RV COLLEGE OF ENGINEERING **®**

BENGALURU-560059

(Autonomous Institution Affiliated to VTU, Belagavi)



“Reddit Clone”

# Report

**Database Management Systems Laboratory Project (16IS64)**

*Submitted By*

Saksham Garg (1RV17IS035) Rohit M (1RV17IS032)

# Under the Guidance of

**Dr.Padmashree T Smitha G.R Assistant Prof.,Dept. of ISE Assistant Prof., Dept of ISE**

***in partial fulfillment for the award of degree of***

***Bachelor of Engineering***

***in***

**INFORMATION SCIENCE AND ENGINEERING**

## Jan-May 2020



**CERTIFICATE**

Certified that the Mini Project work entitled ‘Reddit Clone’ has been carried out as a part of Database Management Systems Laboratory(16IS64) in partial fulfillment for the award of degree of **Bachelor of Engineering** in **Information Science and Engineering** of the Visvesvaraya Technological University, Belagavi during the year **2019-2020** by **Saksham Garg (1RV17IS035), Rohit M (1RV17IS032),** who are bonafide students of **RV College of Engineering®**, Bengaluru. It is certified that all the corrections/suggestions indicated for the internal assessment have been incorporated in the report deposited in the departmental library. The report has been approved as it satisfies the academic requirements in respect of work prescribed by the institution for the said degree.

|  |  |  |
| --- | --- | --- |
| **Dr.Padmashree T**  **Assistant Prof** | **Smitha G.R**  **Assistant Prof** | **Dr Sagar B M**  **Head of the Department** |
| Department of ISE, | Department of ISE, | Department of ISE, |
| RVCE, Bengaluru-59 | RVCE, Bengaluru-59 | RVCE, Bengaluru-59 |
| Name of the Examiners  1. |  | Signature with Date |
| 2. |  |  |

# DECLARATION

We **Saksham Garg, Rohit M,** are students of Eighth Semester B.E Department of Information Science and Engineering, **RV College of Engineering ®,** bearing **USN: 1RV17IS035, 1RV17IS032,** hereby declare that the project titled “***Reddit Clone***” has been carried out as a part of Database Management Systems Laboratory(16IS64) by us and submitted in partial fulfillment of the program requirements for the award of degree in Bachelor of Engineering in Information Science and Engineering of the **Visvesvaraya Technological University, Belagavi** during the year **2019-2020.**

Further we declare that the content of the report has not been submitted previously by anybody for the award of any degree or diploma to any other University.

|  |  |  |
| --- | --- | --- |
| **Place: Bengaluru** | **Name** | **Signature with Date** |
| **Date:** | **Saksham Garg** |  |
|  | **Rohit M** |  |

# TABLE OF CONTENTS

## CHAPTER 1 1

INTRODUCTION 1

* 1. Terminology 2
  2. Purpose 2
  3. Motivation 2
  4. Problem Statement 2
  5. Objective 3
  6. Scope and Relevance

## CHAPTER 2 12

REQUIREMENT SPECIFICATION 12

* 1. Specific Requirements 13
     1. Functional Requirements 13
     2. Non-Functional Requirements 13
     3. Hardware Requirements 14
     4. Software Requirements 14

## CHAPTER 3 15

DESIGN

* 1. E-R Diagram
     1. Schema Representation 15
  2. UML Diagrams
     1. Class Diagram
     2. Sequence Diagram
     3. Activity Diagram
  3. Front End Design

CHAPTER 4 26

IMPLEMENTATION DETAILS 26

* 1. Database implementation 26
     1. Table Creation 26
     2. [Table Population 27](#_TOC_250009)
     3. Query Execution and Output
     4. Security features
  2. [Front End implementation 28](#_TOC_250008)
     1. [Form Creation 28](#_TOC_250007)
     2. Connectivity to the Database 28
     3. Report generation 28
     4. Security features

CHAPTER 5 29

TESTING AND RESULTS 29

* 1. [Database Testing](#_TOC_250006)
     1. [Test cases 29](#_TOC_250005)
  2. [Front End Testing 30](#_TOC_250004)
     1. Test cases 32
  3. [System Testing 33](#_TOC_250003)
     1. [Test cases 33](#_TOC_250002)

CHAPTER 6 38

CONCLUSION 38

* 1. [Limitations 38](#_TOC_250001)
  2. [Future Enhancements 39](#_TOC_250000)

REFERENCES 40

APPENDIX A- CODE SNIPPETS 45

APPENDIX B –SCREENSHOTS 59

# CHAPTER 1 INTRODUCTION

Reddit is a social news platform that allows users to discuss and vote on content that other users have submitted. To help police the site and prevent spammers from bombarding readers, Reddit came up with “karma” points. Users get karma by their comments and links being up-voted by others in the community. Reddit is broken up into more than a million communities known as “subreddits,” each of which covers a different topic. The name of a subreddit begins with /r/, which is part of the URLs that Reddit uses. For example, /r/nba is a subreddit where people talk about the National

Basketball Association, while /r/boardgames is a subreddit for people to discuss board games. Those are straightforward subreddits, but they can get weird, such as

/r/birdswitharms, a subreddit devoted to pictures of birds…with arms. If you’re just

looking at Reddit for the first time, you may be a bit confused by what you are seeing, so here’s a quick rundown. The homepage (or “front page”) shows you various posts that are currently trending on the site, pulled from a variety of subreddits. You can click the sort button in the upper left to sort by posts that are “hot” (trending), new, controversial, and more.

## Terminology:

* + - **JavaScript:** A high level programming language used to make the HTML pages responsive using functions, multi-paradigm. It is object-oriented and prototype based.
    - **SQL:** (Structured Query Language) A domain specific language used in programming and designed for managing data and to communicate with the database.
* **Reddit:** Reddit is a social news platform that allows users to discuss and vote on content that other users have submitted.
* **Sub\_Reddit:** A forum dedicated to a specific topic on the website Reddit.

## Purpose

Reddit contains the collective knowledge of thousands of users, much like Quora. For hobbies or studies, it provides you with a place to ask questions (like Quora), and get meaningful and helpful answers across a huge range of topics.

Think of Reddit as your extended circle of friends. There’s people in your friend group that cover knowledge across every occupation. The majority of these people are helpful, some are not. Just like having bad friends in real life.

Reddit is a Knowledge transfer station, as well as a place for many people to relax and share experiences. From discussion tough times about family members with Cancer, to sharing Keto weight loss pics, to discussing TinyRails (the mobile android Train game), to sharing cute, offensive, funny, disgusting videos and ground-breaking news.

## Motivation

The main motivation behind making a reddit clone was to understand the in depth working of a major social media site, we also wanted to learn how to use newer

technologies like React, Node and Express. The best part of making the clone was that it could be used as a platform among the students and faculty within college to interact and collaborate with each other.

## Problem Statement:

To build a clone of the reddit app, where users can interact with each other to collaborate and discuss on various topics related to any topic.

# CHAPTER 2 REQUIREMENT SPECIFICATION

This chapter explains in detail about the requirements needed to run and understand the functionalities of the project, it is divided into different sections.

## SPECIFIC REQUIREMENT

* + - Create User
    - Enable user to create POST
    - Enable user to create SUB\_REDDIT
    - Enable user to write COMMENTS
    - Enable user to read POST’S and SUB\_REDDIT

## Functional Requirements

In this clone of reddit, the aim is to create a close replica which can be used for college purposes. The first step is to have all users login through an email-id(rvce id), and set up a password and username, doing the above registers them as a user, and this user

is entered along with their details into the users database. A user can create a subreddit page, and can make a post of any kind on the page, these posts maybe a text, Image, video link etc. A user can also upvote, downvote and comment on any post, the number of upvotes and downvotes of a post decide the users “Karma”.

## Non-Functional Requirements

* + - * **Availability:** Since we have used Node.js and MySQL, the availability will be 24x7.
      * **Ease of Usability:** The frontend design implemented here is very intuitive and easy to use and doesn’t require any special skills and pre-requisites to use. The

menus used in the system are easily understandable for the convenience of the users. All relevant data should be portrayed in an intuitive manner.

* + - * **Performance:** The simple design at the frontend provides a reliable way to establish quick and efficient communication with the database at the backend thus improvising performance of the system.
      * **Reliability :** The system shall be recovered within 10 minutes if it is down. The system shall be recovered without intervention at user terminal if it is down . The system shall show appropriate messages at the terminal when the system is down. The system shall generate appropriate messages and errors when the user enters invalid data.

## Hardware Requirements

* + - * Intel Pentium 4 or more
      * RAM 1GB or more
* Hard disk of minimum partition of 40 GB

## Software Requirements

* + - * A database system (SQL 2005 or higher)
      * Text Editor (VS Code or Notepad++)
      * Chrome or Firefox Browser
      * MySql
      * express.js
      * React
      * node.js

## E-R DIAGRAM

**CHAPTER 3 DESIGN**

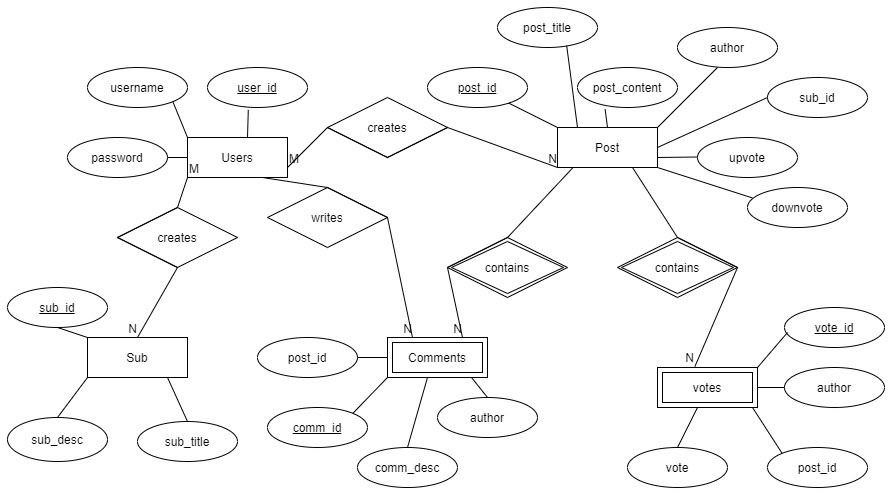
The Entity-Relationship diagram also known as ER diagram is a pictorial representation of the tables used in the Database and their attributes, and the relationship among the entities of the database with their attributes.

Fig 3.1 : E-R Diagram for Reddit Clone

## Explanation of E-R:

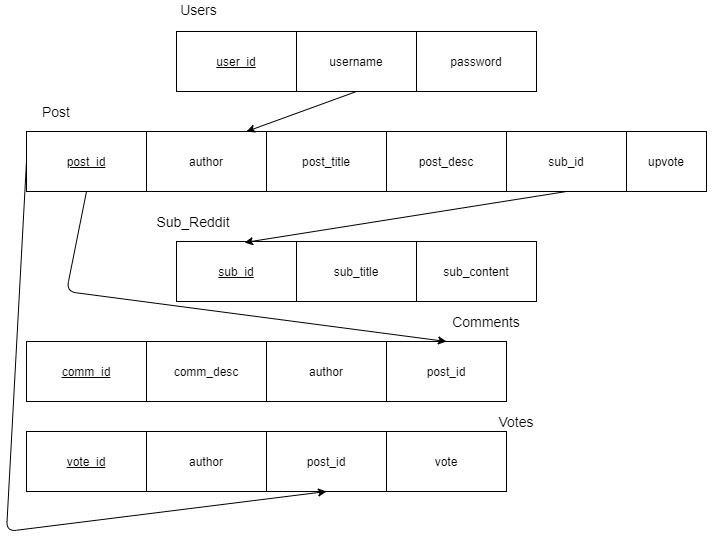
**Relationships**

* + - One or more user can create sub\_reddit.
    - One user can create N posts.
    - One user can write N comments.
    - One post contain N comments.
    - One post contain N votes.

## Operations

* + - User can create Post.
    - User can create Sub\_Reddit.
    - User writes comments.
    - Each sub\_reddit contain specific posts.
    - User can read all the posts.

## 3.1.1 Schema Diagram

****

* 1. **UML DIAGRAM**
     1. **Class Diagram**

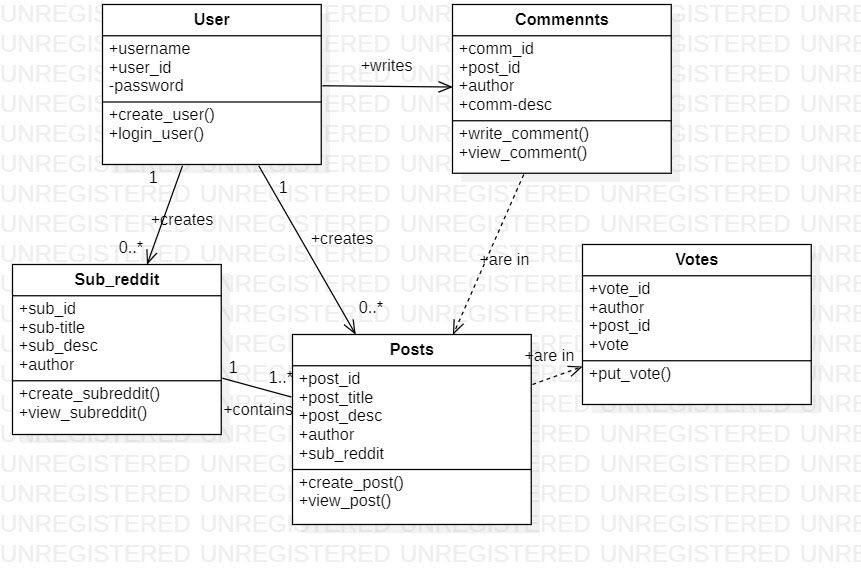
****

Fig 3.3 : Class diagram showing all the classes and attributes of the application

* + - * A class diagram lists all the attributes and operations of the modules of database architecture.
      * This shows that User can create sub\_reddit and posts.
      * Some of the operations or features are shared between two modules in order to establish a systematic relationship and to manage the architecture collectively.

## Activity Diagram

* + - * Activity diagram is the pictorial representation of the flow of the architecture when a certain action is performed.
      * This helps in debugging of the error source when an unexpected error occurs and the user is unable to locate it.
      * It gives an overall idea of the modification status of the application.

## Sequence Diagram

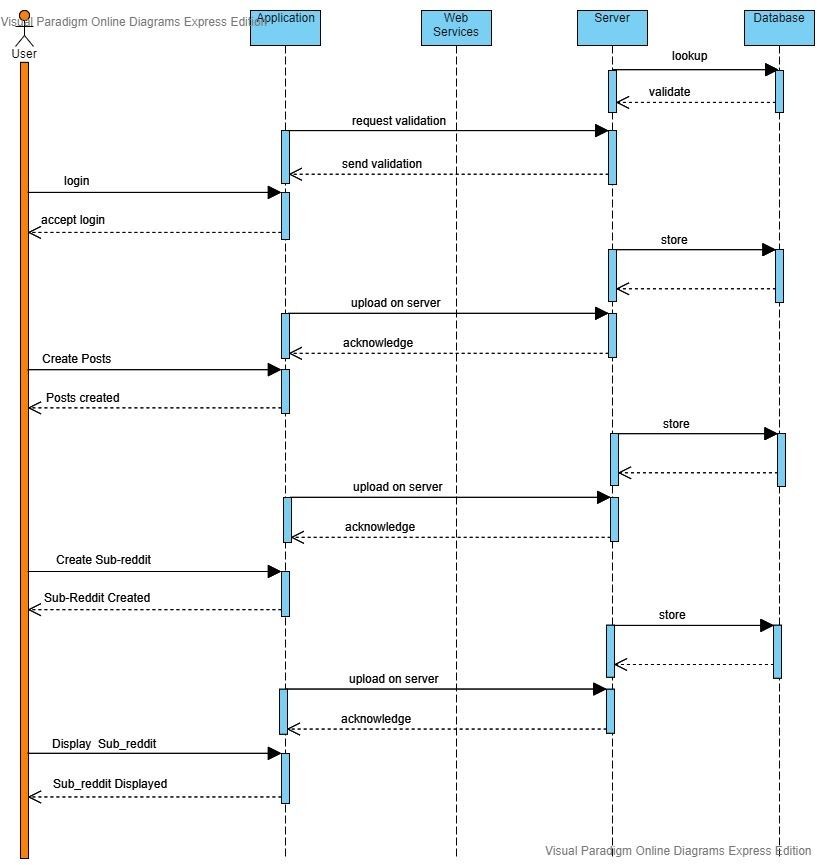
****

Fig 3.4 Sequence diagram showing the communications among the various application components and how the data is transferred and stored in the database, it is mainly used to explain the flow when a user logs in or logs out, and how login and logout affects the backend.

# CHAPTER 4 IMPLEMENTATION DETAILS

* 1. **Database Implementation 4.1.1Table Creation**

The tables created for this project are:

* + - User
    - Post
    - Sub\_Reddit
    - Comments
    - Votes 1.To define table User:

Fig 4.1: User Table Snapshot.

CREATE TABLE `users` (

`user\_id` int(11) NOT NULL AUTO\_INCREMENT,

`username` varchar(50) NOT NULL,

`password` varchar(50) NOT NULL, PRIMARY KEY (`user\_id`),

UNIQUE KEY `username` (`username`)

)

1. To define table Post:

`downvote` int(11) DEFAULT '0',

`sub\_id` int(11) DEFAULT NULL, PRIMARY KEY (`post\_id`),

KEY `author` (`author`), KEY `sub\_id` (`sub\_id`),

*CONSTRAINT* `post\_ibfk\_2` *FOREIGN KEY* (`sub\_id`) *REFERENCES* `sub` (`sub\_id`)

)

Fig 4.2 : Post Table Snapshot.

1. To create table Sub\_Reddit:

CREATE TABLE `sub` (

`sub\_id` int(11) NOT NULL AUTO\_INCREMENT,

`sub\_title` varchar(100) NOT NULL,

`sub\_desc` varchar(1000) NOT NULL, PRIMARY KEY (`sub\_id`),

UNIQUE KEY `sub\_title` (`sub\_title`)

)

Fig 4.3 : Sub\_Reddit Table Snapshot.

1. To create table comments:

Fig 4.4 : Comments Table Snapshot.

1. To create table votes:

CREATE TABLE `votes` (

`vote\_id` int(11) NOT NULL AUTO\_INCREMENT,

`author` varchar(50) DEFAULT NULL,

`post\_id` int(11) DEFAULT NULL,

`vote` int(11) DEFAULT '0', PRIMARY KEY (`vote\_id`),

UNIQUE KEY `author` (`author`), KEY `post\_id` (`post\_id`),

*CONSTRAINT* `votes\_ibfk\_2` *FOREIGN KEY* (`post\_id`) *REFERENCES* `post` (`post\_id`)

)

Fig 4.5 : Votes Table Snapshot.

# 4.1.2 Table Population

Table population in the project is done through the components mentioned in the application.These pages are rendered according to the components defined in the ‘src’ folder of client.

# Front End Implementation

# Form Creation

List of forms used in the project:

* + - * Sign Up
      * Login
      * Create Post
      * Write Comment
      * Create Sub\_Reddit

render(){

var head = "Sign Up";

return(

<div>

<Header *value* ={head}/>

<form *className*="login-box" >

<div >

<label *htmlFor*="username">Username</label>

<input *className*="form-control" *type*="text" *name*="username" *value* = {*this*.state.Username} *onChange*={*this*.handleuser}/>

</div>

<div>

<label *htmlFor* = "password">Password</label>

<input *className*="form-control" *type*="text" *name*="password" *value* = {*this*.state.Password} *onChange*={*this*.handlepass}/>

</div>

<button *type*="submit" *onClick* ={*this*.onSubmit}>Submit</button>

</form>

</div>

);

}

}

Fig 4.15 : Signup Form Snapshot for registering a new user.

<Header *value* ={head}/>

<form *className*="login-box" >

<div >

<label *htmlFor*="username">Username</label>

<input *className*="form-control" *type*="text" *name*="username" *value* = {*this*.state.Username} *onChange*={*this*.handleuser}/>

</div>

<div>

<label *htmlFor* = "password">Password</label>

<input *className*="form-control" *type*="text" *name*="password" *value* = {*this*.state.Password} *onChange*={*this*.handlepass}/>

</div>

<button *type*="submit" *onClick* ={*this*.onSubmit}>Submit</button>

</form>

</div>

);

}

}

Fig 4.16 : Login Form Snapshot for entering username and password.

<Form.Label>Post Title</Form.Label>

<input *className*="form-control"

*type*="text" *placeholder*="Enter Title here" *value* = {*this*.post\_title}

*onChange* ={*this*.handletitle} />

<Form.Text *className*="text-muted">Limit is

50 total characters.</Form.Text>

</Form.Group>

<Form.Group *controlId*="formBasicText">

<Form.Label>Post content</Form.Label>

<input *className*="form-control" *type*="text" *placeholder*="Enter post content here" *value* =

{*this*.post\_content} *onChange*={*this*.handlecontent}/>

<Form.Text *className*="Text-muted">Linit is a total of 200 characters</Form.Text>

</Form.Group>

<Button *variant*="primary" *type*="submit" *onClick*={*this*.handlesubmit}>Submit</Button>

</Form>

</div>

</div>

)

}

else{

return(

<div>

<h1>Need to be logged in to create a post</h1>

</div>

)

}

}

}

Fig 4.17 Create Post Form Snapshot for creating a new post.

)

}

}

Fig 4.18 : Create Sub\_Reddit Form Snapshot for creating sub\_reddit.

render(){

return(

<div>

<Header *value* ={"Create comment"}/>

<div *className* = "post">

<Form>

<Form.Group *controlId*="formBasicText">

<Form.Label>Comment Text</Form.Label>

<input *className*="form-control"

*type*="text" *placeholder*="Enter Comment here" *value* = {*this*.comment}

*onChange* ={*this*.handlecomment} />

<Form.Text *className*="text-muted">Limit is 1000 total characters.</Form.Text>

</Form.Group>

<Button *variant*="primary" *type*="submit" *onClick*={*this*.handlesubmit}>Submit</Button>

</Form>

</div>

</div>

)

}

}

Fig 4.19 : Writing Comments form snapshot.

# Connectivity with the database

**const express = require('express'); const cors = require('cors');**

**const mysql = require('mysql');**

**const bodyParser =require('body-parser');**

**const app = express();**

***//Queries to get all users and posts* const GET\_POSTS = 'select \* from post'; const GET\_USERS = 'select \* from users';**

**const connection = mysql.createConnection({ host:'localhost',**

**user:'root', password:'151997', database:'reddit'**

**});**

**connection.connect(function(*err*){ err?console.log(err):console.log('connection');**

**});**

Fig 4.20 Connecting with the database.

## Database Testing

**CHAPTER 5 TESTING AND RESULTS**

This testing is the basic step to validate and check whether the entries input by the user are valid and also follow the conventions defined in the database such as the datatype, character limit, file type, patterns for email, etc. This ensures that the functional

requirements run smoothly and the system gives respective errors messages if the data entered is invalid. It also checks the data mapping, ACID (Atomicity, Consistency, Isolation, Durability) properties validation, Data integrity and Business role conformance.

## Test Cases

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TESTCASE ID** | **TESTCASE** | **EXPECTED OUTPUT** | **ACTUAL OUTPUT** | **STATUS**  **(PASS/FAIL)** |
| TDT01 | User details are entered in a valid database. | User should be added with the details into the database. | User is added to the database with the details. | PASS |
| TDT02 | User details entered into an invalid database. | Message should be displayed by saying ‘Invalid Database’. | Database not found. | FAIL |
| TDT03 | Post details are entered in a valid database. | Post should be added with the details into the database. | Post is added to the database with the details. | PASS |
| TDT04 | Post details entered into an invalid database. | Message should be displayed by saying ‘Invalid Database’. | Database not found. | FAIL |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TDT05 | Sub\_reddit details are entered in a valid database. | Sub\_Reddit should be added with the details into the database. | Sub\_Reddit is added to the database with the details. | PASS |
| TDT06 | Sub\_reddit details entered into an invalid database. | Message should be displayed by saying ‘Invalid Database’. | Database not found. | FAIL |
| TDT07 | User logins with correct credentials. | Message should be displayed saying ‘Logged in successfully’. | Message displayed saying ‘Logged in successfully’. | PASS |
| TDT08 | User logins with incorrect credentials | Error message should be displayed saying ‘Invalid credentials’. | Error message displayed saying ‘Invalid credentials’. | FAIL |

* 1. **Front End Testing**

This part of testing is the process where we check the Graphical user interface of the website/application and the integration of the HTML pages at the frontend with the databases at the backend, the actions done at the frontend should be reflected

respectively as defined at the database and the backend. This includes adding of new members to the databases through forms, and also makes sure that all the required fields are entered accordingly.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TESTCASE ID** | **TESTCASE** | **EXPECTED OUTPUT** | **ACTUAL OUTPUT** | **STATUS**  **(PASS/FAIL)** |
| TFE01 | Register new user with proper credentials. | Message displaying  “Registered Successfully” | Message displaying  “Registered Successfully” | PASS |
| TFE02 | Register new user with  improper credentials. | Error message displaying  “Registering Failed” | Error message displaying  “Registering Failed” | FAIL |
| TFE03 | User login with proper credentials | Login should be valid. | Login should be valid. | PASS |
| TFE04 | User login with improper credentials. | Login Failed. | Login Failed. | PASS |
| TFE05 | Displaying the home page. | Home page displaying successfully. | Home page displaying successfully. | PASS |
| TFE06 | Displaying valid Posts. | Posts are displayed accurately. | Posts are displayed accurately. | PASS |
| TFE07 | Displaying a Sub\_reddit. | Sub\_Reddit displayed successfully. | Sub\_Reddit displayed successfully. | PASS |
| TFE08 | Displaying posts of a specified user. | Display posts of a specified user. | Display posts of a specified user. | PASS |
| TFE09 | Displaying posts of a specified Sub\_Reddit. | Display posts successfully. | Display posts successfully. | PASS |
| TFE10 | Comments of a post. | Display  Comments | Display  Comments | PASS |

## System Testing

This stage of testing involves checking the compatibility of the system associated with the project; it also checks the performance of the system when the application/website is run on it. Password hashing and security algorithms are also tested in this stage.

## Test Cases

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TESTCASE ID** | **TESTCASE** | **EXPECTED OUTPUT** | **ACTUAL OUTPUT** | **STATUS**  **(PASS/FAIL)** |
| TST01 | User login through a valid password. | User session  login should be successful. | User session started | PASS |
| TST02 | User login through an invalid password. | Error message saying ‘Invalid Credentials’ should be displayed. | ‘Invalid credentials’ is displayed. | FAIL |
| TST03 | Stable Internet connection available in the system. | All the icons  referred trough Font Awesome (fa) should be clearly visible. | Icons for  Home, login, user, are displayed. | PASS |
| TST04 | Stable internet connection unavailable in the system. | Icons should not be visible but the appropriate text describing the icon is visible. | Icon texts like ‘User\_Icon’,  ‘Home\_Icon’ etc. are visible. | FAIL |
| TST05 | Node.js proper installed with all the required modules. | The application and server should run smoothly. | The application runs smoothly without errors | PASS |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TST06 | Node.js is not completely  installed and some modules are missing. | There should be appropriate error  messages like ‘Module not found’,  ‘Command not found’. | Message displayed saying  ‘Command not found’. | FAIL |
| TST07 | Display supports HD image view | Full image of the satellite should be displayed in the Description page of the page. | Full image is displayed with the set  resolution size | PASS |
| TST08 | Display does not support HD image view | Full image is not displayed in the  Description page | Image not displayed and background is blank. | FAIL |
| TST09 | System Browser support full HTML5 | All the functionalities will run properly | Functionalities run properly | PASS |
| TST10 | System browser does not support HTML5  (Internet Explorer 8 or lower) | Some HTML5 functionalities might not work properly like autofocus, video and audio tags are not supported. | Some functionalities don’t work smoothly | FAIL |

Reddit Clone Conclusion

## Limitations

**CHAPTER 6 CONCLUSION**

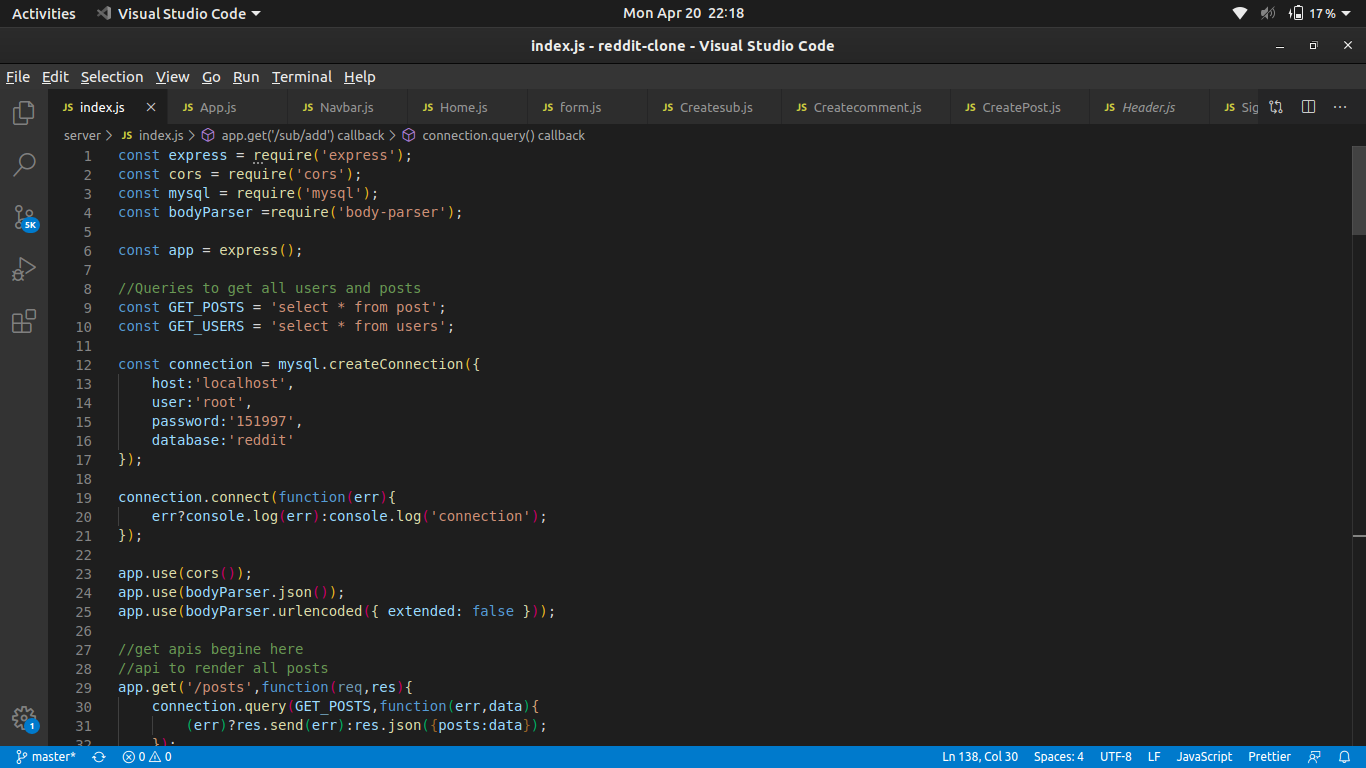
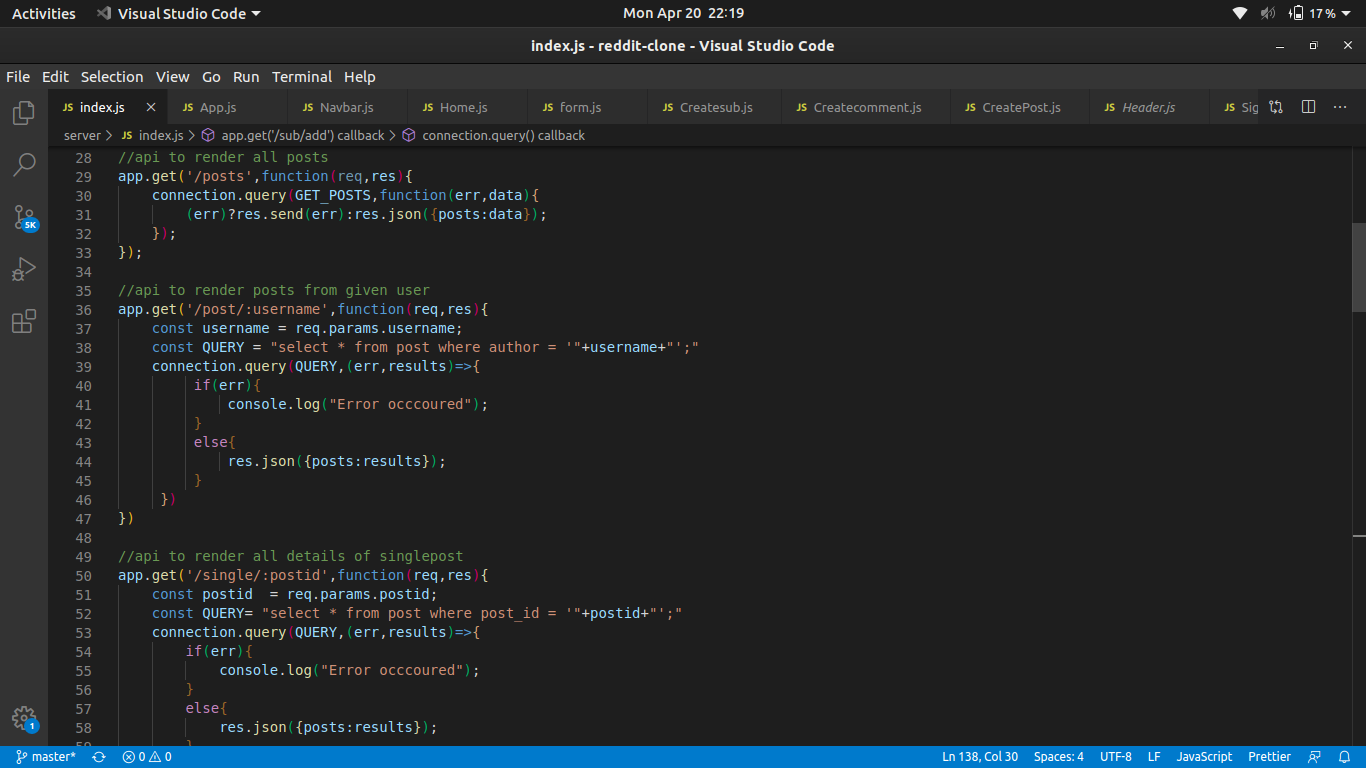
Currently the clone although functioning is not able to perform all the functions that

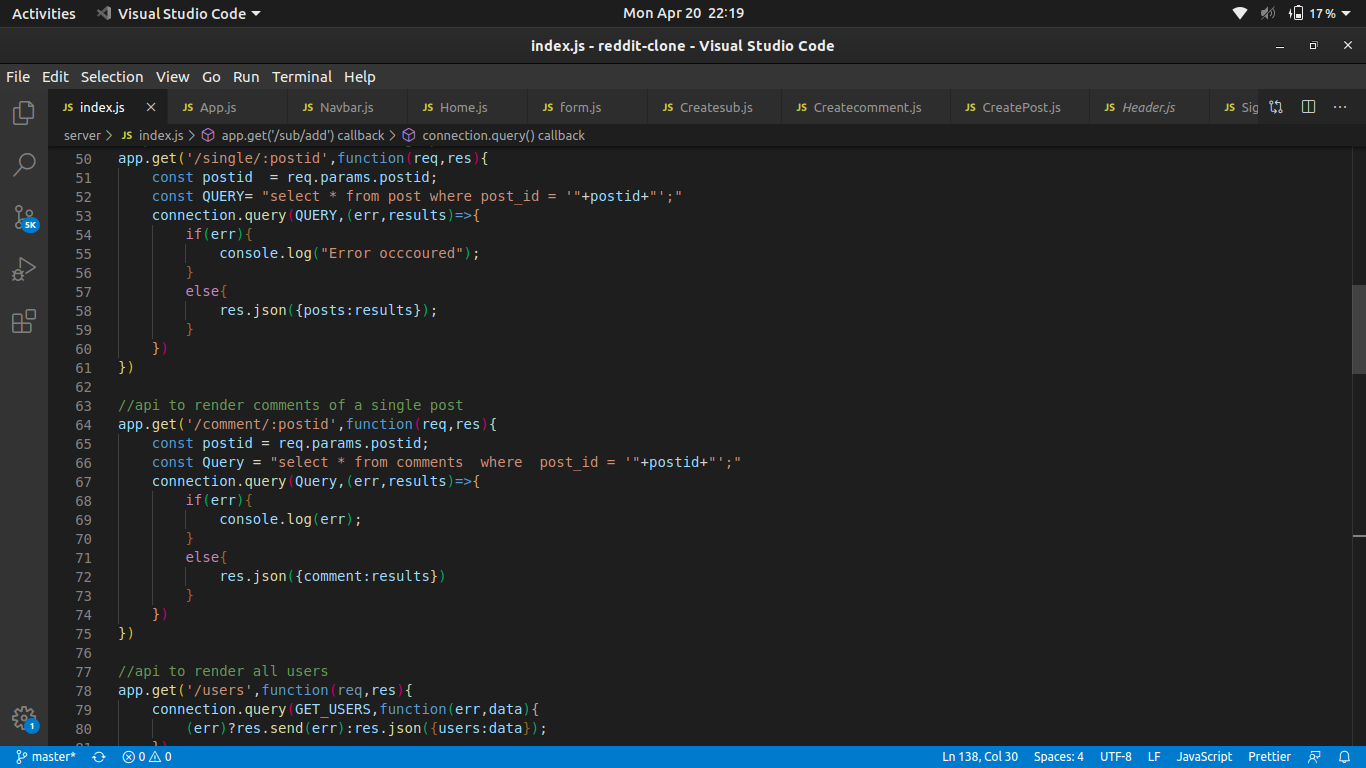
reddit does, most notably it is limited currently by text, Users can only interact using text and images, videos, files and likes can not be shared. It is also not secure as the passwords are currently stored in a regular text format.

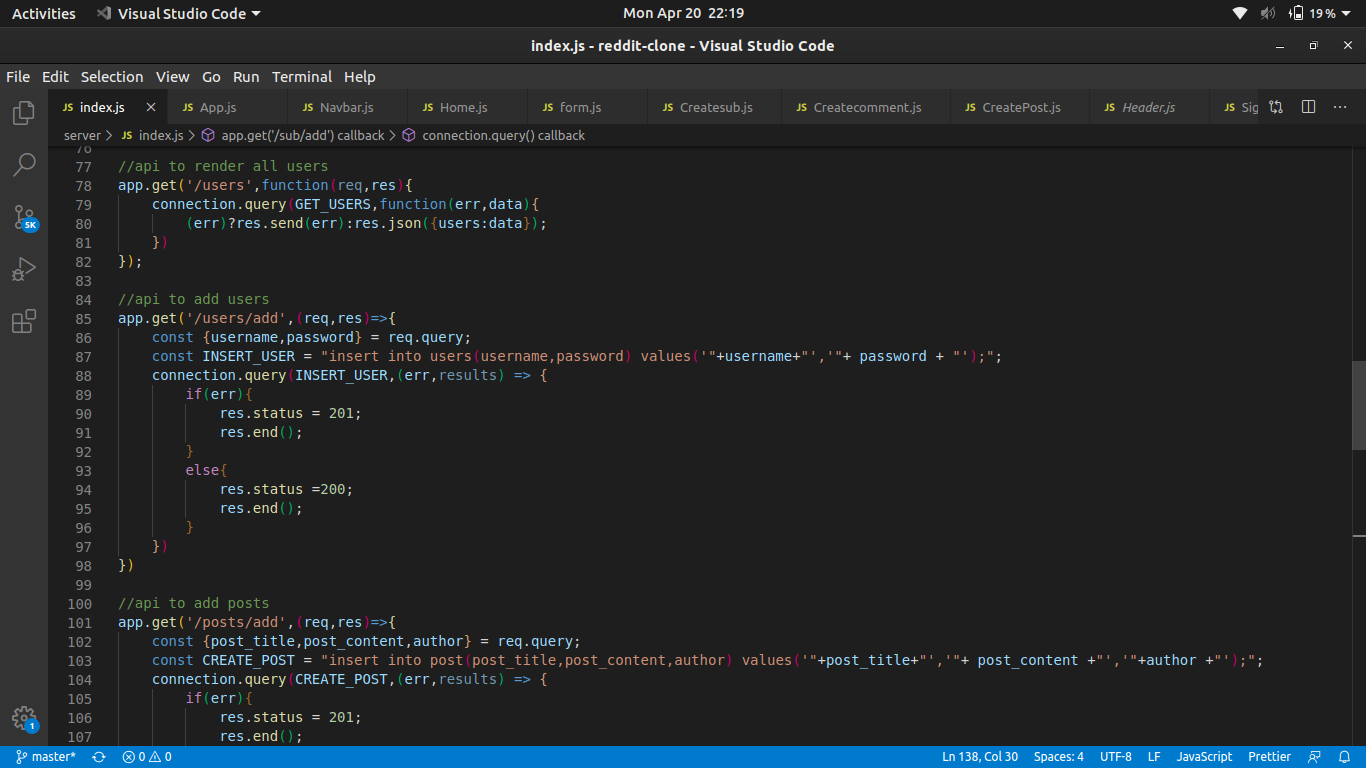
## Future Enhancements

* + - The following steps are what we plan to implement in the app.
    - Sharing of files images and videos.
    - Addition of encryption to password by using SHA sum while storing passwords.
    - Adding email-id to every user and confirming their emails to be RV certified, also confirming if they are actually linked to the email(Verification of user).

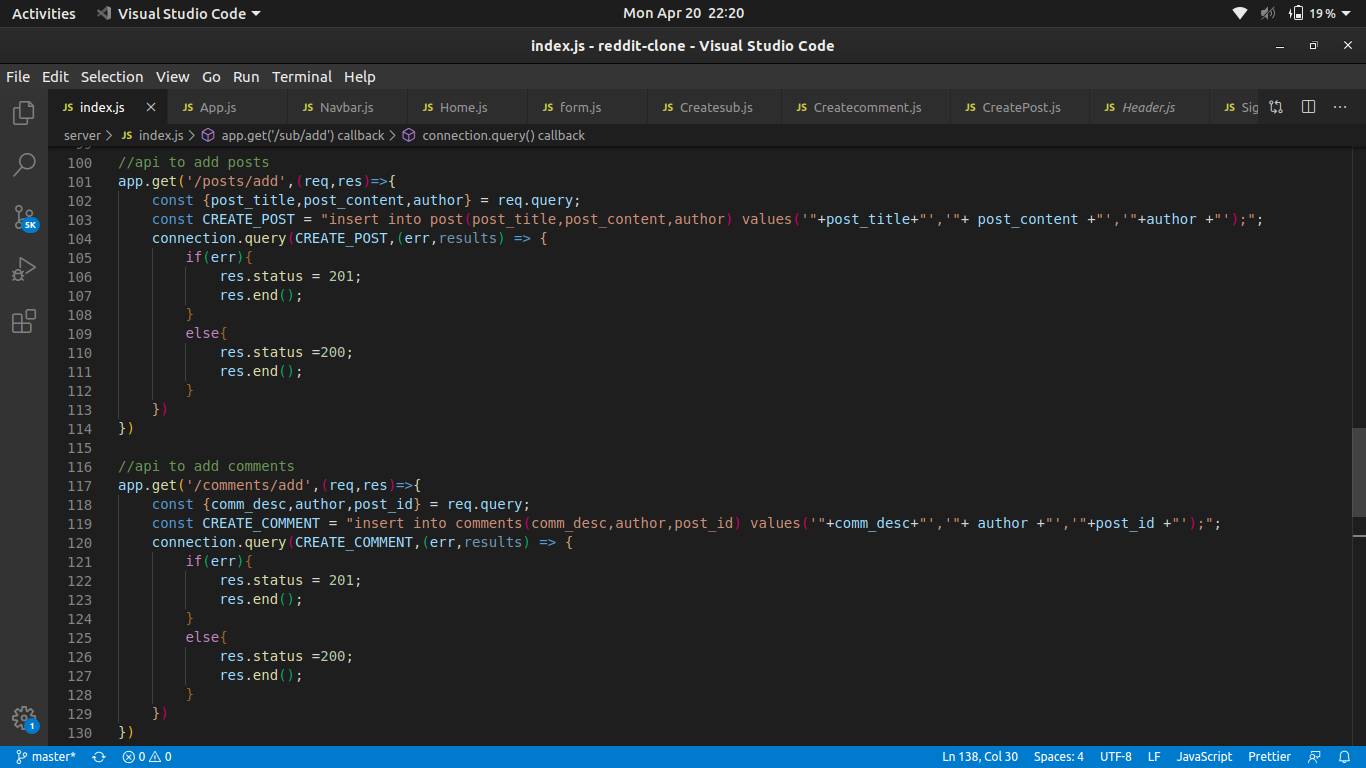
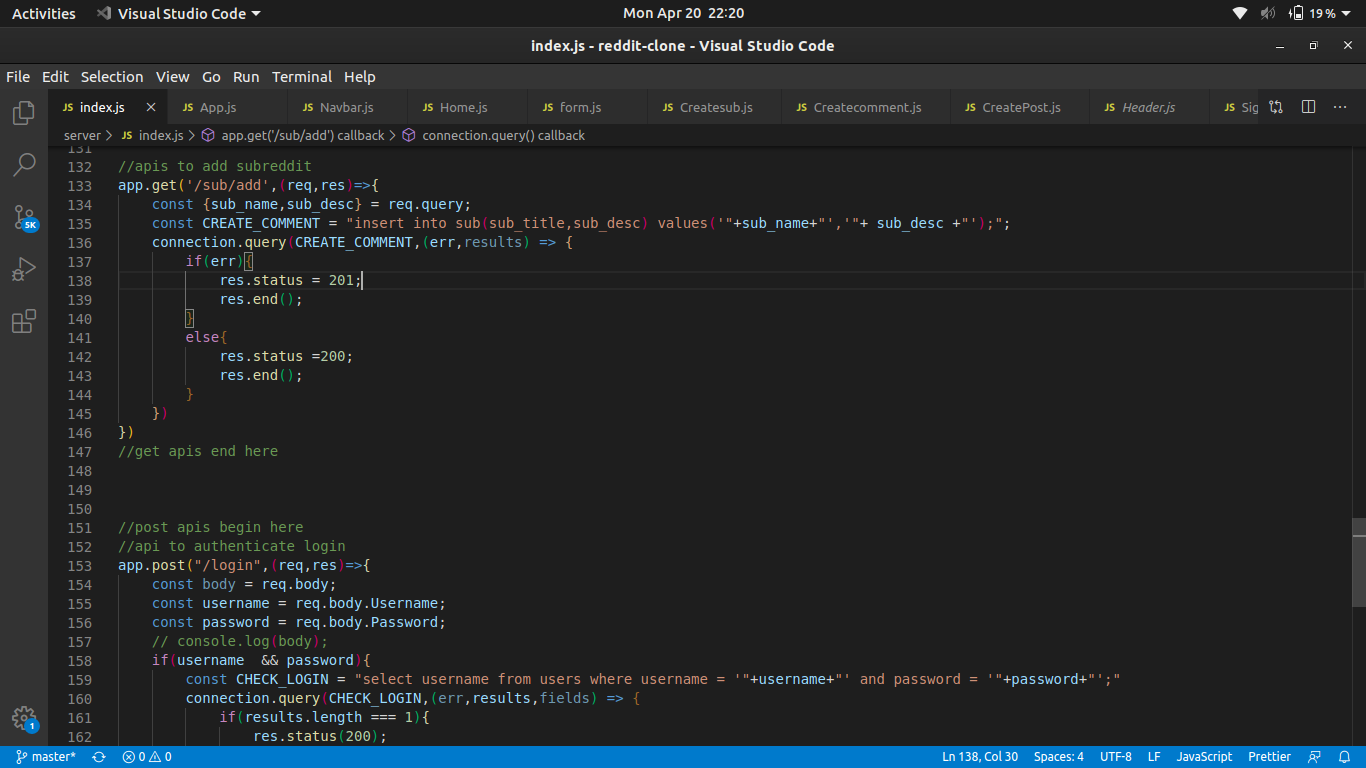
**APPENDIX A**

A1.1A1.2

A1.3



A1.4

A1.5A1.6

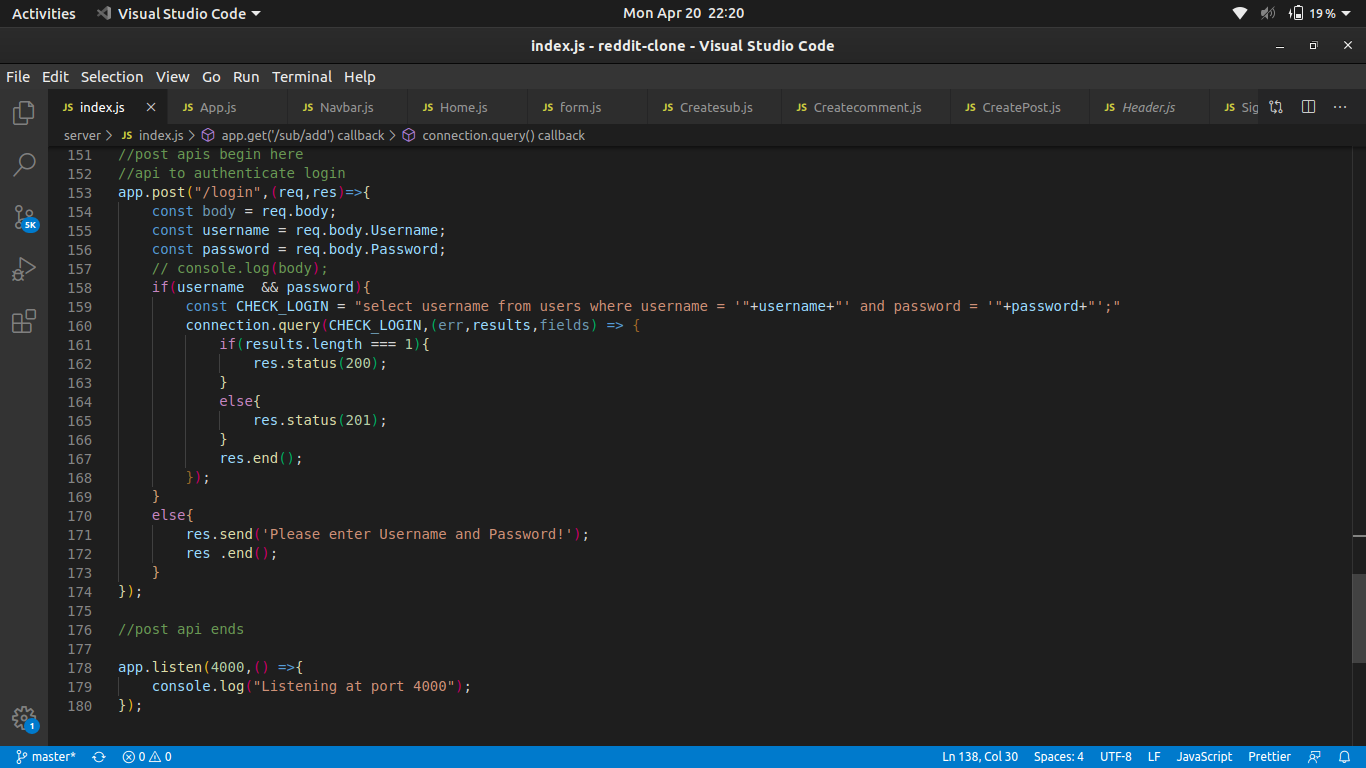
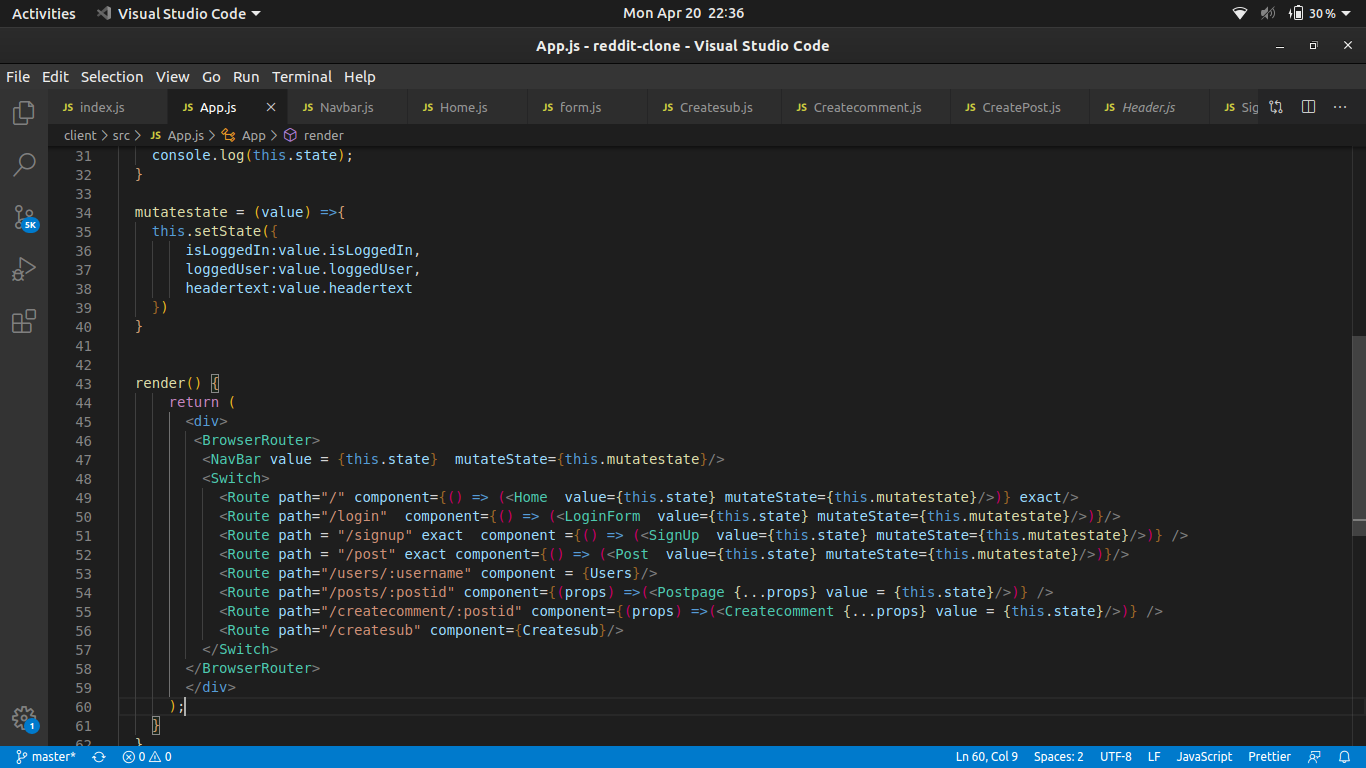
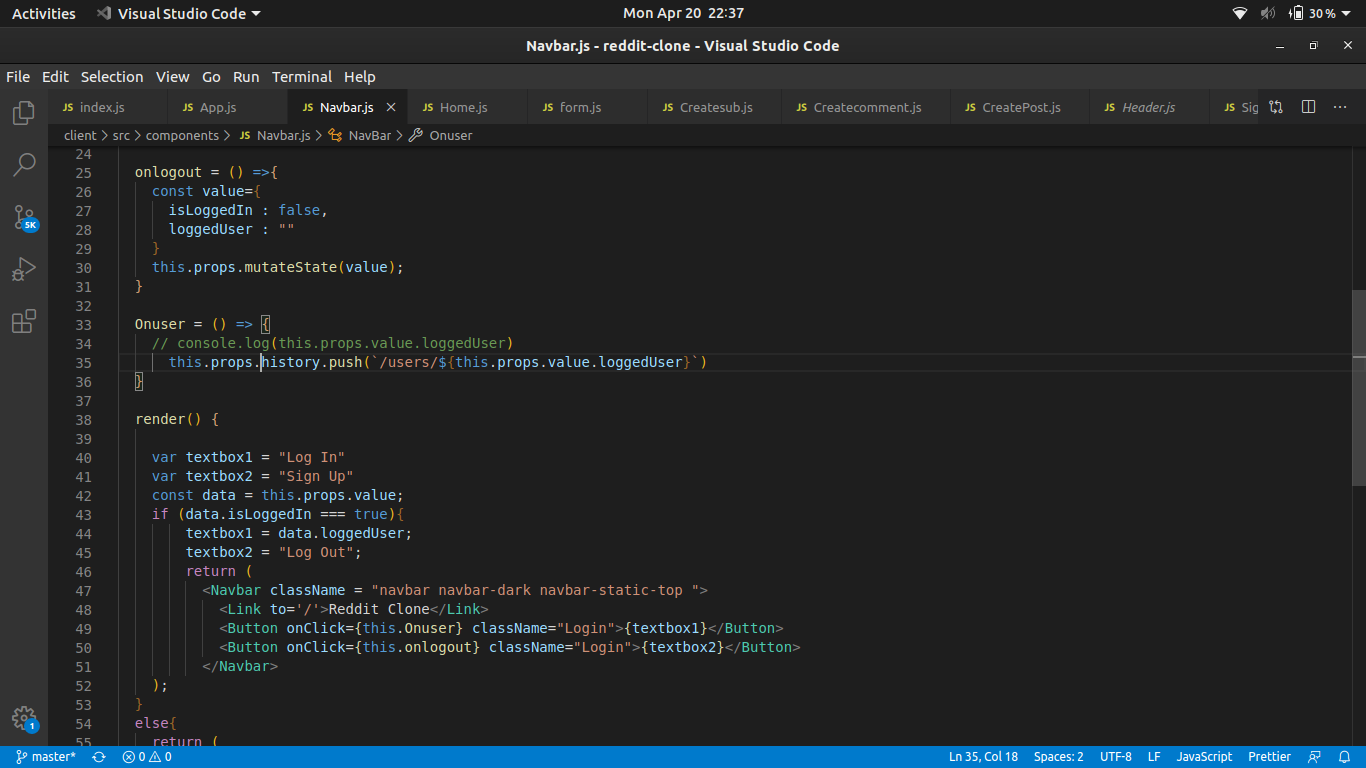
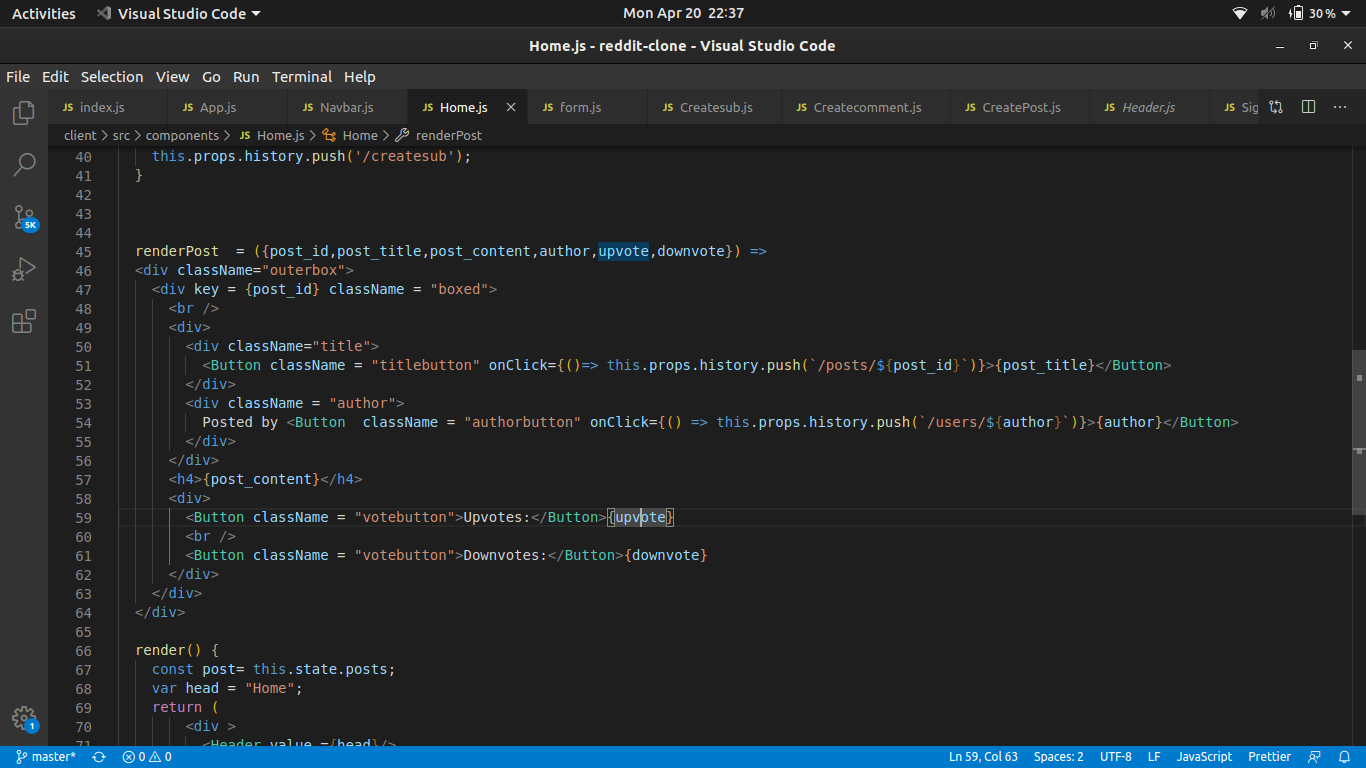
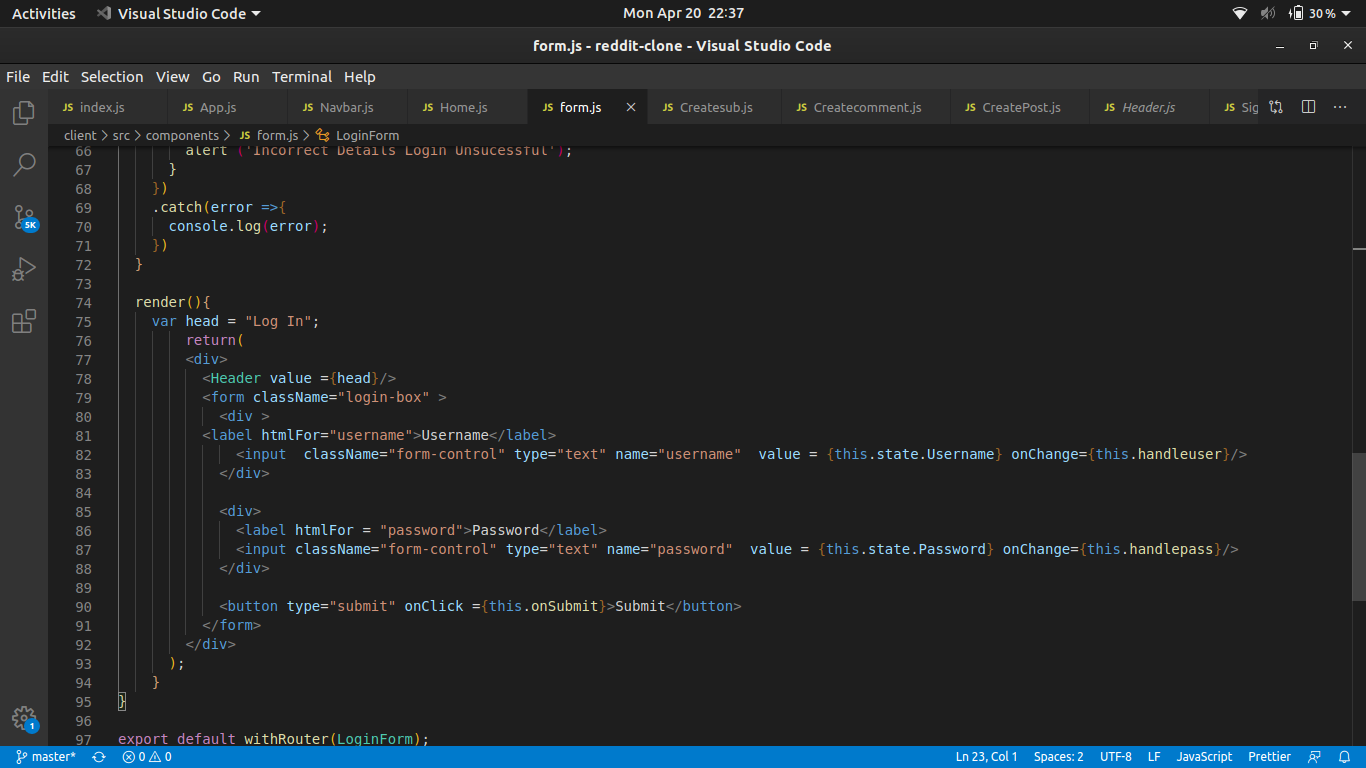
A1.7

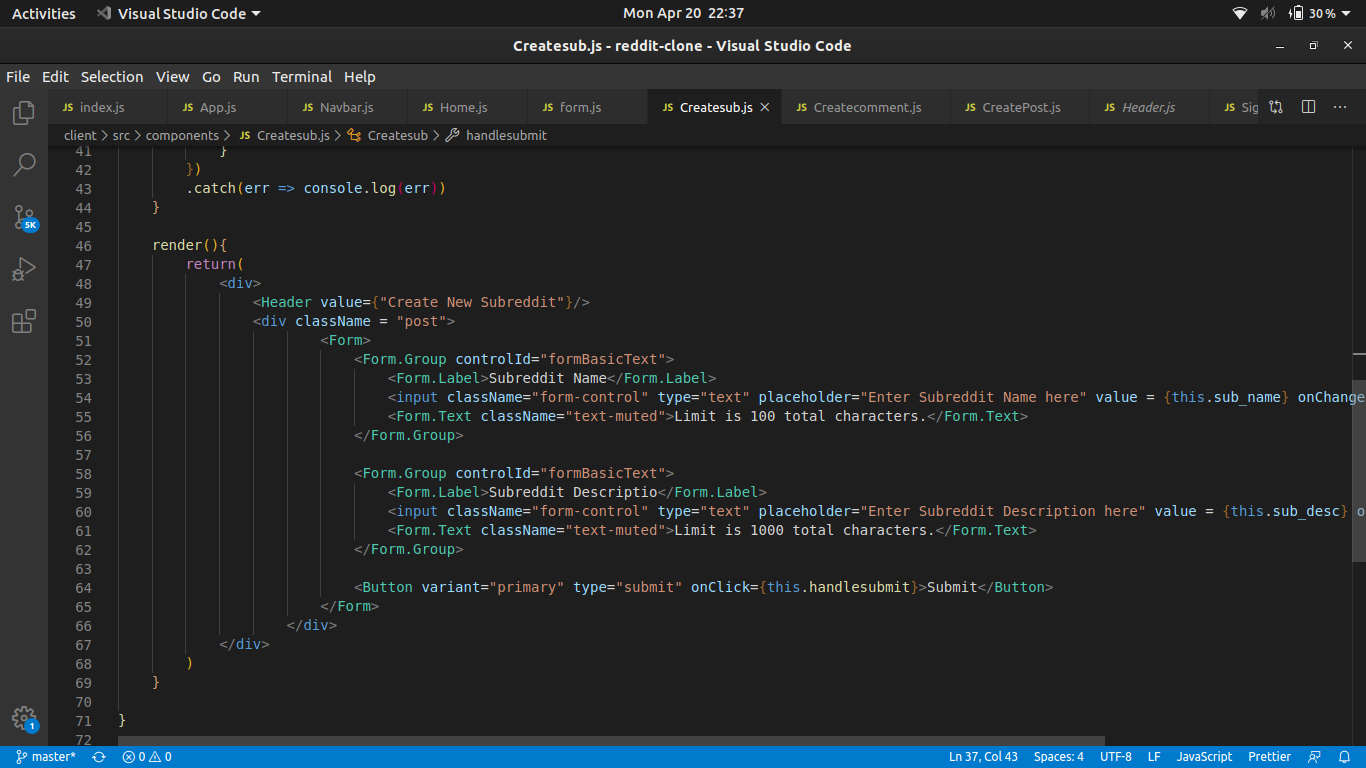
Fig A1-A7 : Snapshot of the server side part of the project

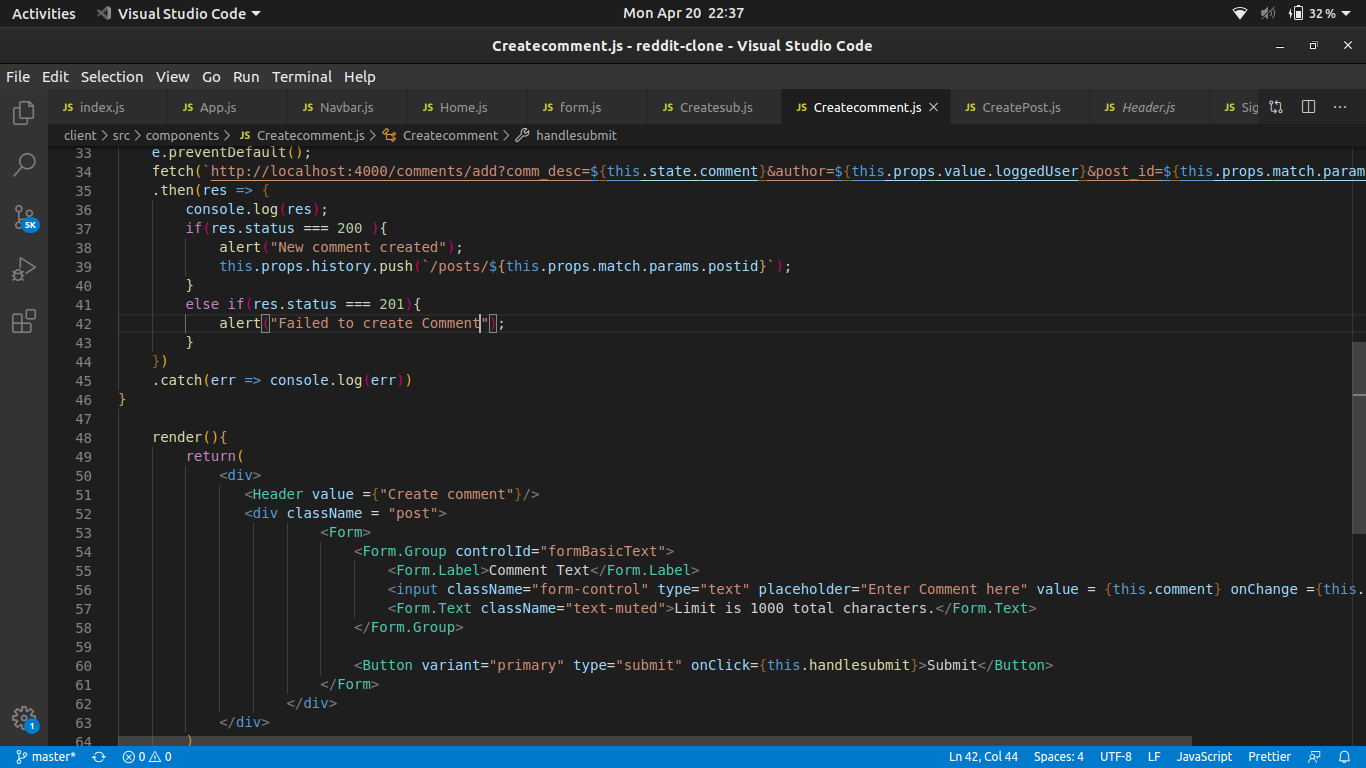
A1.8

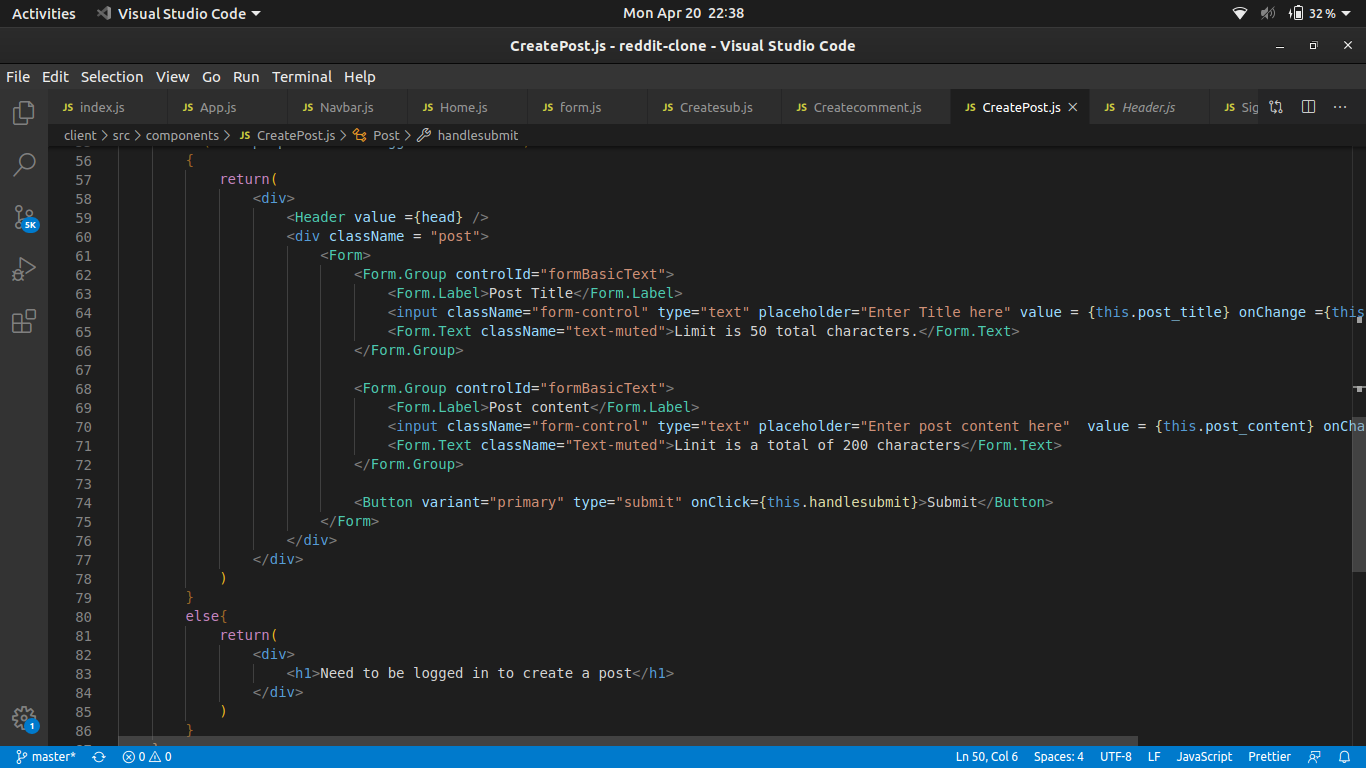
A1.9

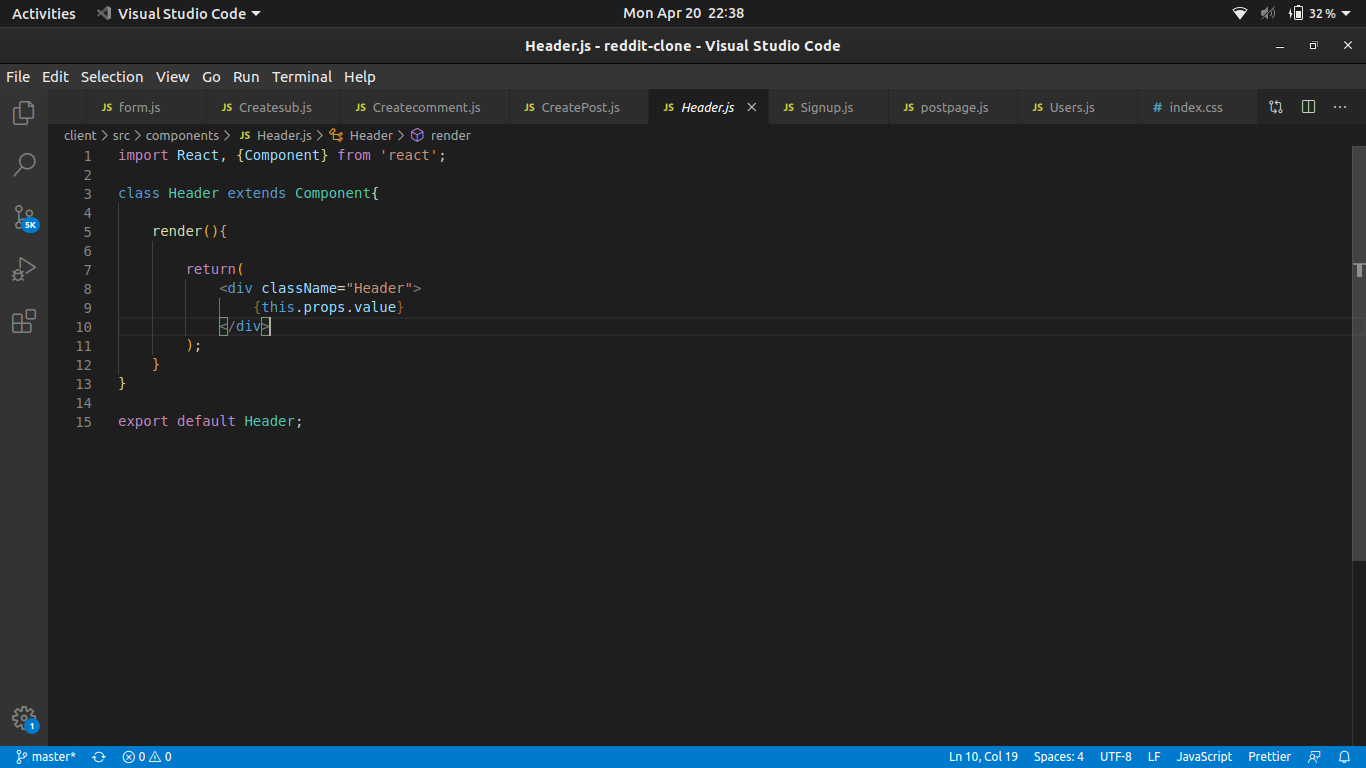
A1.10

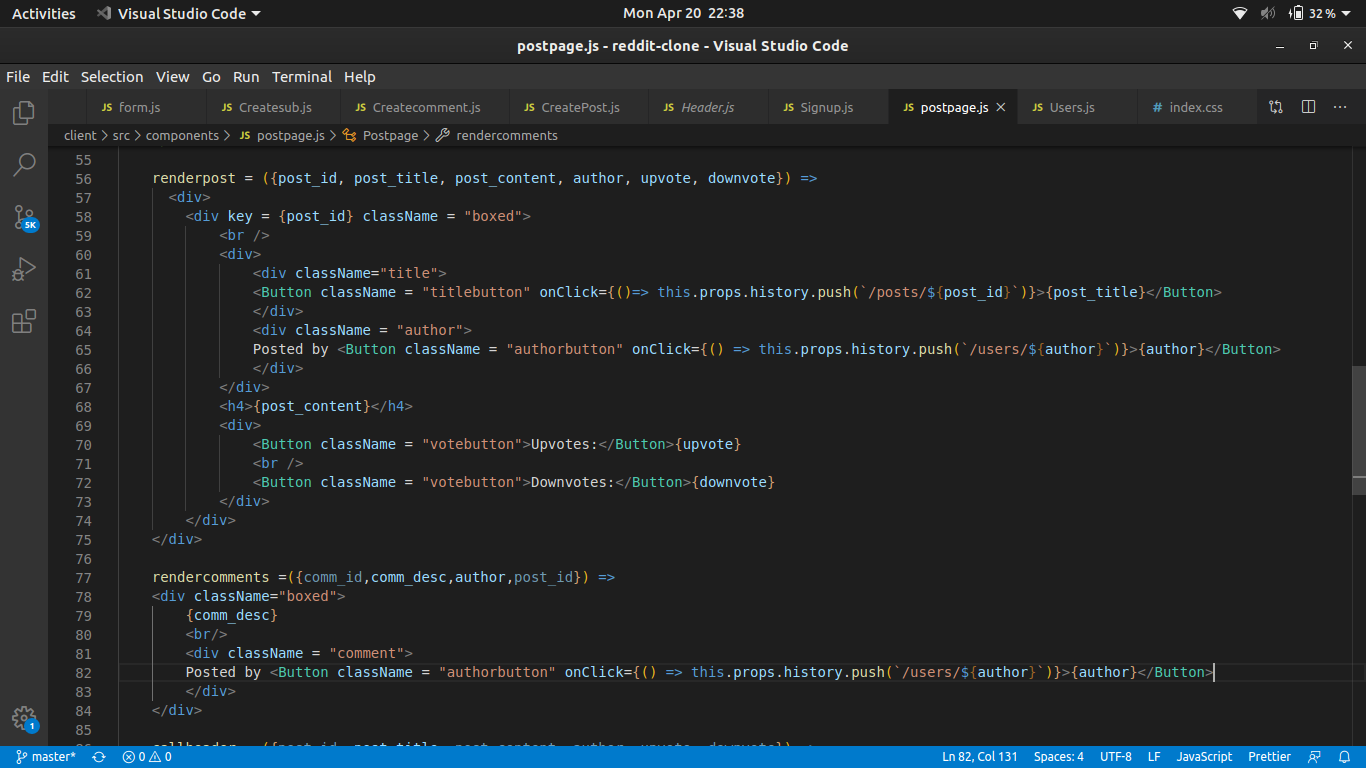
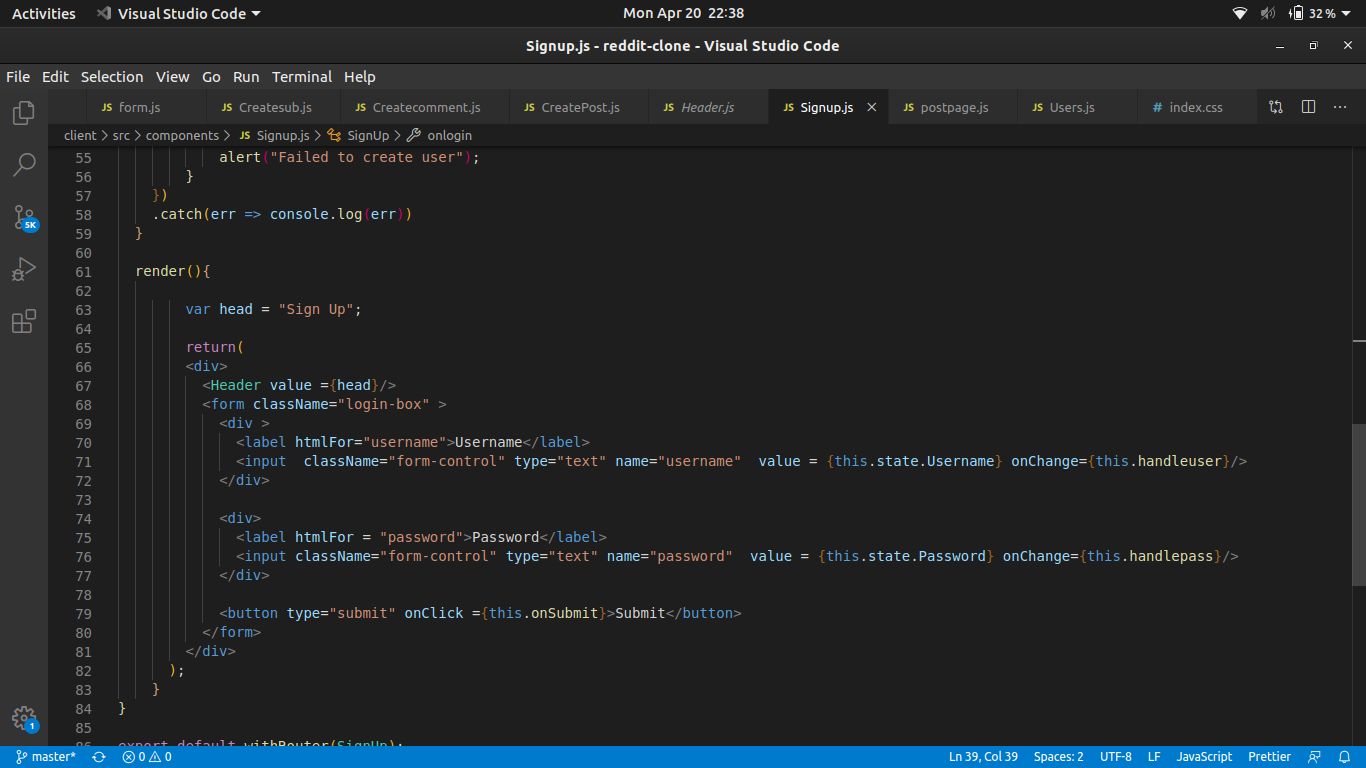
A1.11

A1.12

  
  
  
A1.13

  
A1.14

A1.15

A1.16  
A1.17

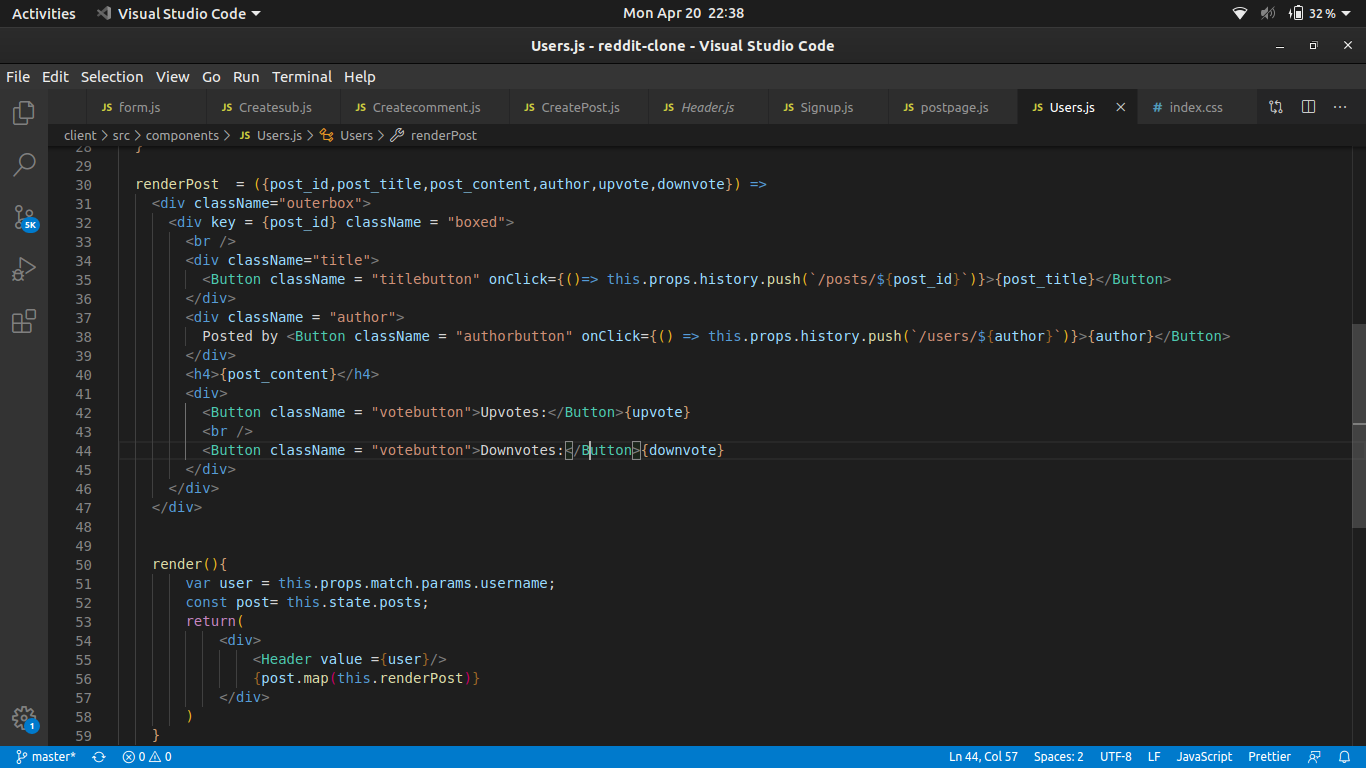
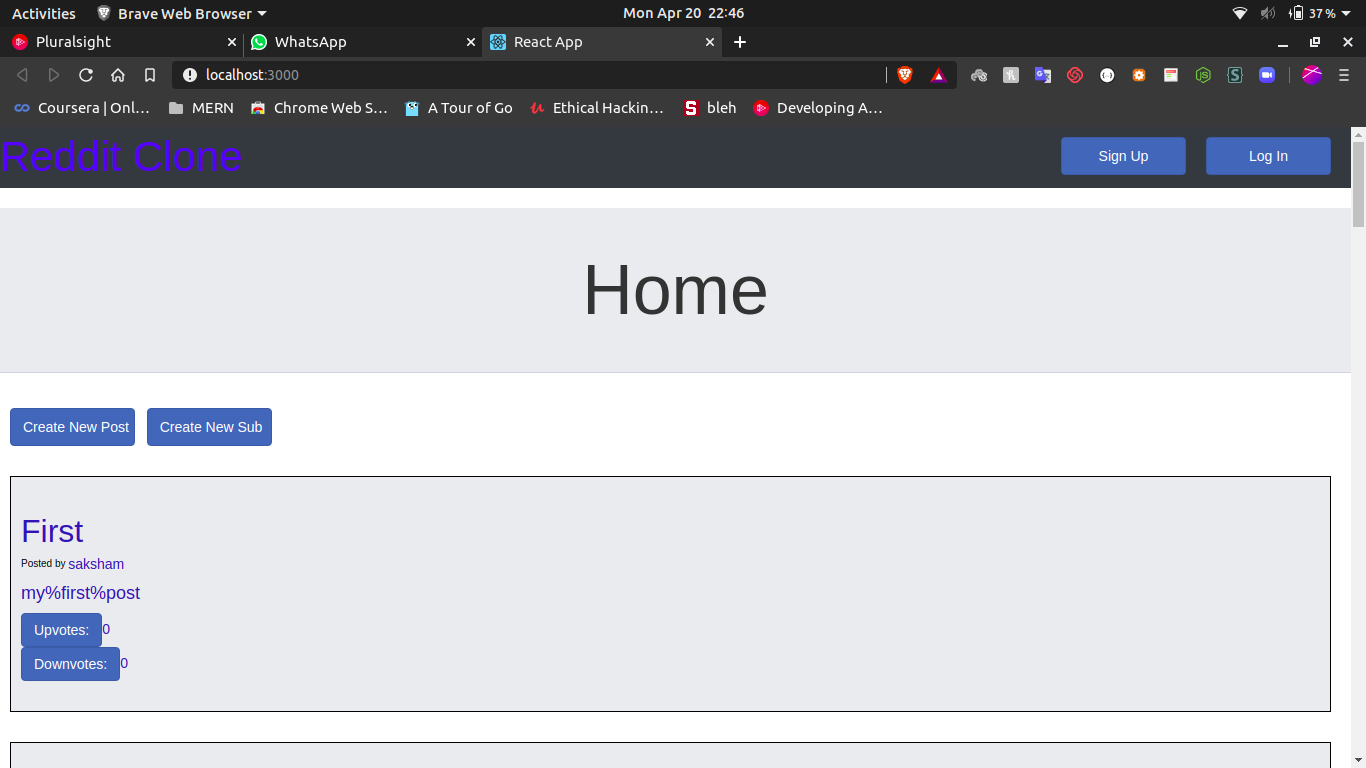
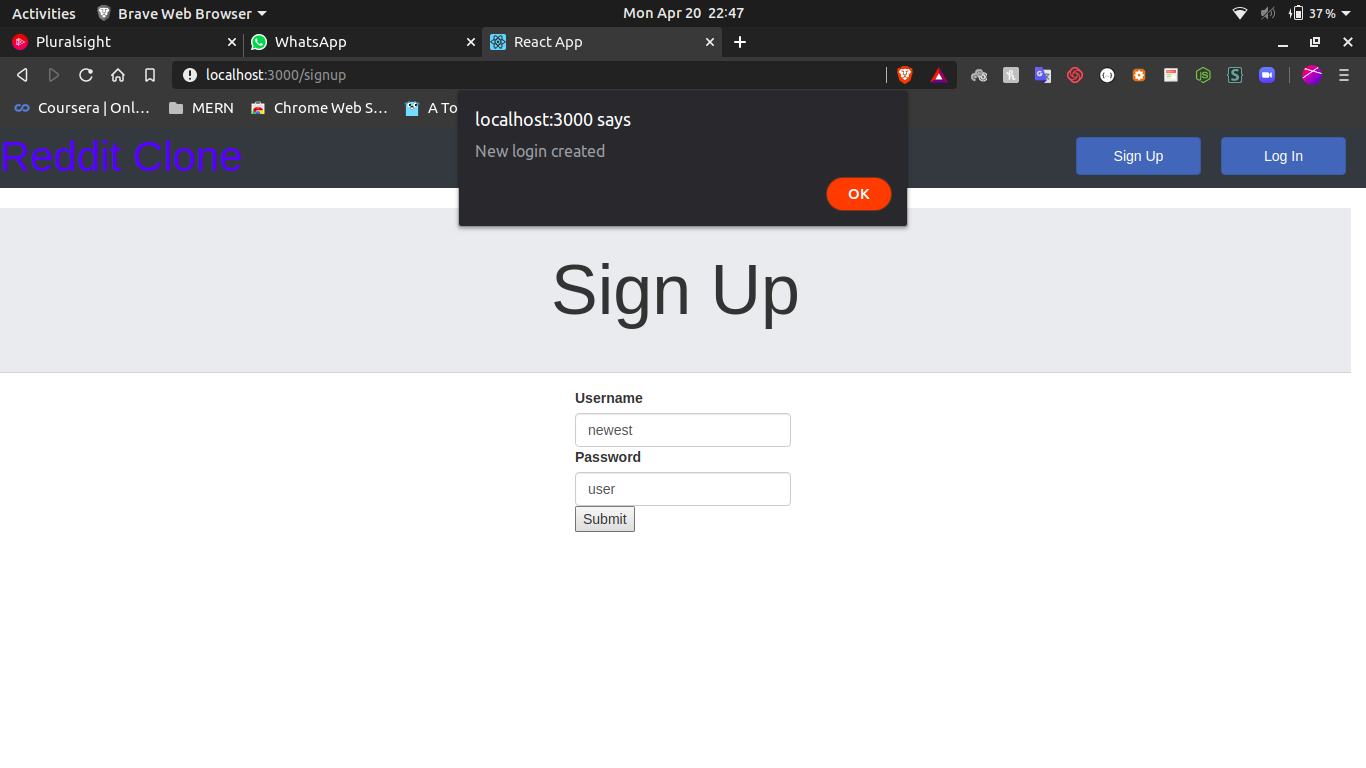
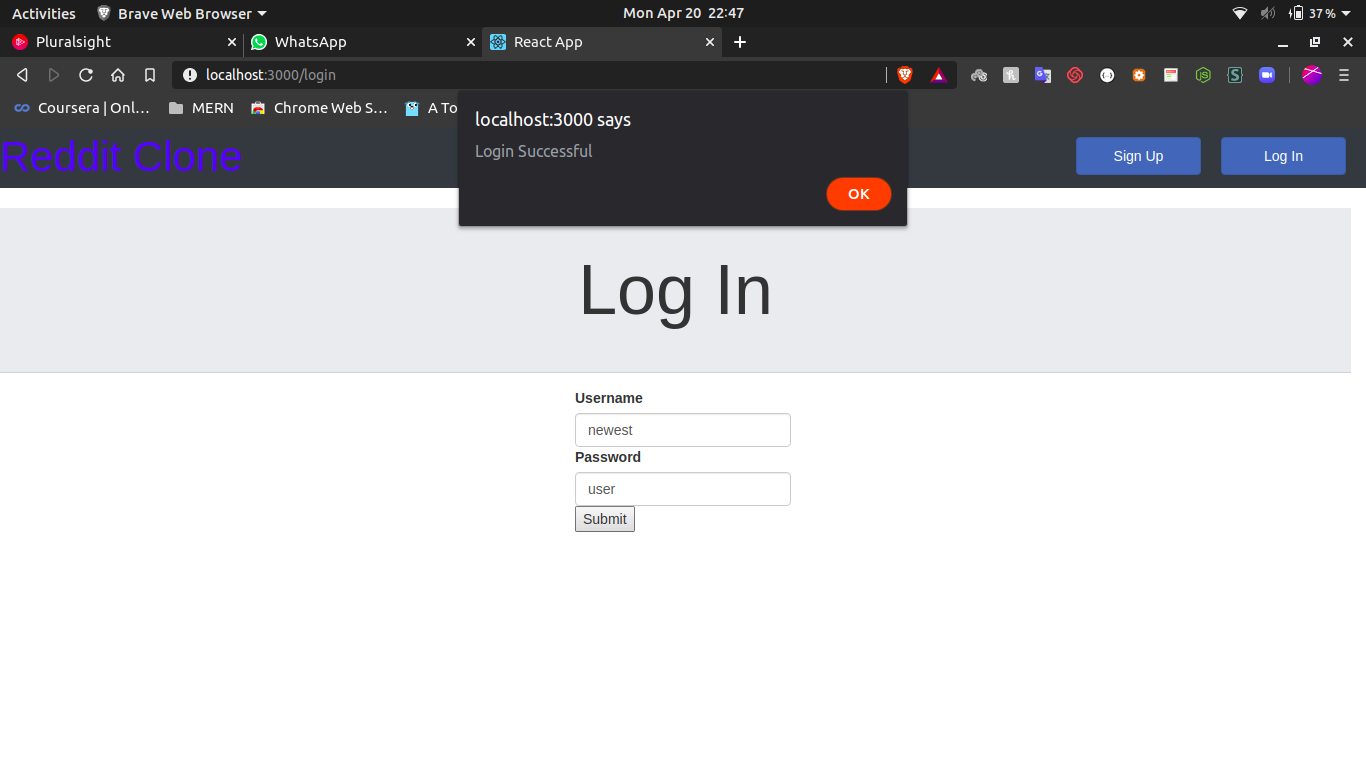
A1.18

Fig A1.8-A1.18 are the codes of all the different components ins this project

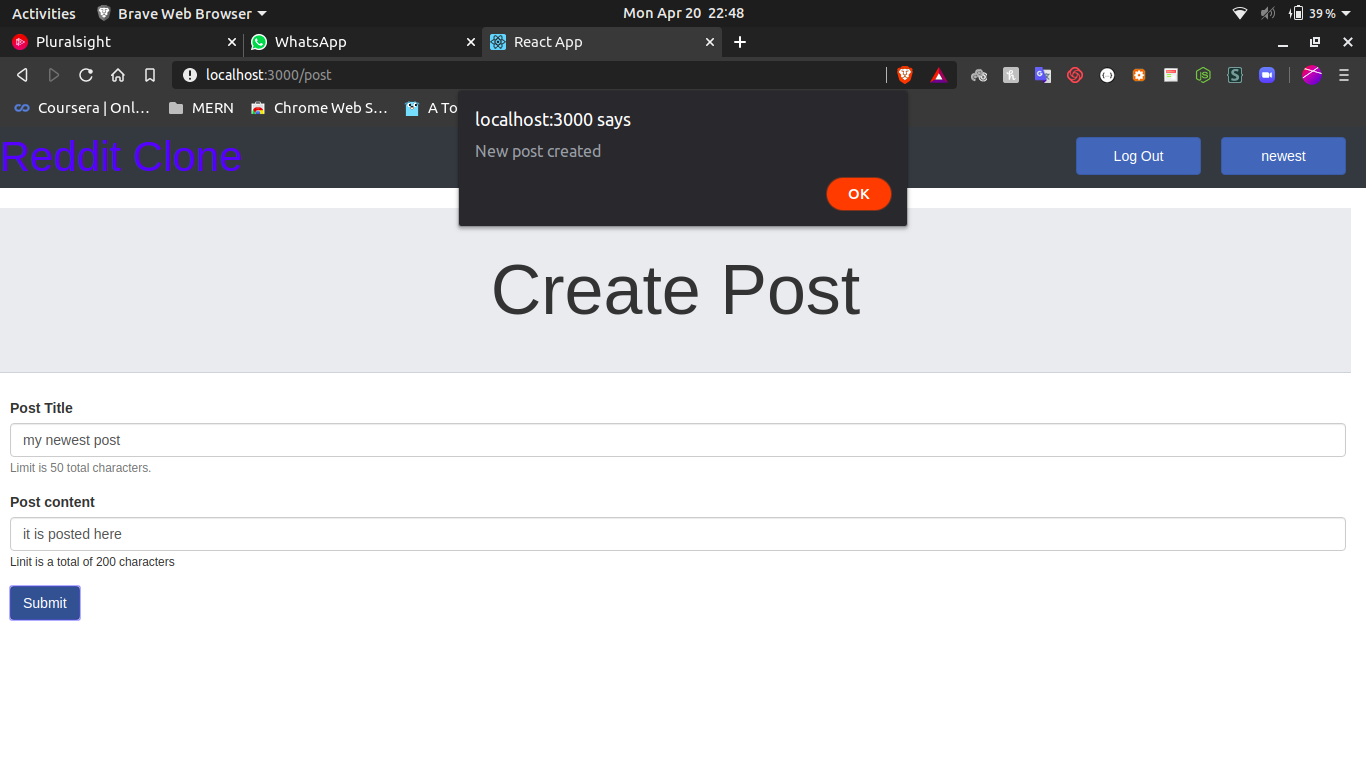
**APPENDIX B**

B1.1

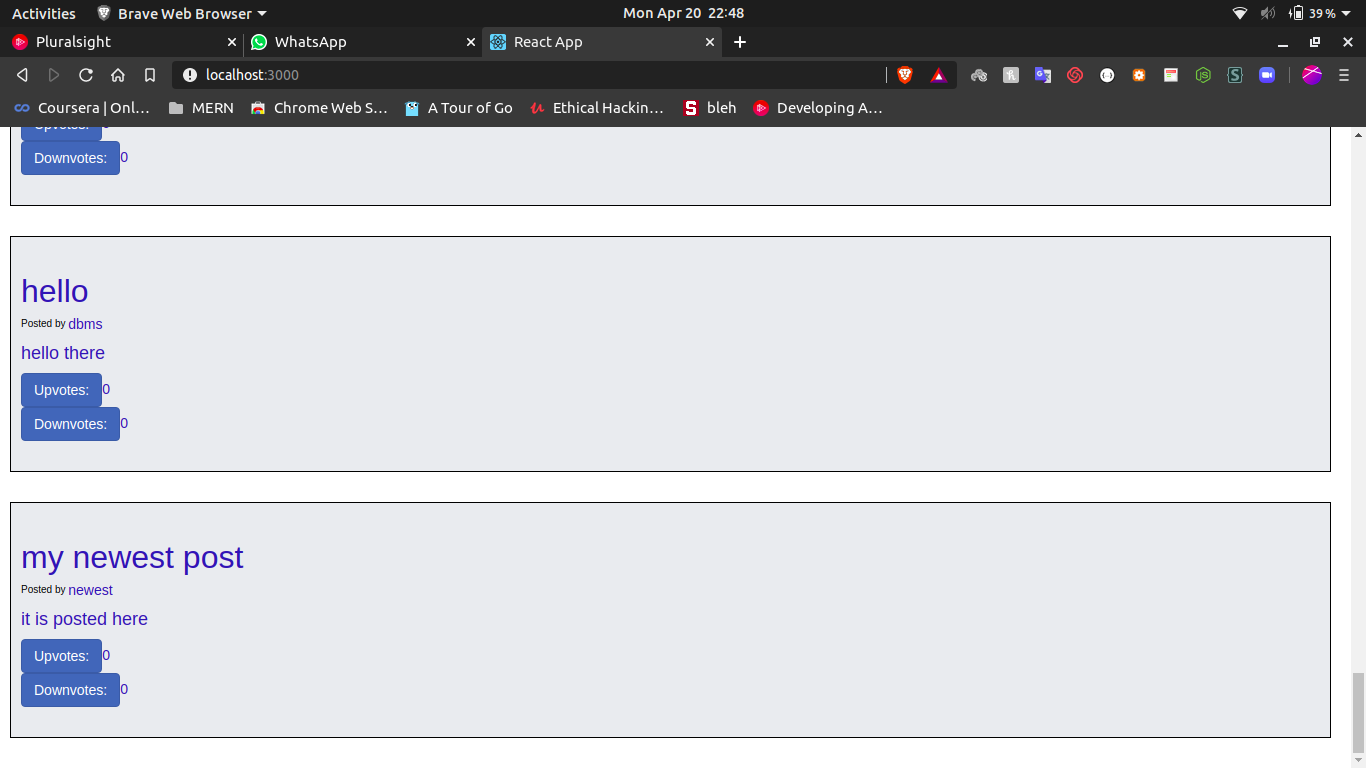
B1.2

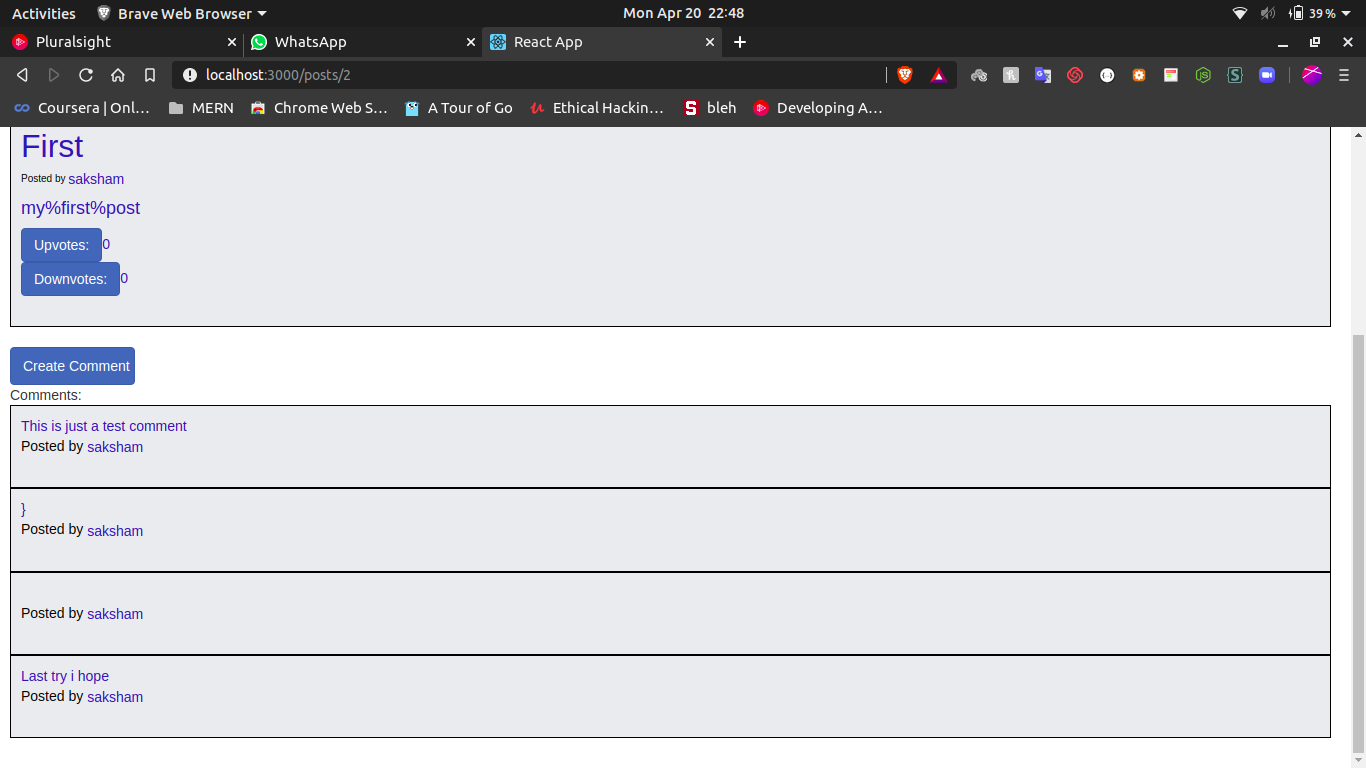


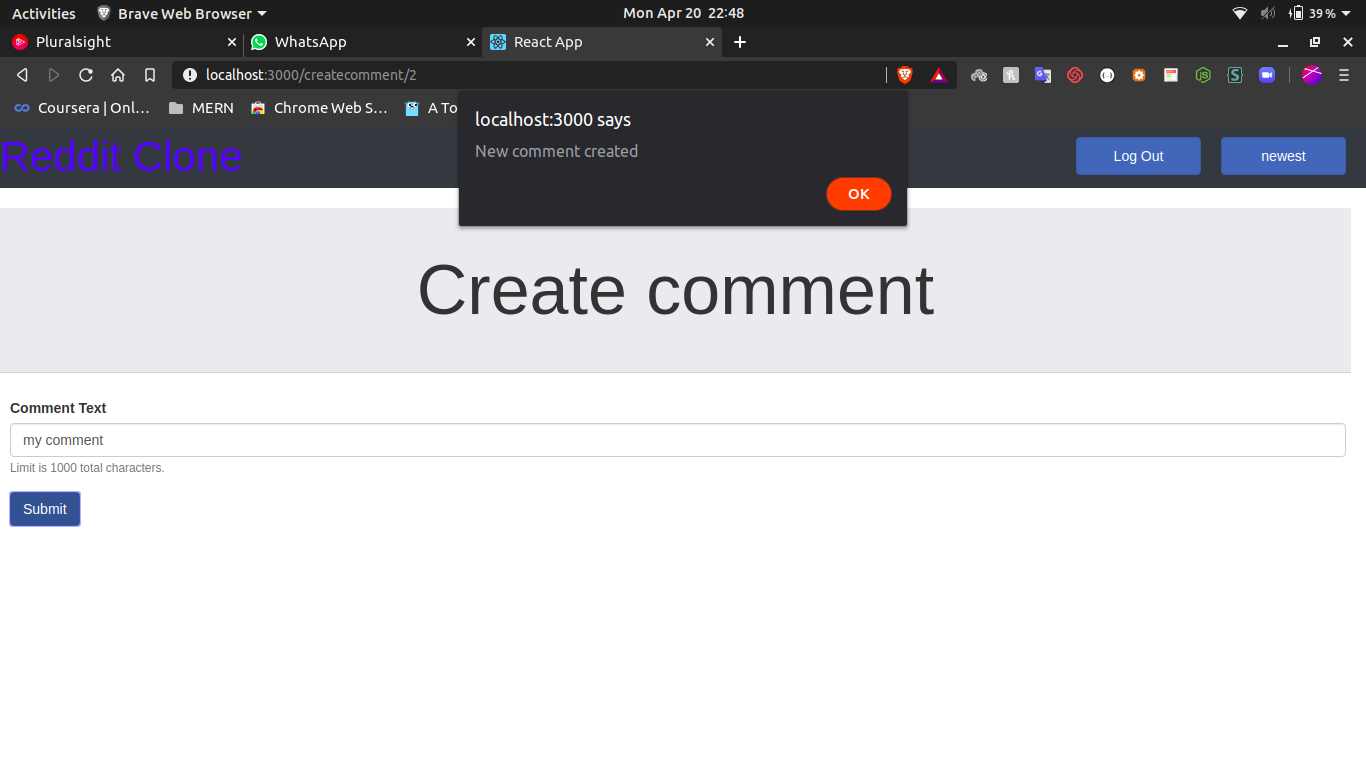
B1.3

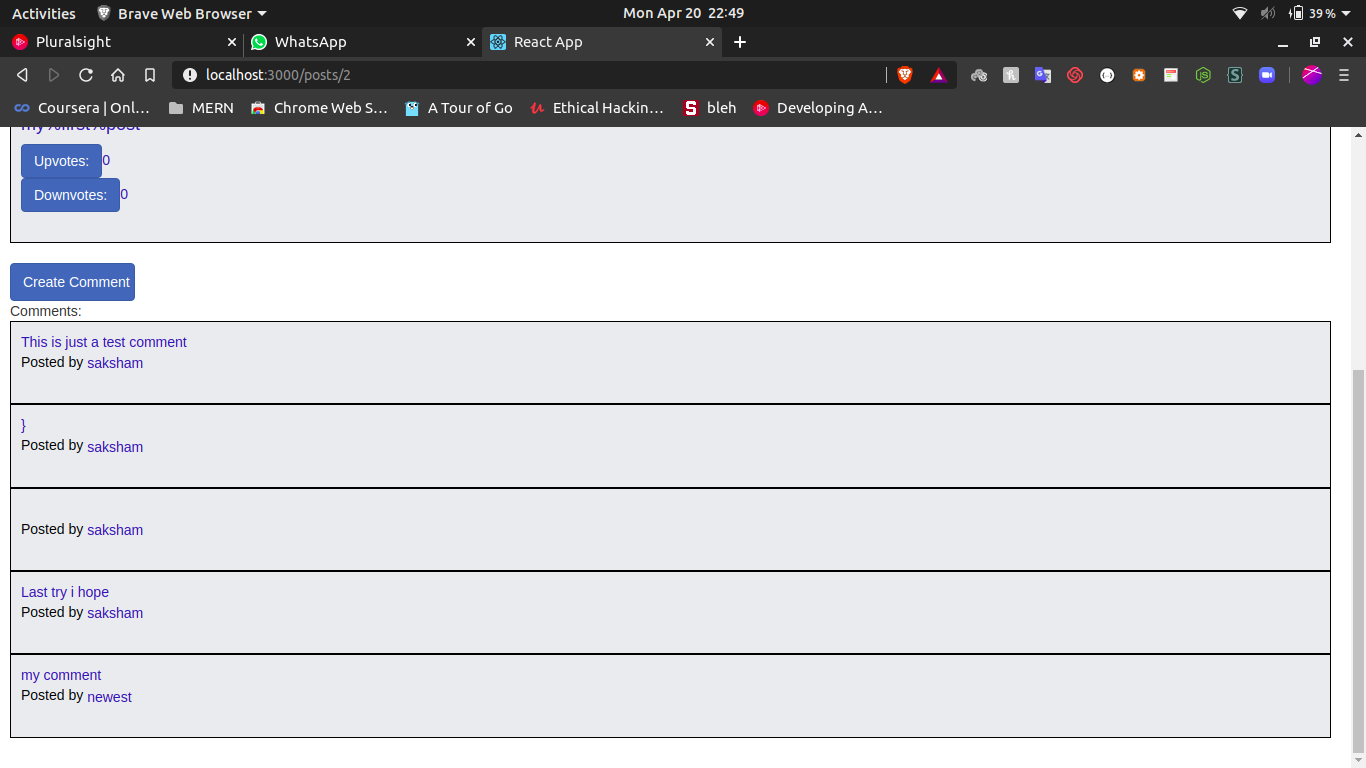


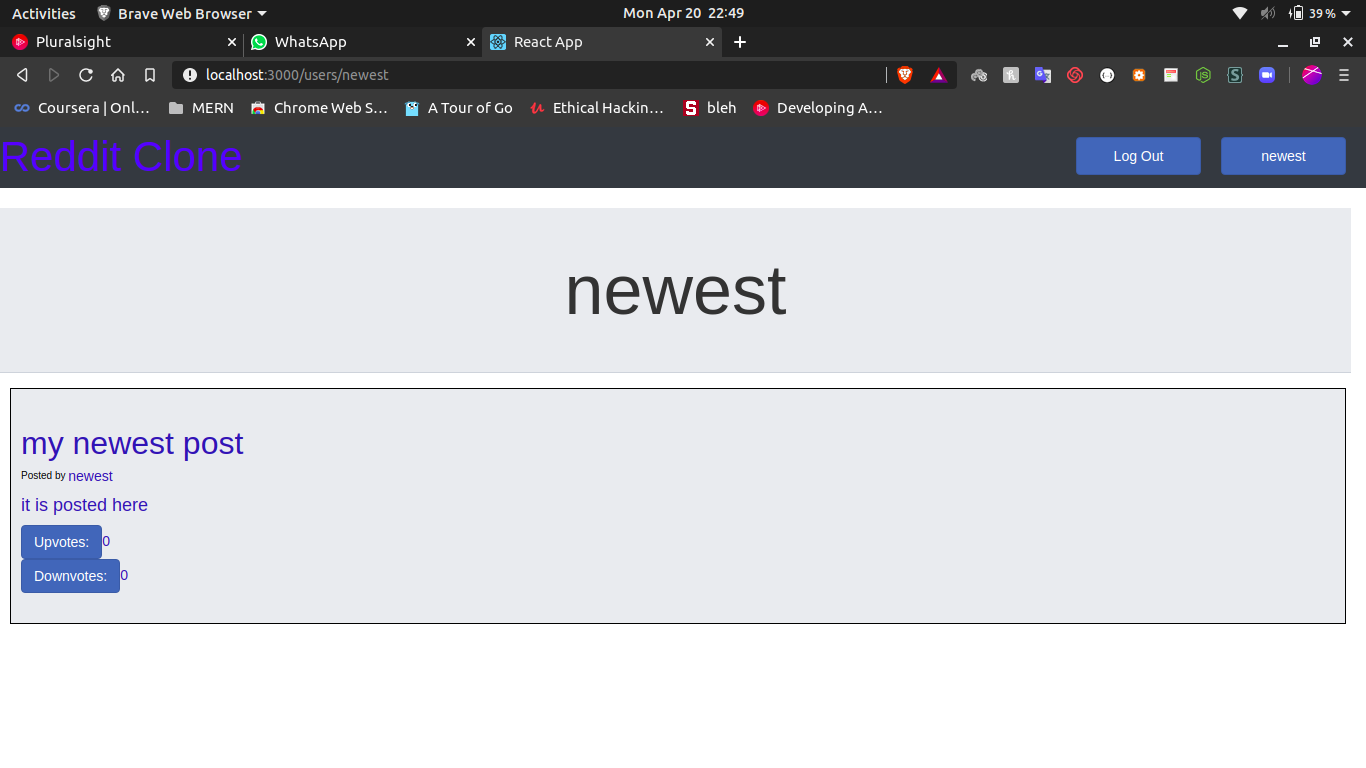
B1.4

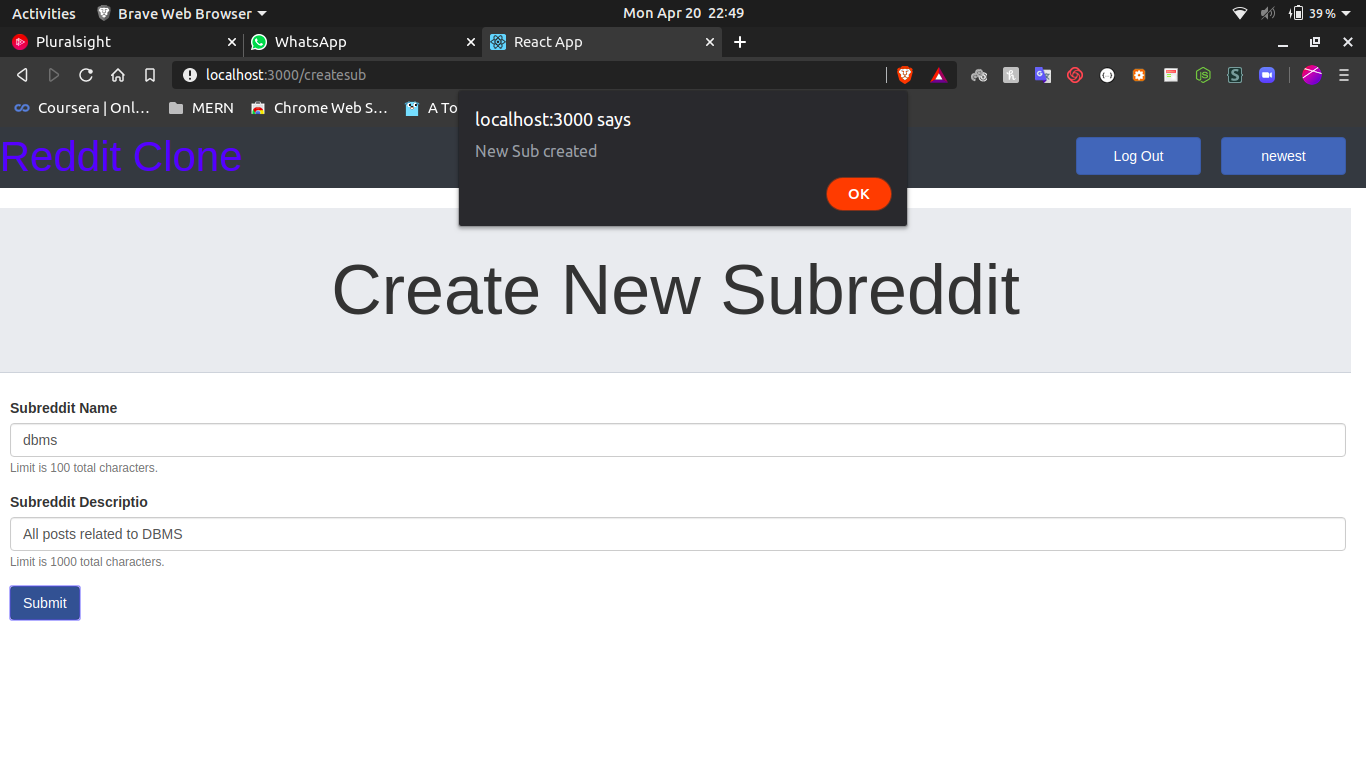
B1.5

B1.6

B1.7

B1.8

B1.9

B1.10