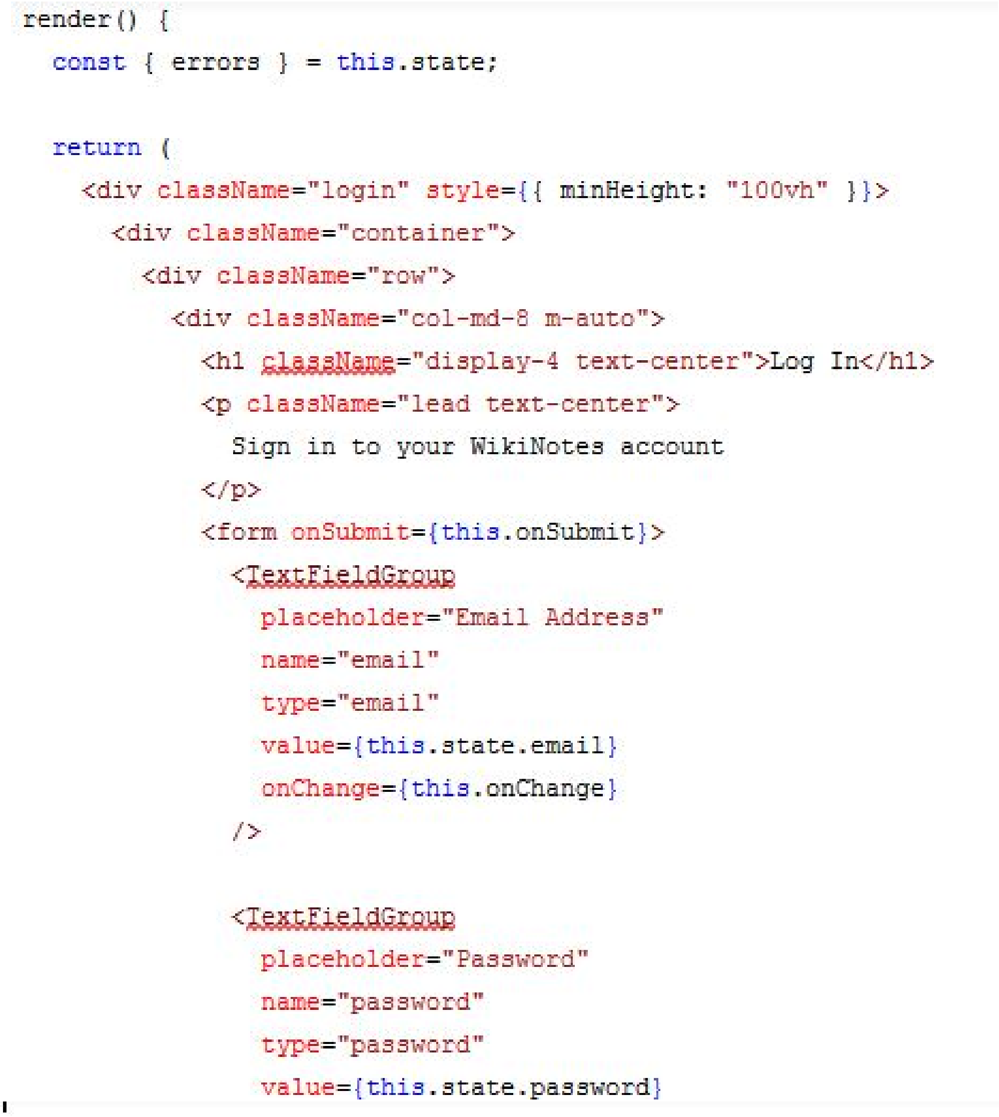
server.js



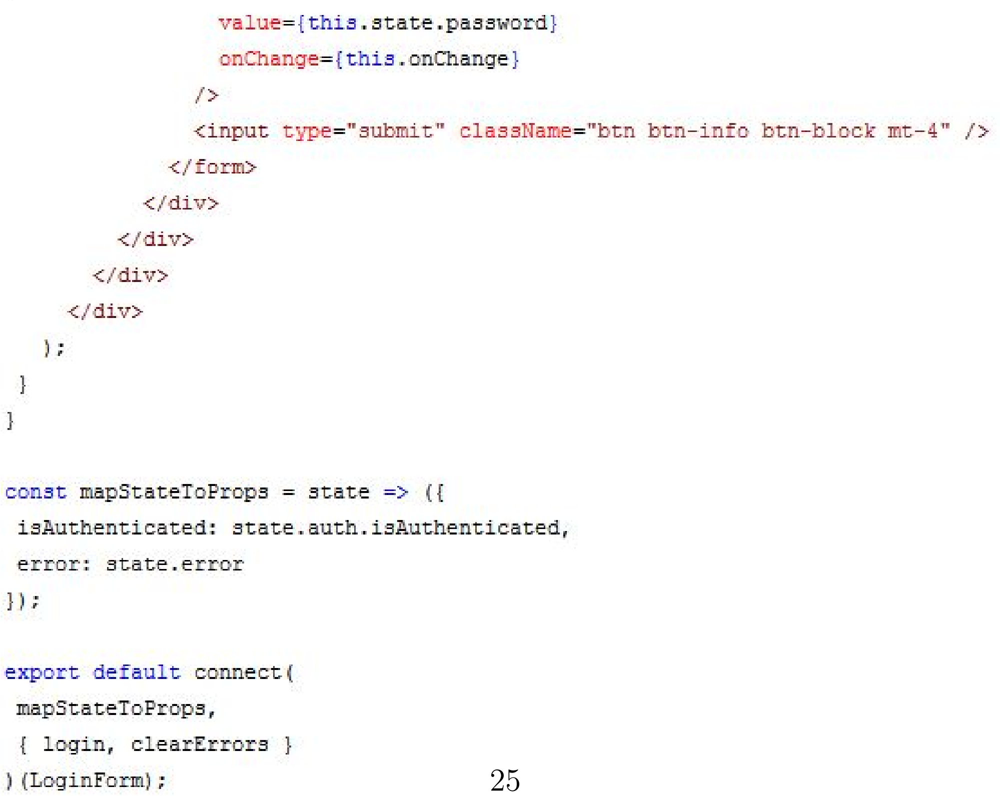
```
const express = require("express");
const mongoose = require("mongoose");
const path = require("path");
const config = require("config");
const users = require("./routes/api/users");
const subjects = require("./routes/api/subjects");
const modules = require("./routes/api/modules");
const topics = require("./routes/api/topics");
const topic = require("./routes/api/topic");
const branch = require("./routes/api/branch");
const auth = require("./routes/api/auth");
const app = express();
const cors = require("cors");
app.use(cors());

// Bodyparser Middleware
app.use(express.json());

// DB Config
const db = config.get("mongoURI");

// Connect to Mongo
mongoose
  .connect(db, {
    useNewUrlParser: true,
```



```
}) // Adding new mongo url parser
 .then(() => console.log("MongoDB Connected..."))
 .catch(err => console.log(err));

// Use Routes
app.use("/api/users", users);
app.use("/api/auth", auth);
app.use("/api/subject", subjects);
app.use("/api/modules", modules);
app.use("/api/topics", topics);
app.use("/api/topic", topic);
app.use("/api/branch", branch);

// Server static assets if in production
if (process.env.NODE_ENV === "production") {
 // Set static folder
 app.use(express.static("client/build"));

 app.get("*", (req, res) => {
   res.sendFile(path.resolve(__dirname, "client", "build", "index.html"));
 });
}
const port = process.env.PORT || 5000;
app.listen(port, () => console.log(`Server running on port ${port}`));
```

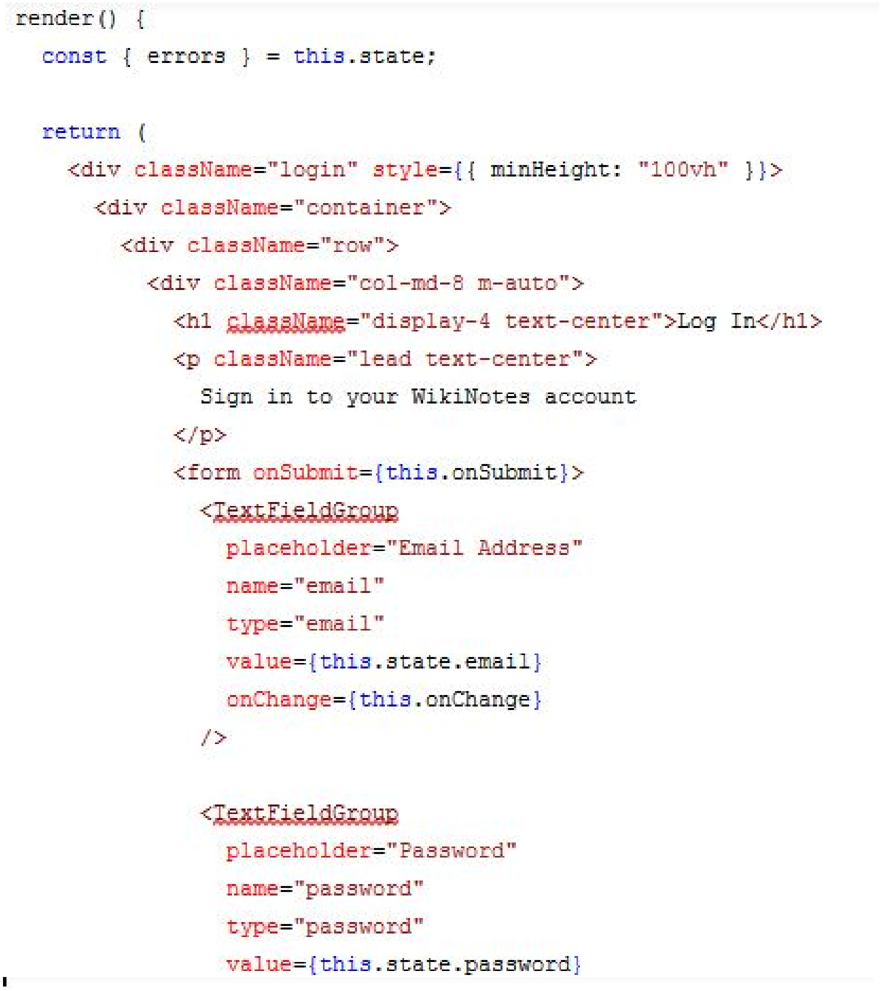


```
 render() {
   const { errors } = this.state;

   return (
     <div className="login" style={{ minHeight: "100vh" }}>
       <div className="container">
         <div className="row">
           <div className="col-md-8 m-auto">
             <h1 className="display-4 text-center">Log In</h1>
             <p className="lead text-center">
               Sign in to your WikiNotes account
             </p>
             <form onSubmit={this.onSubmit}>
               <TextFieldGroup
                 placeholder="Email Address"
                 name="email"
                 type="email"
                 value={this.state.email}
                 onChange={this.onChange}
               />

               <TextFieldGroup
                 placeholder="Password"
                 name="password"
                 type="password"
                 value={this.state.password}
```

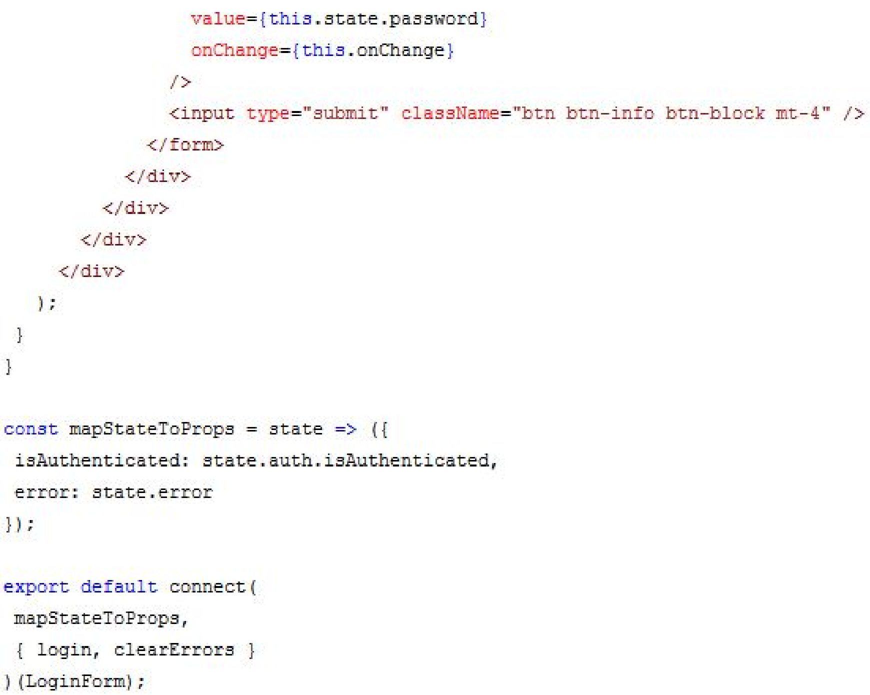


```
                 value={this.state.password}
                 onChange={this.onChange}
               />
               <input type="submit" className="btn btn-info btn-block mt-4" />
             </form>
           </div>
         </div>
       </div>
     </div>
   );
 }
}

const mapStateToProps = state => ({
 isAuthenticated: state.auth.isAuthenticated,
 error: state.error
});

export default connect(
 mapStateToProps,
 { login, clearErrors }
)(LoginForm);
```

package.json of Server

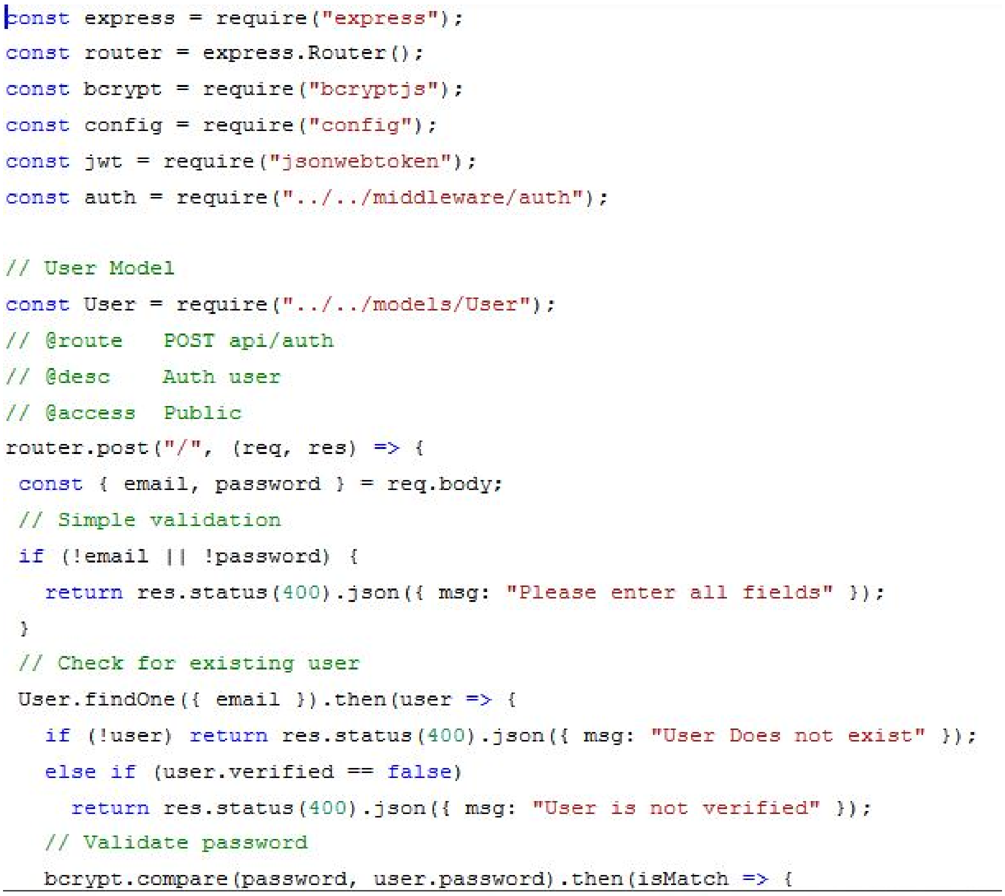


```
render() {
  const { errors } = this.state;

  return (
    <div className="login" style={{ minHeight: "100vh" }}>
      <div className="container">
        <div className="row">
          <div className="col-md-8 m-auto">
            <h1 className="display-4 text-center">Log In</h1>
            <p className="lead text-center">
              Sign in to your WikiNotes account
            </p>
            <form onSubmit={this.onSubmit}>
              <TextFieldGroup
                placeholder="Email Address"
                name="email"
                type="email"
                value={this.state.email}
                onChange={this.onChange}
              />

              <TextFieldGroup
                placeholder="Password"
                name="password"
                type="password"
                value={this.state.password}
```

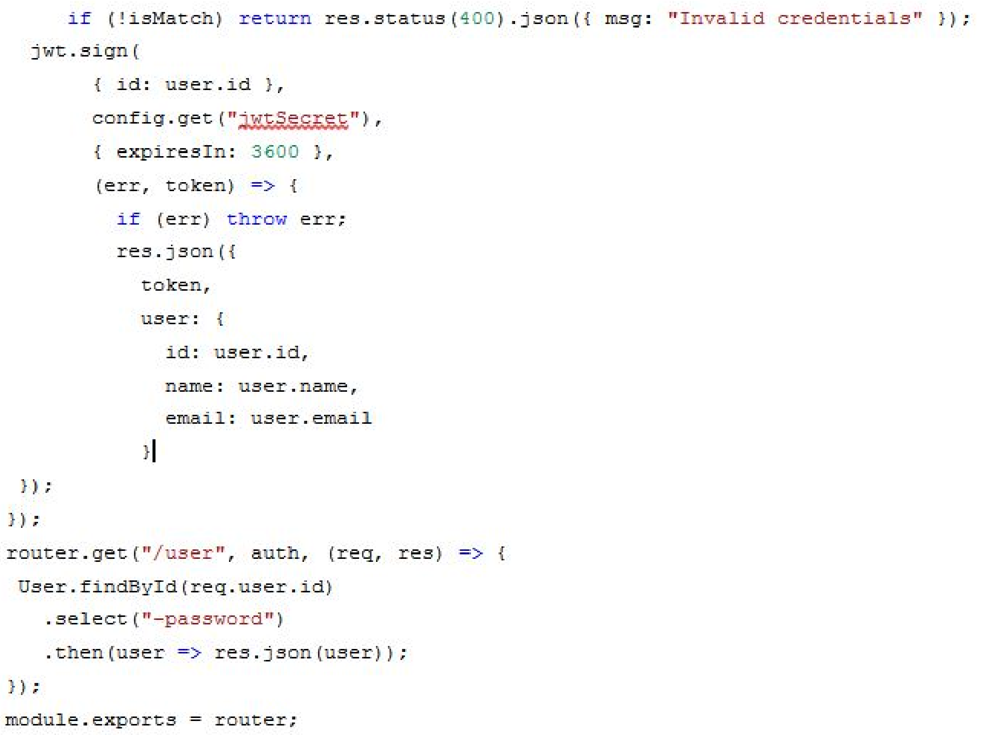


```
                 value={this.state.password}
                 onChange={this.onChange}
               />
               <input type="submit" className="btn btn-info btn-block mt-4" />
             </form>
           </div>
         </div>
       </div>
     </div>
   );
 }
}

const mapStateToProps = state => ({
 isAuthenticated: state.auth.isAuthenticated,
 error: state.error
});

export default connect(
 mapStateToProps,
 { login, clearErrors }
)(LoginForm);
```

Auth.js

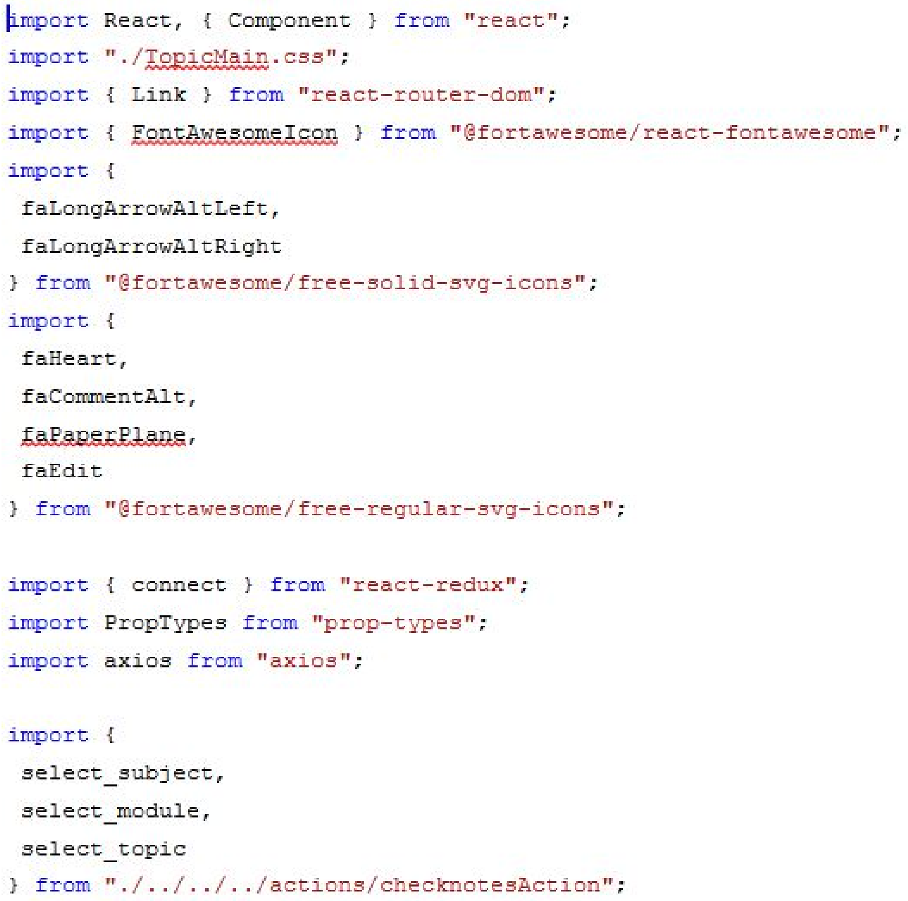


```
const express = require("express");
const router = express.Router();
const bcrypt = require("bcryptjs");
const config = require("config");
const jwt = require("jsonwebtoken");
const auth = require("../../middleware/auth");

// User Model
const User = require("../../models/User");
// @route   POST api/auth
// @desc    Auth user
// @access  Public
router.post("/", (req, res) => {
 const { email, password } = req.body;
 // Simple validation
 if (!email || !password) {
   return res.status(400).json({ msg: "Please enter all fields" });
 }
 // Check for existing user
 User.findOne({ email }).then(user => {
   if (!user) return res.status(400).json({ msg: "User Does not exist" });
   else if (user.verified == false)
     return res.status(400).json({ msg: "User is not verified" });
   // Validate password
   bcrypt.compare(password, user.password).then(isMatch => {
```

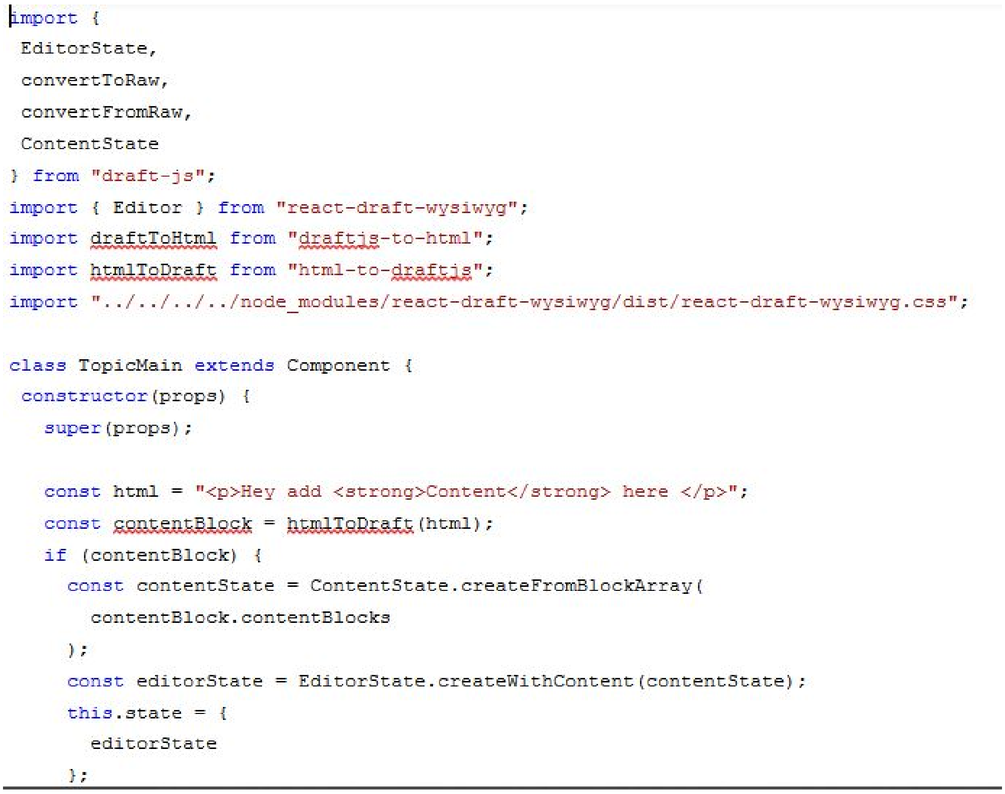


```
     if (!isMatch) return res.status(400).json({ msg: "Invalid credentials" });
  jwt.sign(
       { id: user.id },
       config.get("jwtSecret"),
       { expiresIn: 3600 },
       (err, token) => {
         if (err) throw err;
         res.json({
           token,
           user: {
             id: user.id,
             name: user.name,
             email: user.email
           }
 });
});
router.get("/user", auth, (req, res) => {
 User.findById(req.user.id)
   .select("-password")
   .then(user => res.json(user));
});
module.exports = router;
```

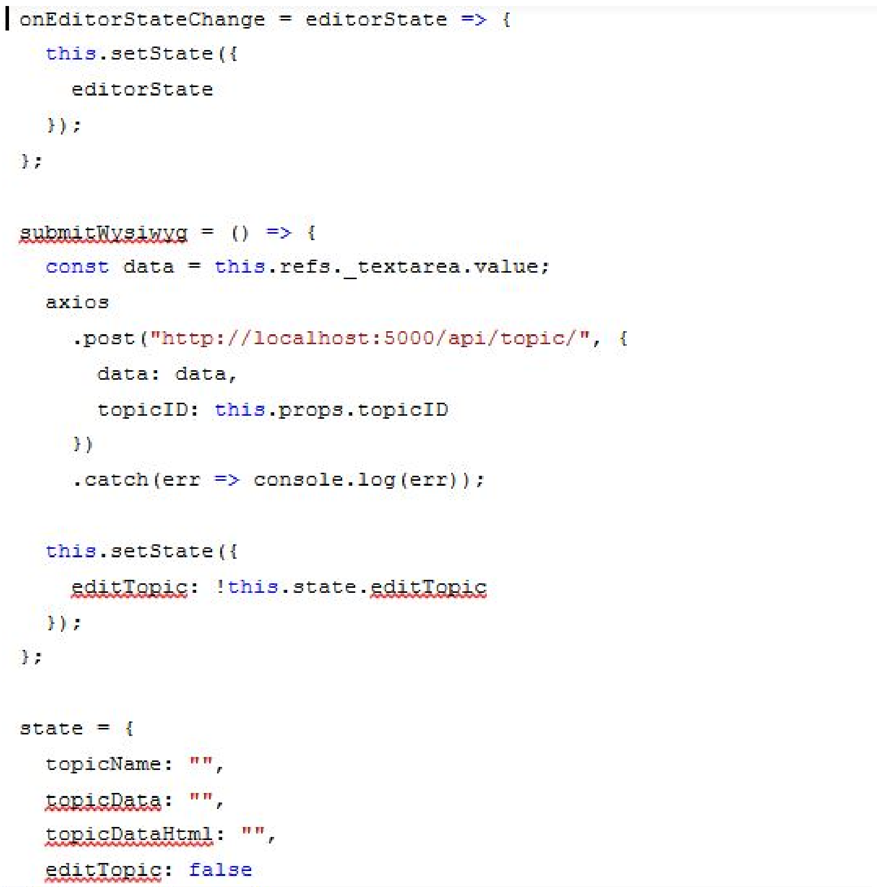
Content Page



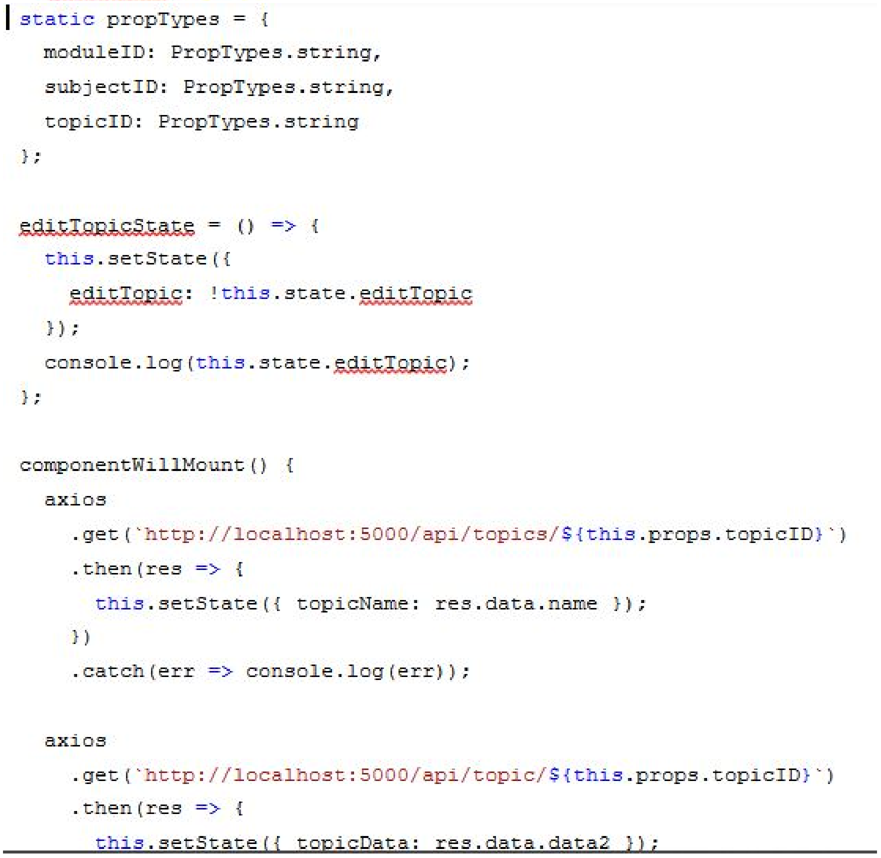
```
import React, { Component } from "react";
import "./TopicMain.css";
import { Link } from "react-router-dom";
import { FontAwesomeIcon } from "@fortawesome/react-fontawesome";
import {
 faLongArrowAltLeft,
 faLongArrowAltRight
} from "@fortawesome/free-solid-svg-icons";
import {
 faHeart,
 faCommentAlt,
 faPaperPlane,
 faEdit
} from "@fortawesome/free-regular-svg-icons";

import { connect } from "react-redux";
import PropTypes from "prop-types";
import axios from "axios";

import {
 select_subject,
 select_module,
 select_topic
} from "./../../../../actions/checknotesAction";
```



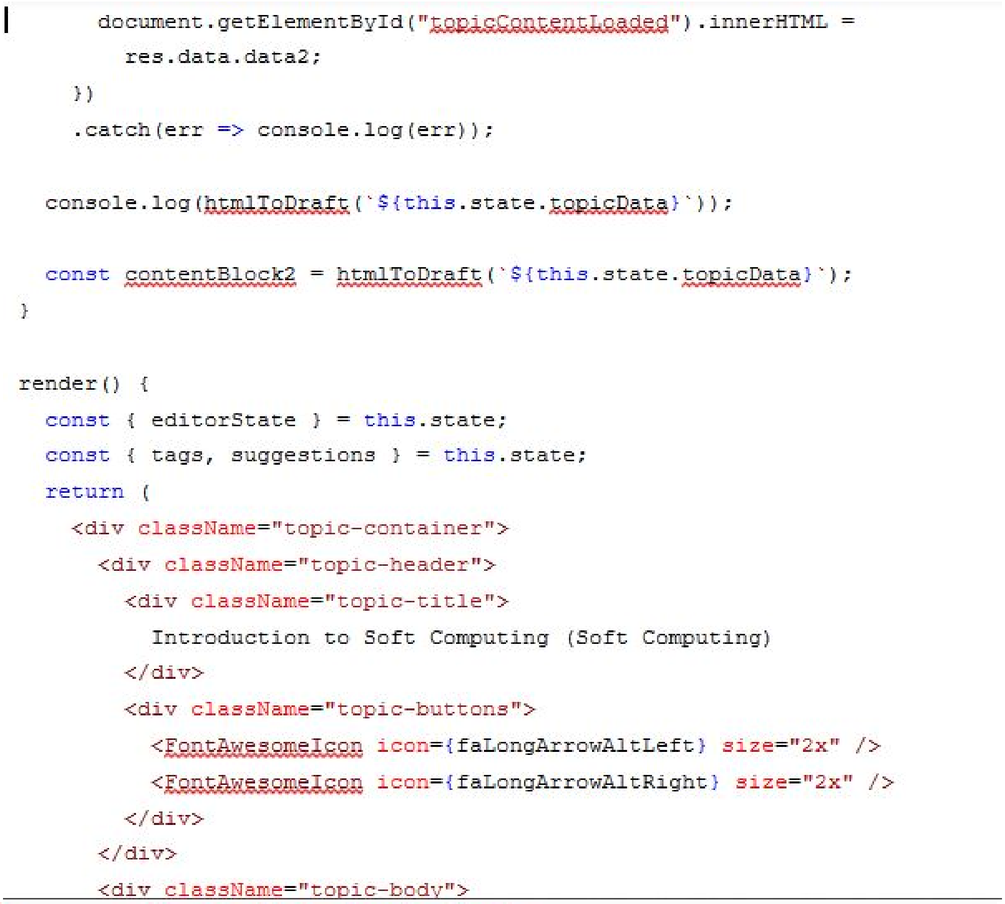
```
import {
 EditorState,
 convertToRaw,
 convertFromRaw,
 ContentState
} from "draft-js";
import { Editor } from "react-draft-wysiwyg";
import draftToHtml from "draftjs-to-html";
import htmlToDraft from "html-to-draftjs";
import "../../../../node_modules/react-draft-wysiwyg/dist/react-draft-wysiwyg.css";

class TopicMain extends Component {
 constructor(props) {
   super(props);

   const html = "<p>Hey add <strong>Content</strong> here </p>";
   const contentBlock = htmlToDraft(html);
   if (contentBlock) {
     const contentState = ContentState.createFromBlockArray(
       contentBlock.contentBlocks
     );
     const editorState = EditorState.createWithContent(contentState);
     this.state = {
       editorState
     };
```



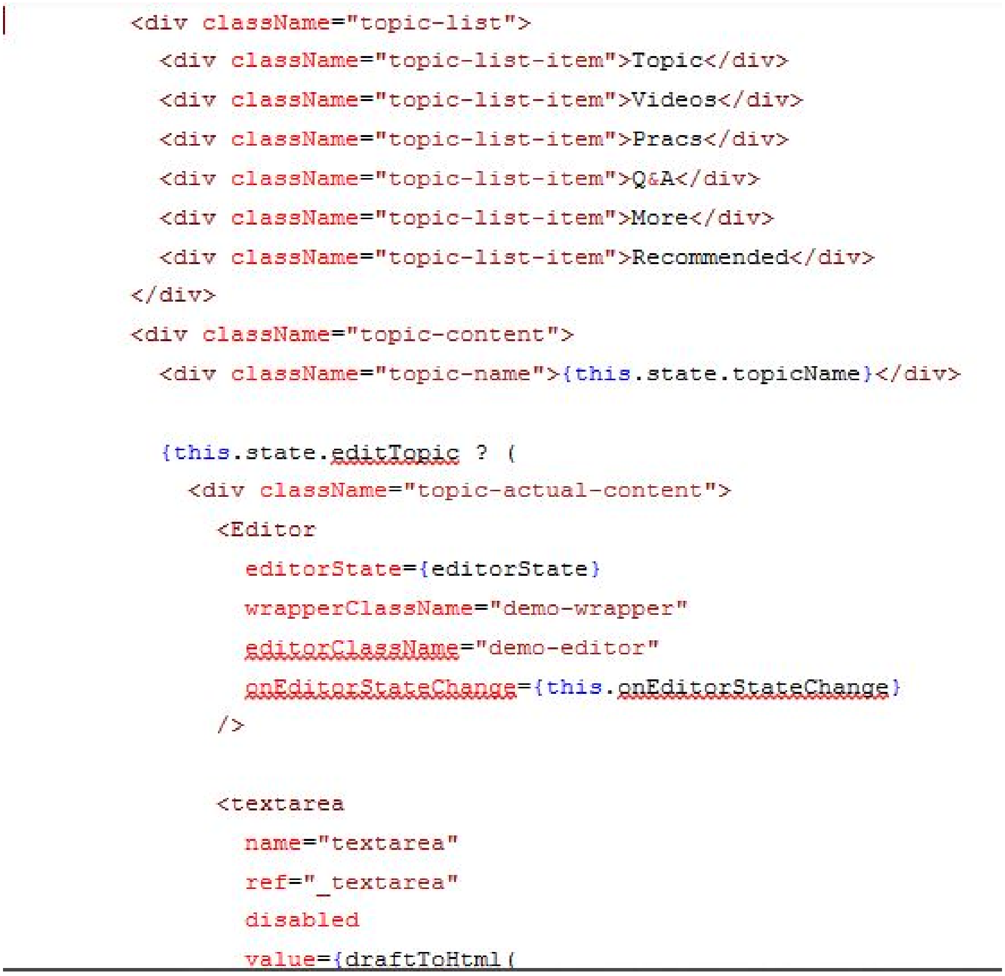
```
onEditorStateChange = editorState => {
  this.setState({
    editorState
  });
};

submitWysiwyg = () => {
  const data = this.refs._textarea.value;
  axios
    .post("http://localhost:5000/api/topic/", {
      data: data,
      topicID: this.props.topicID
    })
    .catch(err => console.log(err));

  this.setState({
    editTopic: !this.state.editTopic
  });
};

state = {
  topicName: "",
  topicData: "",
  topicDataHtml: "",
  editTopic: false
```



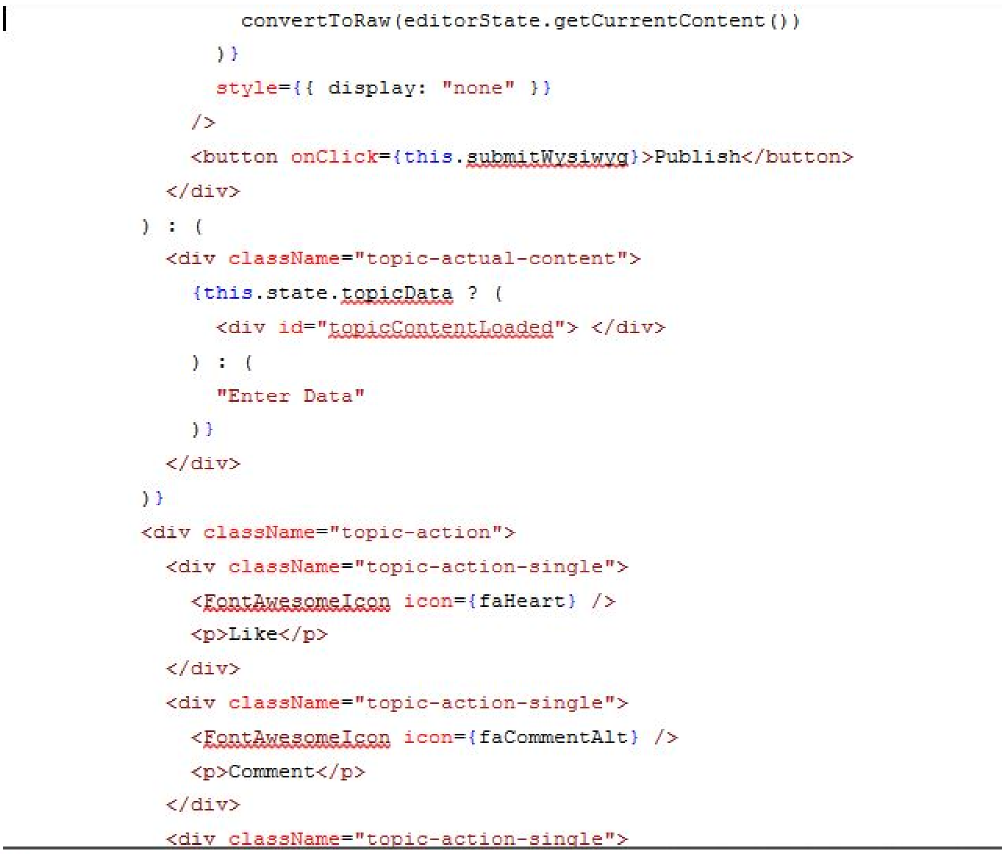
```
static propTypes = {
  moduleID: PropTypes.string,
  subjectID: PropTypes.string,
  topicID: PropTypes.string
};

editTopicState = () => {
  this.setState({
    editTopic: !this.state.editTopic
  });
  console.log(this.state.editTopic);
};

componentWillMount() {
  axios
    .get(`http://localhost:5000/api/topics/${this.props.topicID}`)
    .then(res => {
      this.setState({ topicName: res.data.name });
    })
    .catch(err => console.log(err));

  axios
    .get(`http://localhost:5000/api/topic/${this.props.topicID}`)
    .then(res => {
      this.setState({ topicData: res.data.data2 });
```



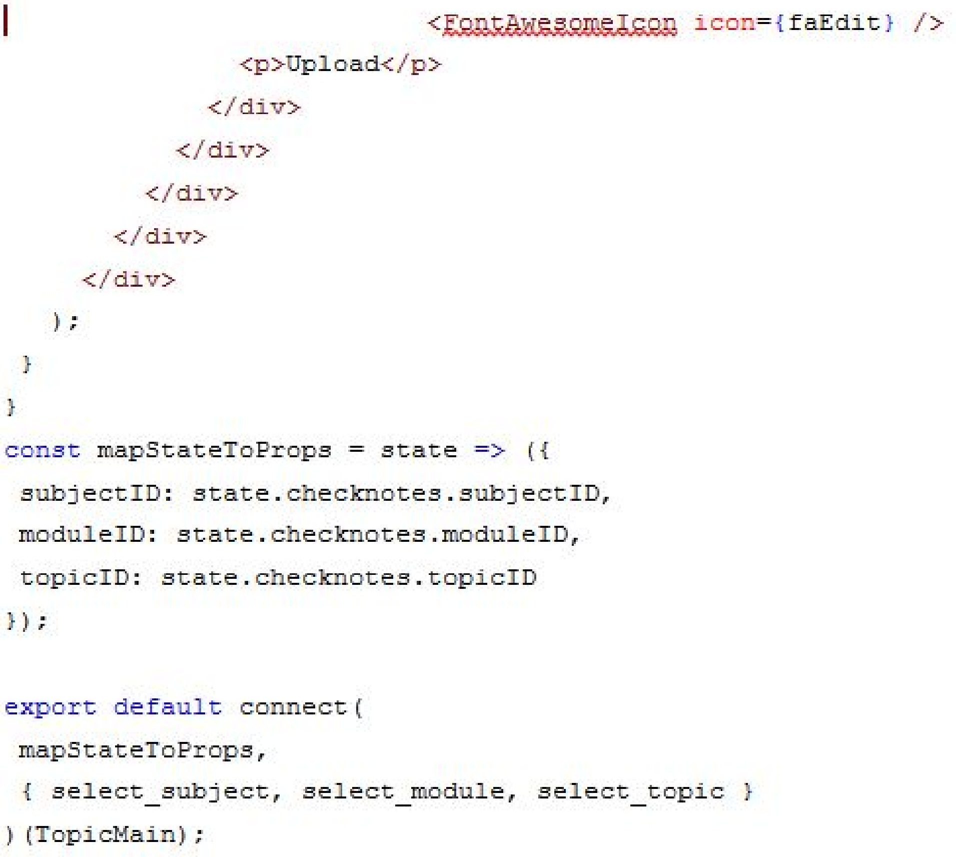
```
      document.getElementById("topicContentLoaded").innerHTML =
        res.data.data2;
    })
    .catch(err => console.log(err));

  console.log(htmlToDraft(`${this.state.topicData}`));

  const contentBlock2 = htmlToDraft(`${this.state.topicData}`);
}

render() {
  const { editorState } = this.state;
  const { tags, suggestions } = this.state;
  return (
    <div className="topic-container">
      <div className="topic-header">
        <div className="topic-title">
          Introduction to Soft Computing (Soft Computing)
        </div>
        <div className="topic-buttons">
          <FontAwesomeIcon icon={faLongArrowAltLeft} size="2x" />
          <FontAwesomeIcon icon={faLongArrowAltRight} size="2x" />
        </div>
      </div>
      <div className="topic-body">
```



```
        <div className="topic-list">
          <div className="topic-list-item">Topic</div>
          <div className="topic-list-item">Videos</div>
          <div className="topic-list-item">Pracs</div>
          <div className="topic-list-item">Q&A</div>
          <div className="topic-list-item">More</div>
          <div className="topic-list-item">Recommended</div>
        </div>
        <div className="topic-content">
          <div className="topic-name">{this.state.topicName}</div>

          {this.state.editTopic ? (
            <div className="topic-actual-content">
              <Editor
                editorState={editorState}
                wrapperClassName="demo-wrapper"
                editorClassName="demo-editor"
                onEditorStateChange={this.onEditorStateChange}
              />

              <textarea
                name="textarea"
                ref="_textarea"
                disabled
                value={draftToHtml(
```



```
        convertToRaw(editorState.getCurrentContent())
      )}
      style={{ display: "none" }}
    />
    <button onClick={this.submitWysiwyg}>Publish</button>
  </div>
) : (
  <div className="topic-actual-content">
    {this.state.topicData ? (
      <div id="topicContentLoaded"> </div>
    ) : (
      "Enter Data"
    )}
  </div>
)}
<div className="topic-action">
  <div className="topic-action-single">
    <FontAwesomeIcon icon={faHeart} />
    <p>Like</p>
  </div>
  <div className="topic-action-single">
    <FontAwesomeIcon icon={faCommentAlt} />
    <p>Comment</p>
  </div>
  <div className="topic-action-single">
```



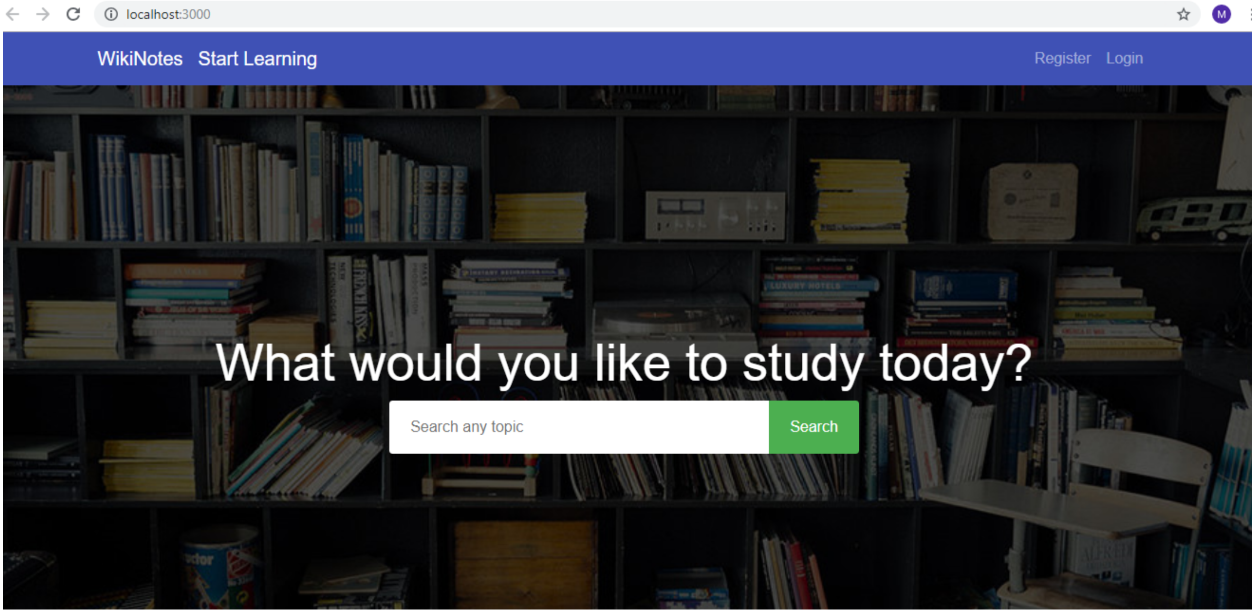
```
                           <FontAwesomeIcon icon={faEdit} />
               <p>Upload</p>
             </div>
           </div>
         </div>
       </div>
     </div>
   );
 }
}
const mapStateToProps = state => ({
 subjectID: state.checknotes.subjectID,
 moduleID: state.checknotes.moduleID,
 topicID: state.checknotes.topicID
});

export default connect(
 mapStateToProps,
 { select_subject, select_module, select_topic }
)(TopicMain);
```

Chapter 7

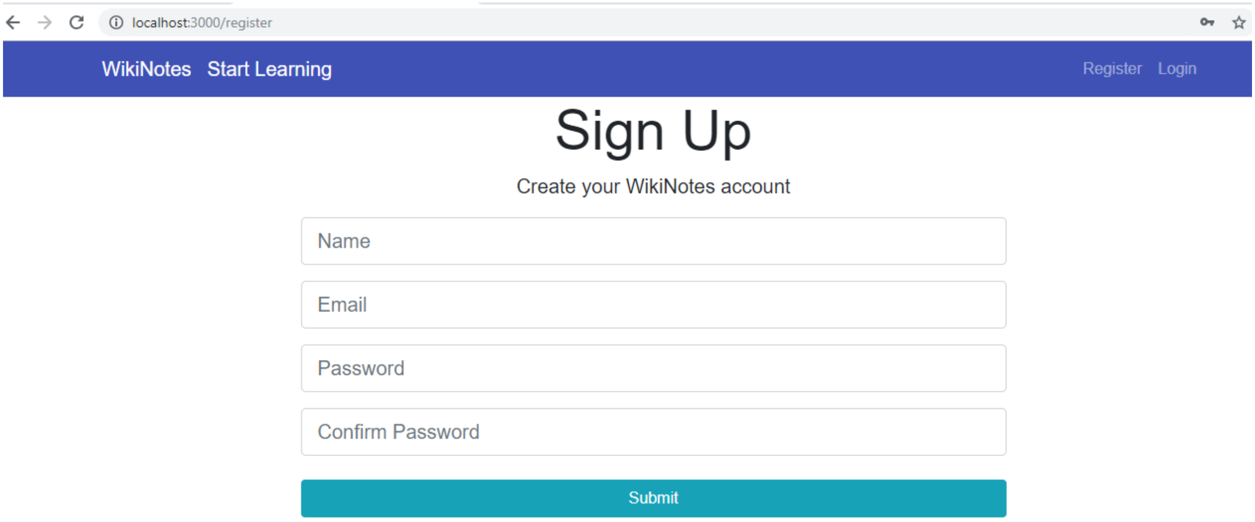
Result

The result is the actual implementation of our project.The working is as follows:

Landing Page



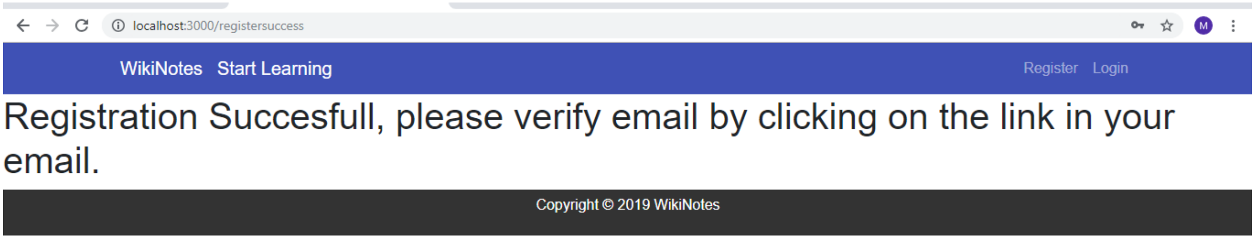
Register Page



Sign Up

Create your WikiNotes account

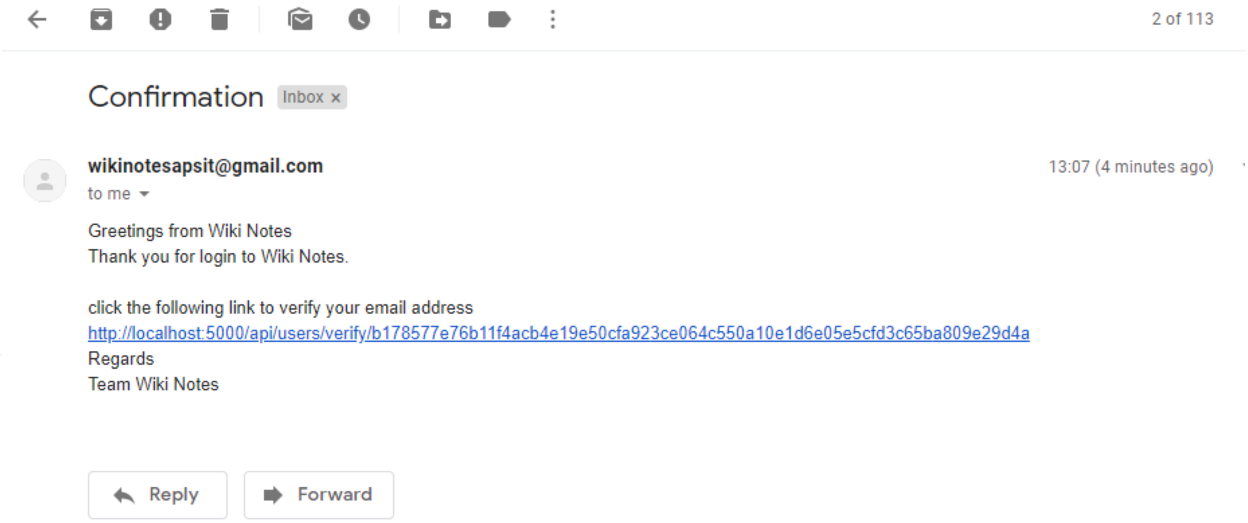
Register success



Registration Succesfull, please verify email by clicking on the link in your email.

Copyright © 2019 WikiNotes

Confirmation in Gmail



Confirmation

wikinotesapsit@gmail.com

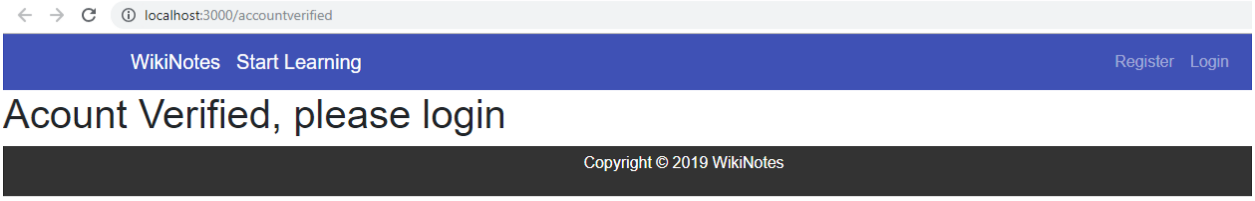
to me

13:07 (4 minutes ago)

Greetings from Wiki Notes
Thank you for login to Wiki Notes.

click the following link to verify your email address
http://localhost:5000/api/users/verify/b178577e76b11f4acb4e19e50cfa923ce064c550a10e1d6e05e5cfd3c65ba809e29d4a
Regards
Team Wiki Notes

Account verification confirmed

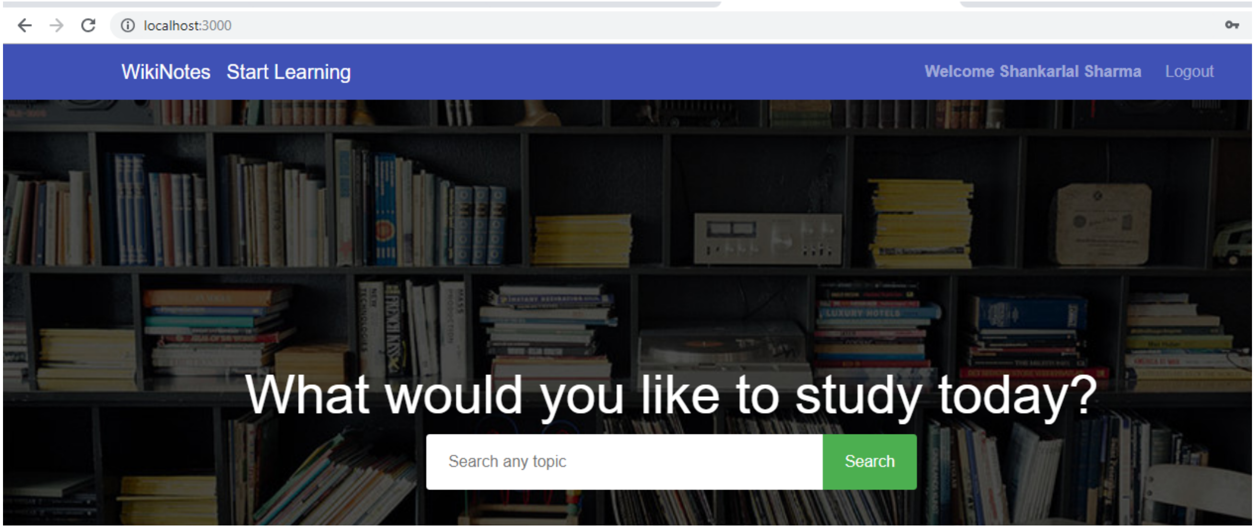


localhost:3000/accountverified

Acount Verified, please login

Copyright © 2019 WikiNotes

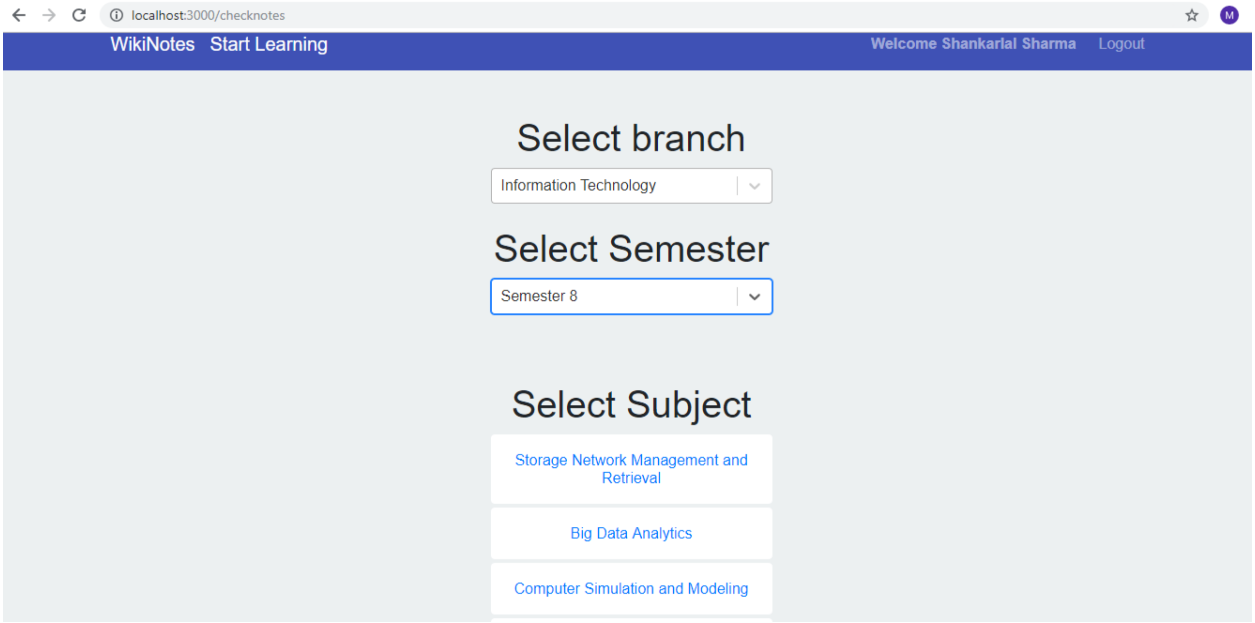
Login page success



localhost:3000

What would you like to study today?

Check notes



localhost:3000/checknotes

Select branch

Information Technology

Select Semester

Semester 8

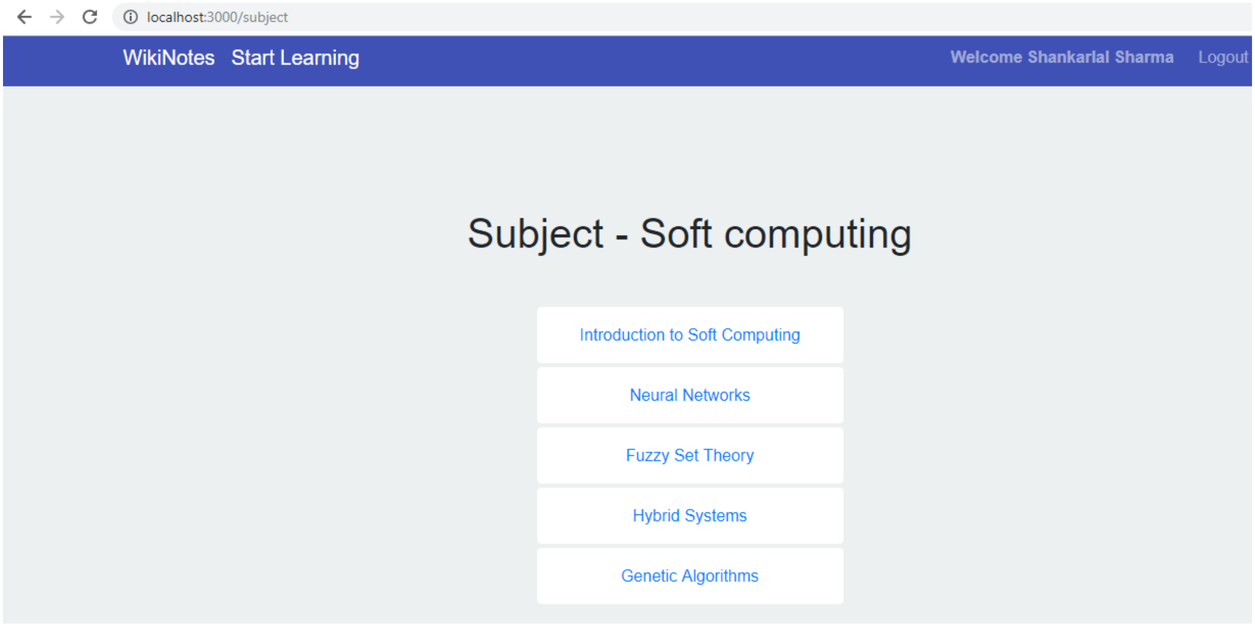
Select Subject

Storage Network Management and Retrieval

Big Data Analytics

Computer Simulation and Modeling

Select Subject



localhost:3000/subject

WikiNotes Start Learning

Welcome Shankarlal Sharma Logout

Subject - Soft computing

Introduction to Soft Computing

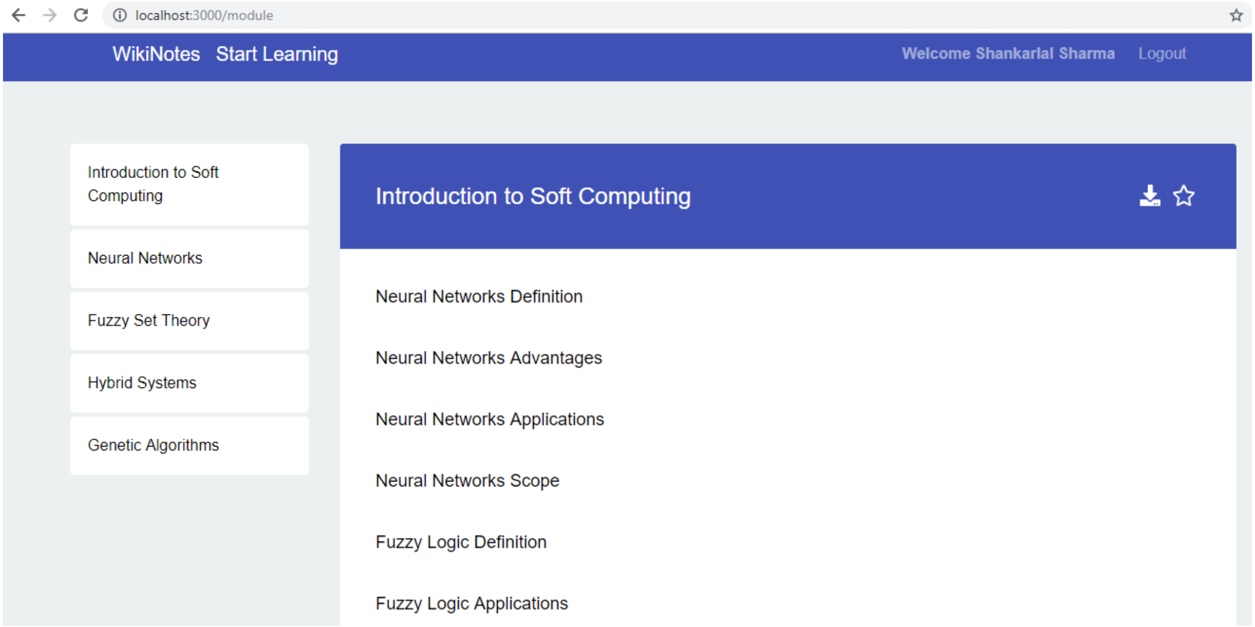
Neural Networks

Fuzzy Set Theory

Hybrid Systems

Genetic Algorithms

Modules page



Introduction to Soft Computing

Neural Networks Definition

Neural Networks Advantages

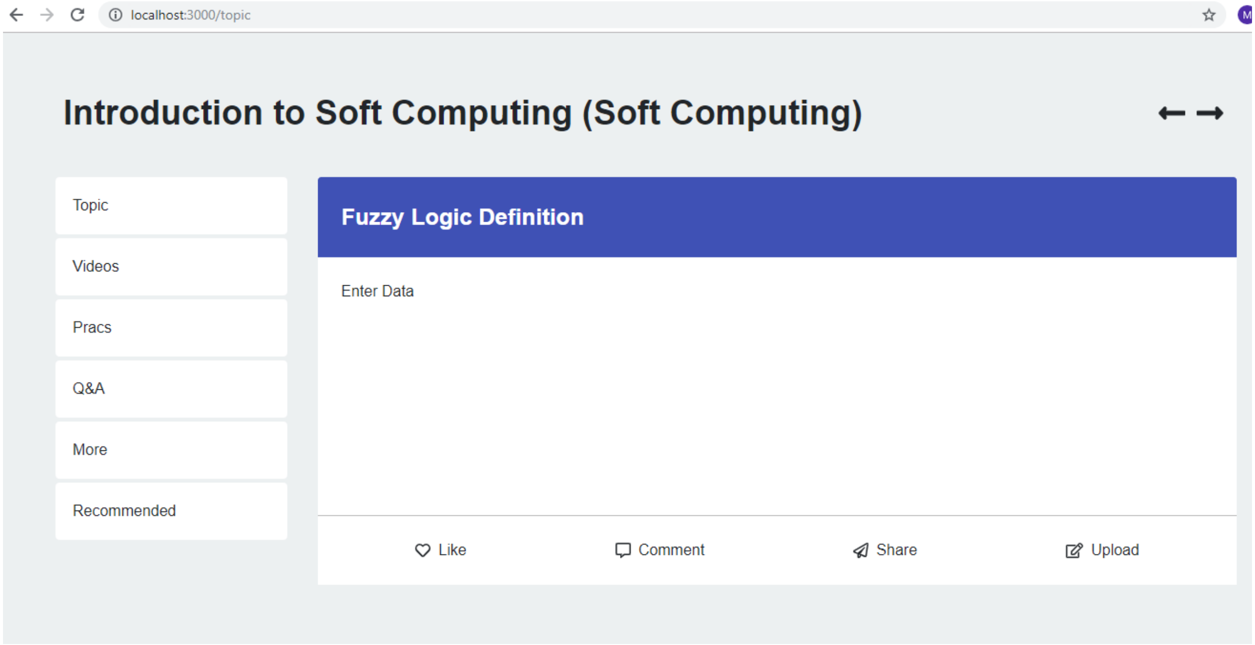
Neural Networks Applications

Neural Networks Scope

Fuzzy Logic Definition

Fuzzy Logic Applications

Topics Page

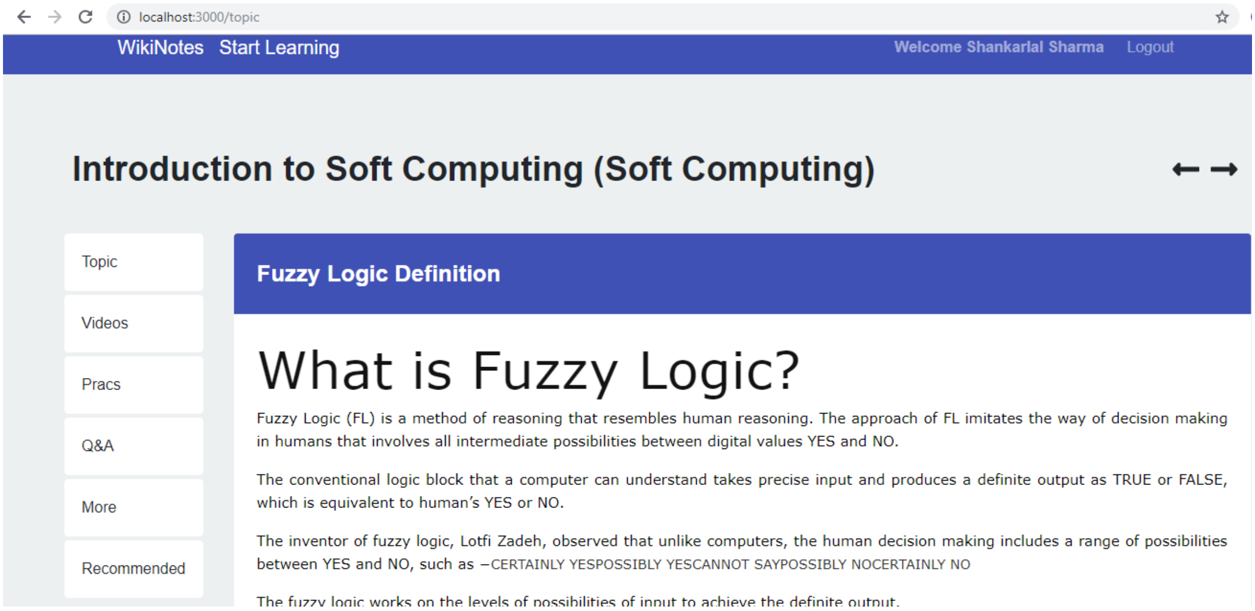


Introduction to Soft Computing (Soft Computing)

Fuzzy Logic Definition

Enter Data

Topic Uploaded on the page



Introduction to Soft Computing (Soft Computing)

Fuzzy Logic Definition

What is Fuzzy Logic?

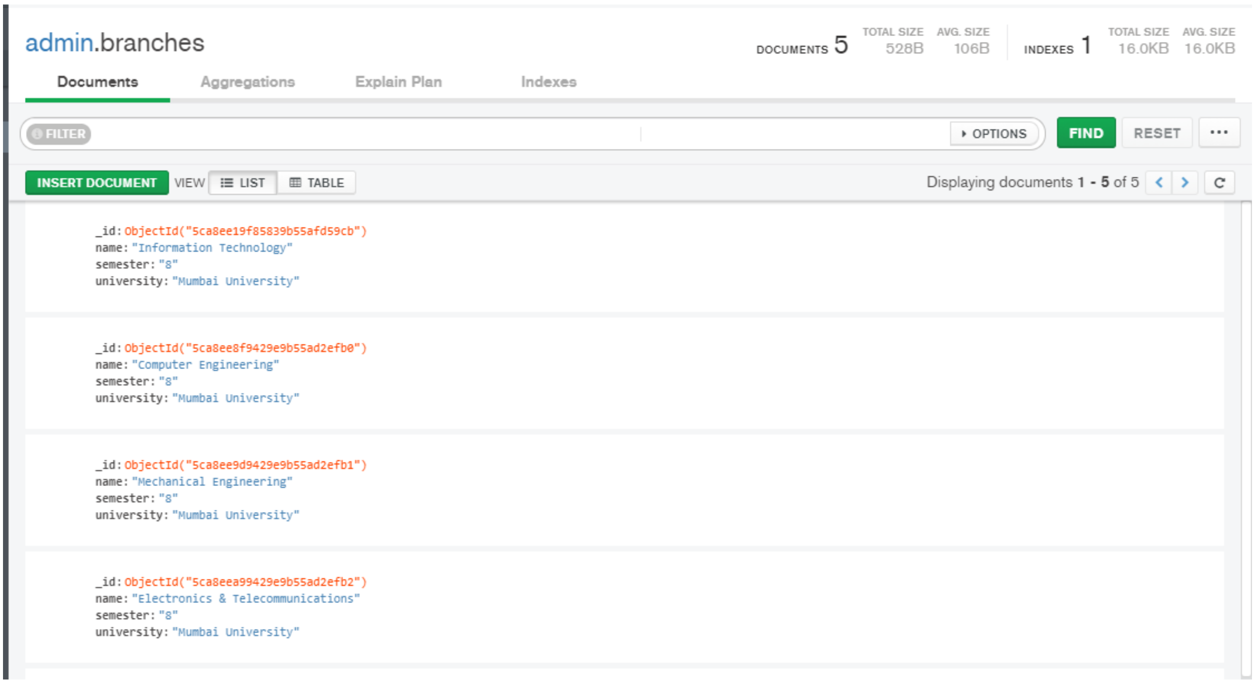
Fuzzy Logic (FL) is a method of reasoning that resembles human reasoning. The approach of FL imitates the way of decision making in humans that involves all intermediate possibilities between digital values YES and NO.

The conventional logic block that a computer can understand takes precise input and produces a definite output as TRUE or FALSE, which is equivalent to human's YES or NO.

The inventor of fuzzy logic, Lotfi Zadeh, observed that unlike computers, the human decision making includes a range of possibilities between YES and NO, such as −CERTAINLY YESPOSSIBLY YESCANNOT SAYPOSSIBLY NOCERTAINLY NO

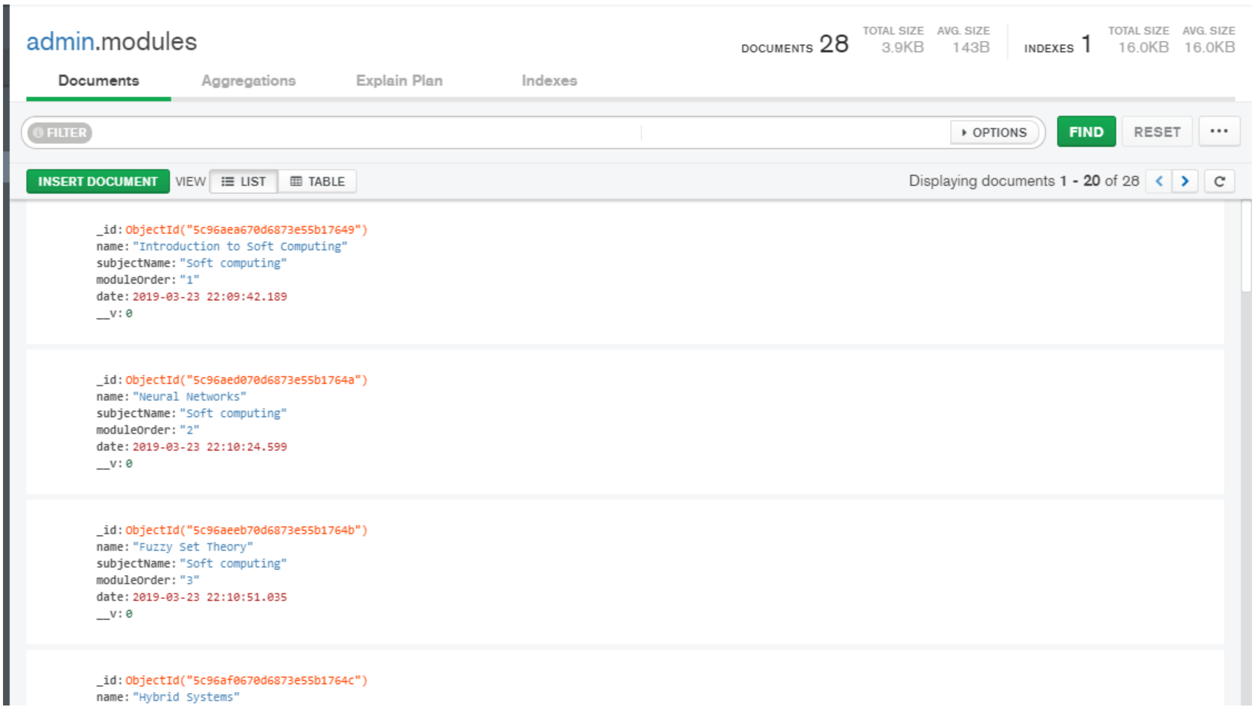
The fuzzy logic works on the levels of possibilities of input to achieve the definite output.

Database entry of Branches



```
_id: ObjectId("5ca8ee19f85839b55afd59cb")
name: "Information Technology"
semester: "8"
university: "Mumbai University"

_id: ObjectId("5ca8ee8f9429e9b55ad2efb0")
name: "Computer Engineering"
semester: "8"
university: "Mumbai University"

_id: ObjectId("5ca8ee9d9429e9b55ad2efb1")
name: "Mechanical Engineering"
semester: "8"
university: "Mumbai University"

_id: ObjectId("5ca8eea99429e9b55ad2efb2")
name: "Electronics & Telecommunications"
semester: "8"
university: "Mumbai University"
```

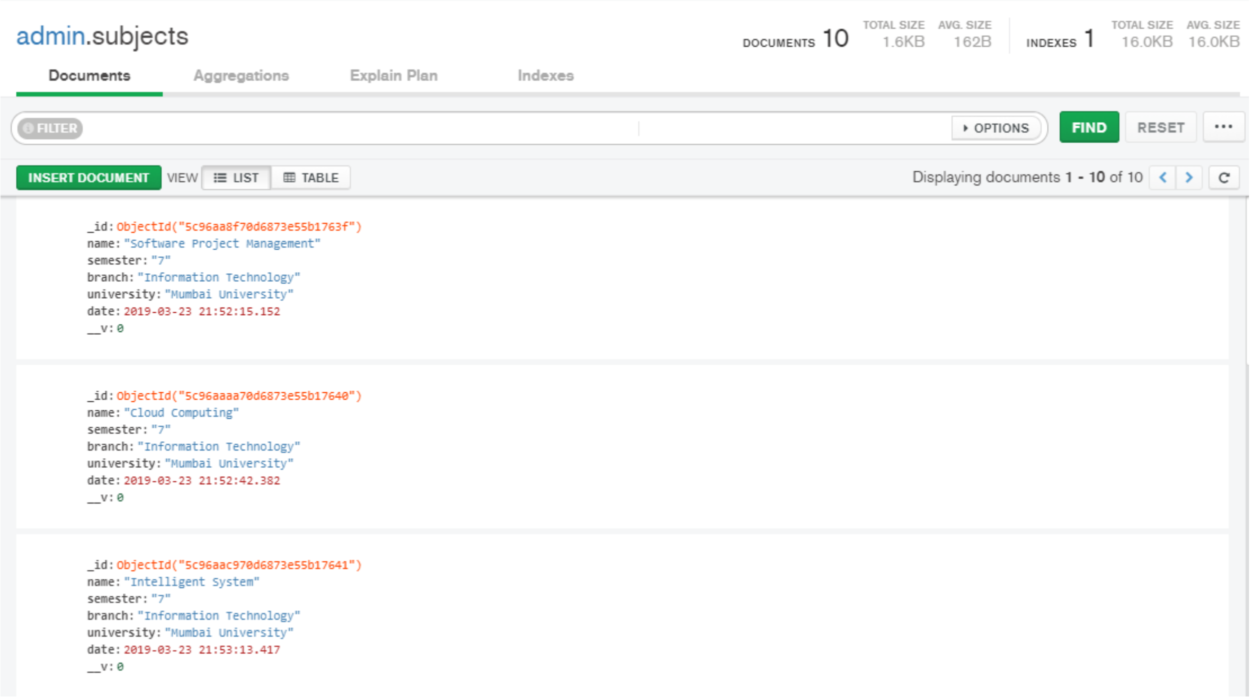
Database entry of Modules



Displaying documents 1 - 20 of 28

```
_id: ObjectId("5c96aea670d6873e55b17649")
name: "Introduction to Soft Computing"
subjectName: "Soft computing"
moduleOrder: "1"
date: 2019-03-23 22:09:42.189
__v: 0

_id: ObjectId("5c96aed070d6873e55b1764a")
name: "Neural Networks"
subjectName: "Soft computing"
moduleOrder: "2"
date: 2019-03-23 22:10:24.599
__v: 0

_id: ObjectId("5c96aeeb70d6873e55b1764b")
name: "Fuzzy Set Theory"
subjectName: "Soft computing"
moduleOrder: "3"
date: 2019-03-23 22:10:51.035
__v: 0

_id: ObjectId("5c96af0670d6873e55b1764c")
name: "Hybrid Systems"
```

Database entry of Subjects

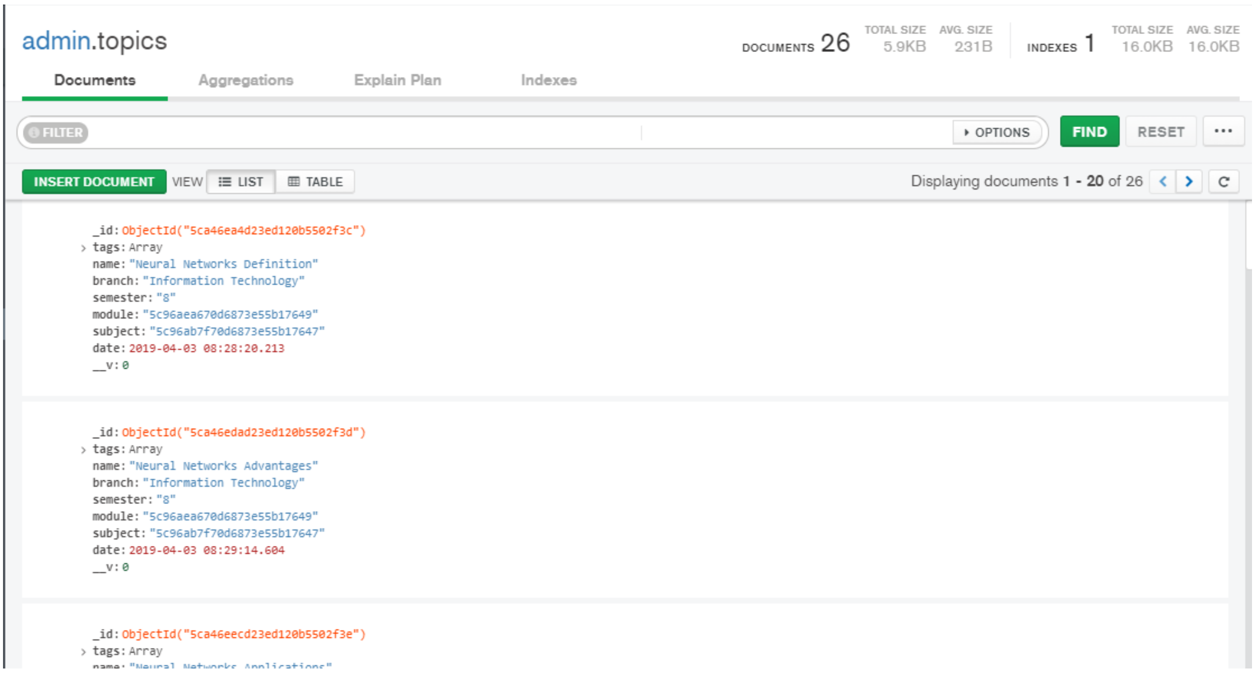


```
_id: ObjectId("5c96aa8f70d6873e55b1763f")
name: "Software Project Management"
semester: "7"
branch: "Information Technology"
university: "Mumbai University"
date: 2019-03-23 21:52:15.152
__v: 0
```

```
_id: ObjectId("5c96aaaa70d6873e55b17640")
name: "Cloud Computing"
semester: "7"
branch: "Information Technology"
university: "Mumbai University"
date: 2019-03-23 21:52:42.382
__v: 0
```

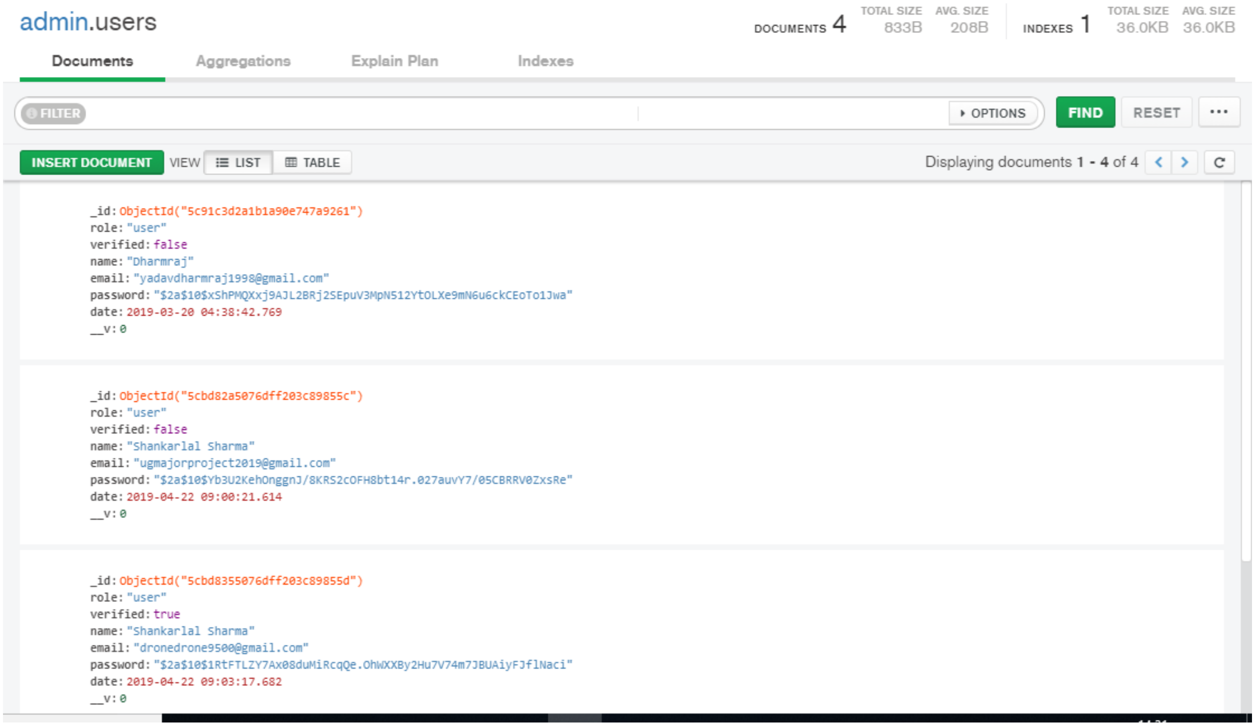
```
_id: ObjectId("5c96aac970d6873e55b17641")
name: "Intelligent System"
semester: "7"
branch: "Information Technology"
university: "Mumbai University"
date: 2019-03-23 21:53:13.417
__v: 0
```

Database entry of Topic



```
_id: ObjectId("5ca46ea4d23ed120b5502f3c")
> tags: Array
  name: "Neural Networks Definition"
  branch: "Information Technology"
  semester: "8"
  module: "5c96aea670d6873e55b17649"
  subject: "5c96ab7f70d6873e55b17647"
  date: 2019-04-03 08:28:20.213
  __v: 0

  _id: ObjectId("5ca46edad23ed120b5502f3d")
> tags: Array
  name: "Neural Networks Advantages"
  branch: "Information Technology"
  semester: "8"
  module: "5c96aea670d6873e55b17649"
  subject: "5c96ab7f70d6873e55b17647"
  date: 2019-04-03 08:29:14.604
  __v: 0

  _id: ObjectId("5ca46eecd23ed120b5502f3e")
> tags: Array
  name: "Neural Networks Applications"
```

Database entry of Users



```
_id: ObjectId("5c91c3d2a1b1a90e747a9261")
role: "user"
verified: false
name: "Dharmraj"
email: "yadavdharmraj1998@gmail.com"
password: "$2a$10$xShPMQXxj9AJL2BRj2SEpuV3MpN512YtOLXe9mN6u6ckCEoTo1Jwa"
date: 2019-03-20 04:38:42.769
__v: 0

_id: ObjectId("5cbd82a5076dff203c89855c")
role: "user"
verified: false
name: "Shankarlal Sharma"
email: "ugmajorproject2019@gmail.com"
password: "$2a$10$Yb3U2KehOnggnJ/8KRS2cOFH8bt14r.027auvY7/05CBRRV0ZxsRe"
date: 2019-04-22 09:00:21.614
__v: 0

_id: ObjectId("5cbd8355076dff203c89855d")
role: "user"
verified: true
name: "Shankarlal Sharma"
email: "dronedrone9500@gmail.com"
password: "$2a$10$1RtFTLZY7Ax08duMiRcqQe.OhWXXBy2Hu7V74m7JBUAiyFJflNaci"
date: 2019-04-22 09:03:17.682
__v: 0
```

Chapter 8

Conclusions and Future Scope

8.1 Conclusion

In this blackbook, the importance of Quality study material for a student is discussed and the methods by which this can be achieved. The solution is provided by means of building a website where students of a particular university will be able to collaborate on making a rich source of study material and the quality of which can be ensured by Professors. However, According to the research, it is also important to provide a rich user interface and quality user experience, otherwise the material will lose their value users will not be able to easily access them. Also, along with study materials, features like rich-text, automatic question generation, dictionary can be implemented.

8.2 Future Scope

- An Android and IOS app can be developed, so that the user will be able to access the notes from anywhere and anytime.
- Automatically adding tags to notes for quicker searches
- A Chat bot can be added that will assist common queries by user.
- The website can be gamified, where users earn points for uploading the notes, and also when their notes get liked by others userrs.
- The question generation and recommending system algorithm can be further be made more accurate and efficient.

Bibliography

[1] Blanka Frydrychova Klimova, Petra Poulova, Michal Slama, eLeaming study materials and students preferences , 2014 Information Technology Based Higher Education and Training(ITHET) , 2014

[2] Ivana Simonova, Petra Poulava, Study Materials in Online Courses, Analysis Reflecting Individual Learning Styles, 2014 IEEE Global Engineering Education Conference(EDUCON), 2014

[3] Sucheta Kolekar, Radhika M. Pai, Manohara Pai M.M., Adaptive User Interface for Elearning Applications based on Learning Styles using Web Logs Analysis: A Hybrid Cloud Architecture,TENCON 2015 - 2015 IEEE Region 10 Conference, 2015

[4] Rolysent K. Paredes, Alexander A. Hernandez, Measuring the Quality of User Experience on Web Services: A Case of University in the Philippines, 2017 IEEE 9th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM), 2017

[5] Mahesh S. Patil, Meenaxi M Raikar, Padmashree Desai, Vijayalakshmi M, Shivalingappa Battur, Parikshit H, G.H Joshi,Leveraging Student Project through MOOC on UX:Case Study 2016 IEEE 4th International Conference on MOOCs, Innovation and Technology in Education (MITE)

[6] Dr.P Pabitha, M.Mohana, S.Suganthi, B.Sivanandhini, ”Automatic Question Generation System”, 2014 International Conference on Recent Trends in Information Technology

[7] Sriramya, P Karthika, R.A.. (2015). ”Providing password security by salted passwordhashing using Bcrypt algorithm.” ARPN Journal of Engineering and Applied Sciences. 10.5551-5556

[8] Kumar N., Chaudhary P. (2018) ”Password Security Using Bcrypt with AES EncryptionAlgorithm.” In: Satapathy S., Bhateja V., Das S. (eds)

Smart Computing and Informatics.Smart Innovation, Systems and Technologies, vol 77. Springer, Singapore

[9] F. Wiemer and R. Zimmermann, High-speed implementation of bcrypt passwordsearch using special-purpose hardware, 2014 International Conference on ReConFigurableComputing and FPGAs (ReConFig14), Cancun, 2014, pp. 1-6

[10] C. Gyrdi, R. Gyrdi, G. Pecherle and A. Olah, A comparative study: MongoDB vs.MySQL, 2015 13th International Conference on Engineering of Modern Electric System (EMES), Oradea, 2015, pp. 1-6.

[11] M. H. Mohamed, M. H. Khafagy and M. H. Ibrahim,**Recommender Systems Chal-lenges and Solutions Survey,** 2019 International Conference on Innovative Trends in Com-puter Engineering (ITCE), Aswan, Egypt, 2019, pp. 149-155.

[12] Martins P., Abbasi M., S F. (2019) "A Study over NoSQL Performance." In: Rocha., Adeli H., Reis L., Costanzo S. (eds) New Knowledge in Information Systems and Tech-nologies. WorldCIST19 2019.

Appendices

Appendix-A: Node.js and NPM Download and Installation

Node.js is a Javascript run-time environment that allow us to execute Javascript code like if we were working on a server. Remember that every web application is meant to be executed in a server (or a local server, if were running it in our computers). In the other hand NPM is a package manager for Javascript, that is, NPM allows us to install Javascript libraries to make our experience even more richer by expanding the basic functionalities.

1. Go to the site **https://nodejs.org/en/download/** and download the necessary binary files. In our example, we are going to the download the 32-bit setup files for Node.js.

2. Double click on the downloaded **.msi** file to start the installation. Click the Run button in the first screen to begin the installation.

3. In the next screen, click the "Next" button to continue with the installation

4. In the next screen Accept the license agreement and click on the Next button.
5. In the next screen, choose the location where Node.js needs to be installed and then click on the Next button.

- 1.First enter the file location for the installation of Node.js. This is where the files for Node.js will be stored after the installation.
- 2.Click on the Next button to proceed ahead with the installation.

6. Accept the default components and click on the next button.

7. In the next screen, click the Install button to start the installation.

8. Click the Finish button to complete the installation.

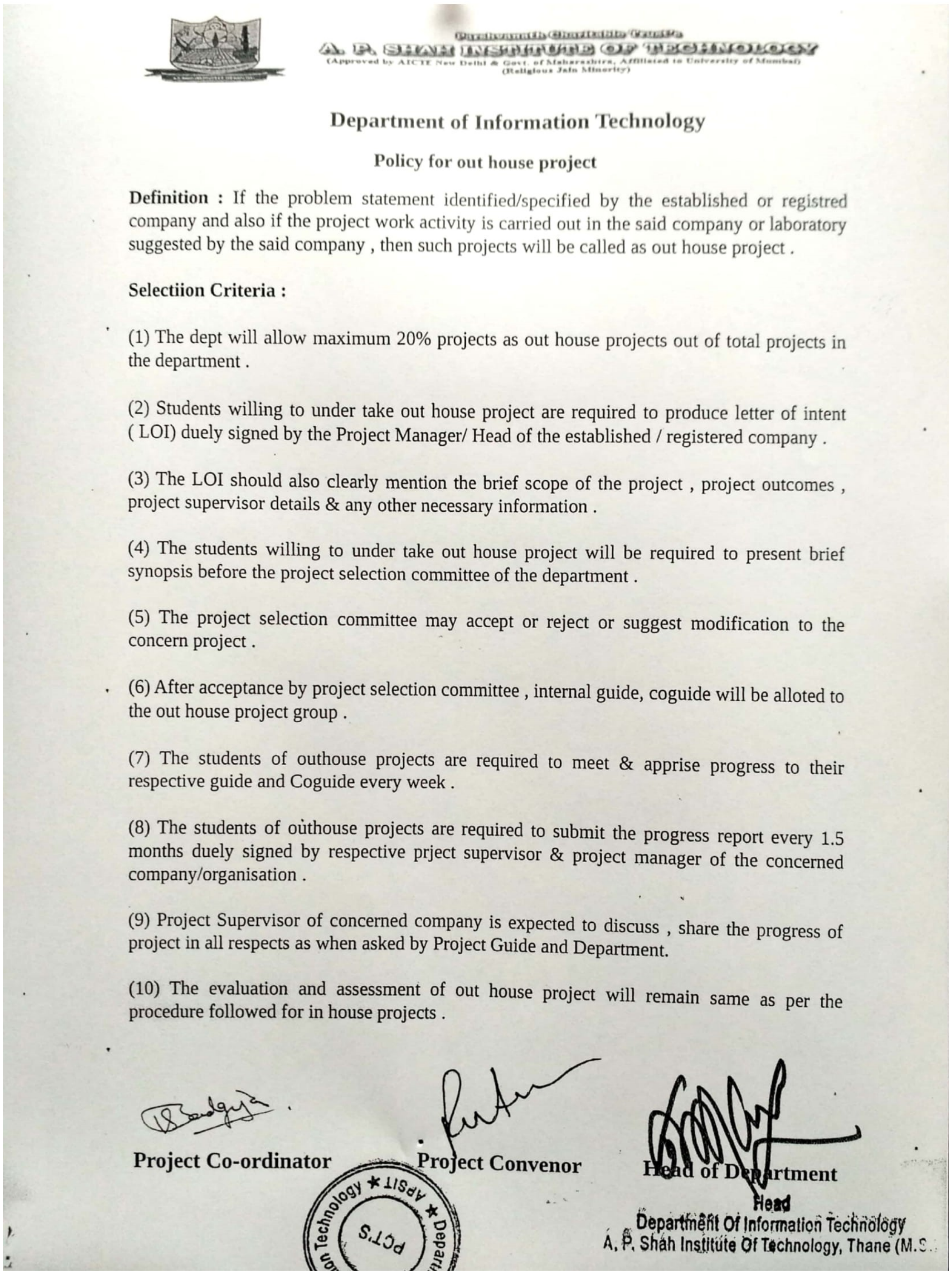
9. The next step is to install Node.js to your local machine using the package manager. This can be done by running the below command in the command prompt. **cinst nodejs install**
If the installation is successful, you will get the message of the successful installation of Node.js.

Publication

[1] Paper entitled **"An Integrated Platform for Knowledge Sharing"** is presented at **"International Conference on Advances in Science, Technology and Engineering (ICASTe-2019)"** by **"Shankarlal Sharma" (15104018), "Dharmraj Yadav" (15104021), "Satyajeet Yadav" (15104025), "Gaurav Babar" (15104033).**

[2] Paper entitled **"Secure Website Authentication using Hashing Algorithms"** is published at **"International Journal for Scientific Research and Development"** by **"Shankarlal Sharma" (15104018), "Dharmraj Yadav" (15104021), "Satyajeet Yadav" (15104025), "Gaurav Babar" (15104033).**

Policy for Out house Project



Parshvanath Charitable Trust's

A. P. SHAH INSTITUTE OF TECHNOLOGY

(Approved by AICTE New Delhi & Govt. of Maharashtra, Affiliated to University of Mumbai)
(Religious Jain Minority)

Department of Information Technology

Policy for out house project

Definition : If the problem statement identified/specified by the established or registred company and also if the project work activity is carried out in the said company or laboratory suggested by the said company , then such projects will be called as out house project .

Selectiion Criteria :

(1) The dept will allow maximum 20% projects as out house projects out of total projects in the department .

(2) Students willing to under take out house project are required to produce letter of intent (LOI) duely signed by the Project Manager/ Head of the established / registered company .

(3) The LOI should also clearly mention the brief scope of the project , project outcomes , project supervisor details & any other necessary information .

(4) The students willing to under take out house project will be required to present brief synopsis before the project selection committee of the department .

(5) The project selection committee may accept or reject or suggest modification to the concern project .

(6) After acceptance by project selection committee , internal guide, coguide will be alloted to the out house project group .

(7) The students of outhouse projects are required to meet & apprise progress to their respective guide and Coguide every week .

(8) The students of outhouse projects are required to submit the progress report every 1.5 months duely signed by respective prject supervisor & project manager of the concerned company/organisation .

(9) Project Supervisor of concerned company is expected to discuss , share the progress of project in all respects as when asked by Project Guide and Department.

(10) The evaluation and assessment of out house project will remain same as per the procedure followed for in house projects .

Project Co-ordinator

Project Convenor

Head of Department

APSIT ★ ... Technology ★ Depart...
PCT's

Head
Department Of Information Technology
A. P. Shah Institute Of Technology, Thane (M.S.

Figure 8.1: Project Policy