

Mini Project

MITRO

(Instagram Clone Application)

Project Report

Department of Computer Engineering & Application
Institute of Engineering & Technology



GLA
UNIVERSITY
MATHURA
Established vide U.P. Act 21 of 2010.

Submitted To: -

Mr. Mandeep Singh (Technical Trainer)

Submitted By: -

Anand Khandelwal (181500090)

Anubhav Gupta (181500108)

Md Rashid (181500381)

Ritik Upadhyay (181500579)

Hrithik Bandil (181500274)

Table of Contents

Abstract	4
Acknowledgement	5
1. Introduction	
1.1 General Introduction to the topic	6
1.2 Motivation	7
1.3 Area of Computer Science	8
1.4 Hardware and Software Requirements	9
2. Problem definition	12
3. Aim & Objectives	13
4. Scope	15
5. Modules	18
6. Design and its Screenshots	23
7. Progress till Date	28
8. Coding	29
9. Implementation	40
10. Database	41
11. Testing	52
References	53

Abstract

The Project that we are making is more of a skill gaining work by which we are seeking to learn different technologies and prospects of developing any software or application. Basically, we are going through each and every step so that in future we can easily make our own app by the knowledge and experience gain by this. MITRO is a ready to use social media software solution which caters to the general users, Influencers, Celebrities and well as Business Profiles.

The report presents the three tasks that is being in continuation during college time at GLA University which are listed below:

1. Understanding of the Problem objective & implication.
2. Understanding of the data & building of the model.
3. Evaluation of the model.

All these tasks have been completed successfully and results were according to expectations. All the tasks were need very systematic approach, starting from the collection of the data to the implementation of the solution and till evaluation of the System. The most challenging task was the domain knowledge, to understand the language. It is one of the major areas and really need very fundamental and conceptual knowledge of HTML, CSS, JavaScript, and Node.JS.

Acknowledgement

It gives us a great sense of pleasure to present the synopsis of the Mini Project (**MITRO – Instagram Clone Application**) undertaken during B.Tech IIIrd Year, this project itself is going to be acknowledgement to the inspiration, drive and technical assistance will be contribute to it by many individuals. We owe special debt of gratitude to **Mr. Mandeep Singh** (Technical Trainer Department of CEA) for providing us with an encouraging platform to develop this project which thus helped us in shaping our abilities towards a constructive goal and for his constant support and guidance to our work. His sincerity, Thoroughness and perseverance is being a constant source of inspiration for us. We believe that he will shower us with all his Extensively experienced ideas and insightful comments at different stages of the project & also taught us about the latest industry-oriented technologies. We also do not like to miss the opportunity to acknowledge the contribution of all faculty members of the department for their kind guidance and co-operation.

Anand Khandelwal (181500090)

Anubhav Gupta (181500108)

Md Rashid (181500381)

Ritik Upadhyay (181500579)

Hrithik Bandil(181500274)

Introduction

1.1 General Introduction to the topic

Mitro is basically a social networking application which allows users to edit and upload photos and short videos through a mobile app. Users can add a caption to each of their posts and use hashtags and location-based geotags to index these posts and make them searchable by other users within the app. Each post by a user appears on their followers' Instagram feeds and can also be viewed by the public when tagged using hashtags or geotags. Users also have the option of making their profile private so that only their followers can view their posts.

As with other social networking platforms, Mitro users can like, comment on and bookmark others' posts, as well as send private messages to their friends via the Instagram Direct feature. Photos can be shared on one or several other social media sites -- including Twitter, Facebook and Tumblr -- with a single click.

Mitro is not only a tool for individuals, but also for businesses. The photo-sharing app offers companies the opportunity to start a free business account to promote their brand and products. Companies with business accounts have access to free engagement and impression metrics.

1.2 Motivation

Recreating a popular app is a fun way to learn a lot of complex programming skills. Instagram clone app includes an authentication system, database, storage, and more. Creating the app with React Native allows you to use the same code base for both Android and iOS devices. The eagerness to know and learn how the application is made and how it works led us to take the project of Instagram Clone. And this is more of a skill gaining work by which we are seeking to learn different technologies and prospects of developing any software or application. Basically, we are going through each and every step so that in future we can easily make our own app by the knowledge and experience gain from it.

1.3 Area of Computer Science

To create a Social Networking platform, one needs to employ people capable of doing systems analysis and design. Once a satisfactory design has been developed, one needs software engineering to build the system. Once the system is built, it needs to be tested and deployed, and then maintained. That again falls into the realm of software engineering.

Software engineering encompasses other specialized roles that include such things as quality assurance, database design, user interface design, etc. Operational computer systems are also supported by roles such as Computer Operations.

An alternative to designing and building a networking platform is to buy a commercially available system that has been designed to satisfy the general requirements for a user. The system would likely require some customization, which again falls into the realm of software engineering.

The only thing computer science has to do with all this is that it provides the theoretical underpinnings of the software engineering — just as physics provides the theoretical underpinnings to building a safe bridge. But physicists don't build bridges — bridge engineers build bridges.

1.4 Hardware and Software Requirements

A) Hardware Requirements (Minimum):

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware. A hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in case of operating systems. An HCL lists tested, compatibility and sometimes incompatible hardware devices for a particular operating system or application. The following sub-sections discuss the various aspects of hardware requirements.

1. i3 processor-based computer
2. 4GB Ram
3. 25 GB Hard Disk Space

B) Software Requirements (Minimum):

Software Requirements deal with defining software resource requirements and pre-requisites that need to be installed on a computer to provide optimal functioning of an application. These requirements or pre-requisites are generally not included in the software installation package and need to be installed separately before the software is installed.

1. Windows 10 OS
2. Visual Studio
3. GitHub
4. Web Browser
5. React
6. MySQL Lite
7. Node JS

1. **Web Browser:** A web browser to run codes and debug and view my work development and use it's inspect feature and console window.
 2. **VS Code:** It's a platform where I can write my code and maintain my Files.
 3. **Git and Git-Hub:** This was required for Hosting my website on internet.
-
1. **HTML5:** HTML5 is a mark-up language used for structuring and presenting content on the World Wide Web. It is the fifth and latest major version of HTML that is a World Wide Web Consortium recommendation.
 - HTML is the standard mark-up language for creating Web pages.
 - HTML describes the structure of a Web page.
 - HTML consists of a series of elements.
 - HTML elements tell the browser how to display the content.
 2. **CSS3:** Cascading Style Sheets (CSS) is a style sheet language used for describing the look and formatting of a document written in a mark-up language.
 - CSS describes how HTML elements are to be displayed on screen, paper, or in other media
 - CSS saves a lot of work. It can control the layout of multiple web pages all at once
 3. **JavaScript:** JavaScript often abbreviated as JS, is a programming language that conforms to the ECMAScript specification. JavaScript is high-level, often just-in-time compiled, and multi-paradigm. It has curly-bracket syntax, dynamic typing, prototype-based object-orientation, and first-class functions.

Alongside HTML and CSS, JavaScript is one of the core technologies of the World Wide Web. JavaScript enables interactive web pages and is an essential part of web applications. The vast majority of websites use it for client-side page behaviors, and all major web browsers have a dedicated JavaScript engine to execute it.

4. T-SQL: Transact-SQL (T-SQL) is Microsoft's and Sybase's proprietary extension to the SQL (Structured Query Language) used to interact with relational databases. T-SQL expands on the SQL standard to include procedural programming, local variables, various support functions for string processing, data processing, mathematics, etc. and changes to the DELETE and UPDATE statements.

Transact-SQL is central to using Microsoft SQL Server. All applications that communicate with an instance of SQL Server do so by sending Transact-SQL statements to the server, regardless of the user interface of the application.

Stored procedures in SQL Server are executable server-side routines. The advantage of stored procedures is the ability to pass parameters.

5. REACT:

React (also known as React.js or ReactJS) is an open-source, front end, JavaScript library for building user interfaces or UI components. 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. However, React is only concerned with state management and rendering that state to the DOM, so creating React applications usually requires the use of additional libraries for routing, as well as certain client-side functionality.

Problem Statement

We define social network sites as web - based services that allow individuals to construct a public or semi - public profile within a bounded system , articulate a list of other users with whom they share a connection and view and traverse their list of connections and those made by others in a system .

Social networking sites are not only for you to communicate or interact with other people globally but, this is also one effective way for business promotion. A lot of business minded people these days are doing business online and use these social networking sites to respond to customer queries. It isn't just a social media site used to socialize with your friend but also , represents a huge pool of information from day to day living.

Aim & Objective: -

A complete Social Networking website will be developed based on Semantic Web Service - oriented model which will have following features.

- To have attractive and Secure Login page to access
- Make new user account in more user friendly and proper validation of details
- Search People easily on entire network
- Creating a public profile having social, professional and personal information
- Ease of editing of profile anytime
- Trust and authentication of user by viewing his profile
- Chat with Online Friends
- Upload and Share Images on network
- Administration page to keep eye on user operation
- Easily password recovery processing

Registered Users can create their profile which will contain their details such as interests etc.

- Login, Signup, Quick Login (this means app remembers that you had logged in, so you only have to enter password to login again).
- AJAX used almost everywhere (From login to logout).
- Follow, Unfollow, Recommend, Profile views, Block, Unblock.
- Like, Comment (Sticker, Text, Image), Share, remove share,
- Tag, remove tag, copy post link, delete post, Edit post, Open post.
- Post Text, Image (with filters), Video, Audio, Link, Document,
- Location with emojis, Font size, Tag, mentions, hashtags and location.
- Edit your profile (From username, Bio, social links to tags).
- Explore people, photos, videos, audios and groups.
- Message emoji, Text, Image, Sticker.
- Delete conversation, unsend all your messages, edit conversation
- Title and get all info about the conversation.
- Create a group with your followings.
- Add/remove members.
- Edit group's settings.
- Change your password.
- Change account type private or public (Default: public).
- Change email and phone visibility.
- Get all your login details

Scope of the Project: -

Social Networking site is an online community designed to make social life of people more active and stimulating. The social network can help you to maintain existing relationships with peoples and share pictures and messages, and establish new ones by reaching out to people you've never met before.

This website also provides the features of E- classified all at one place. The main idea behind site is to share your thoughts with all your friends which can be read by all the users using the website. This site can be handled by the user as he wants for example adding videos and photos also. This website enhances Advertisement of products. People using this website can buy sell products from this site . The main function behind this advertisement function is to help people to buy products which are trusted in a circle.

Modules:

The entire project mainly consists of 4 modules, which are-

- 1) Admin
- 2) Login
- 3) User
- 4) Instant Chat

1. Admin module:

- Signup their account.
- Login (No approval required)
- Can register / view / approve / reject (approve those user who request access)
- Can operate whole app significantly
- Admin can handle all access regarding the users
- Can delete post if find anything irrelevant
- Can delete user account if doing anything suspicious

2. Login:

- Can login or signup through email or phone number
- Can set the username
- Can set the password

3. User:

- Access account by entering id and password
- Can set their profile picture
- Can check other people profiles and news feed
- Can follow the social influencers , their friends ,pages etc
- Can block other user if misbehaving
- Can private their account

4. Instant Chat:

- Can message someone directly if they know the other user
- Can send images directly
- Can send other peoples feed or picture though message

- Manage own profile

5. Pharmacist module:

- Maintain medicine
- Keep records of hospitals stock medicines and status
- Manage medicine categories
- Watch prescription of patient
- Provide medication to prescriptions

6. Laboratory's module:

- Watch prescription list
- Upload diagnostic report
- Preview of report files. like Xray images, CT scan, MRI reports
- Manage own profile

7. Accountant module:

- Create invoice for payment
- Order invoice to patient
- Take cash payment
- Watch payment history of patients
- Manage own profile

Completed Modules:

Admin

- Signup their account. Then Login (No approval Required).
- Can register/view/approve/reject/delete user if misbehaving or doing any suspicious thing.
- Can operate whole site significantly.
- Admin can handle all concerns regarding the users
- Can verify users account if the account is genuine or any public celebrity.

User

- Can set the username ,password and login to the site.
- Can set the profile picture.
- Follow each other people account by searching by their username or name then follow them.
- Can message any other user directly .
- Get the notification on the site if someone followed them or send them a message.
- Block any other user if they are misbehaving with them.
- Can set whether to make there account public or private so that normal peoples can view their uploaded pictures.

Drawbacks / Loopholes

- Payment Module can also add so that users can pay each other via site only.
- Voice & Video call feature can also be there in the site.
- A global Search module can be inserted so that all the popular feeds or the feeds similar to that u liked showed there.

Design

INTRODUCTION TO UML: UML Design

The Unified Modelling Language (UML) is a standard language for specifying, visualizing, constructing, and documenting the software system and its components. It is a graphical language, which provides a vocabulary and set of semantics and rules. The UML focuses on the conceptual and physical representation of the system. It captures the decisions and understandings about systems that must be constructed. It is used to understand, design, configure, maintain, and control information about the systems.

The UML is a language for: Visualizing Specifying Constructing Documenting

Visualizing

Through UML we see or visualize an existing system and ultimately, we visualize how the system is going to be after implementation. Unless we think, we cannot implement. UML helps to visualize, how the components of the system communicate and interact with each other.

Specifying

Specifying means building, models that are precise, unambiguous and complete UML addresses the specification of all the important analysis design, implementation decisions that must be made in developing and deploying a software system.

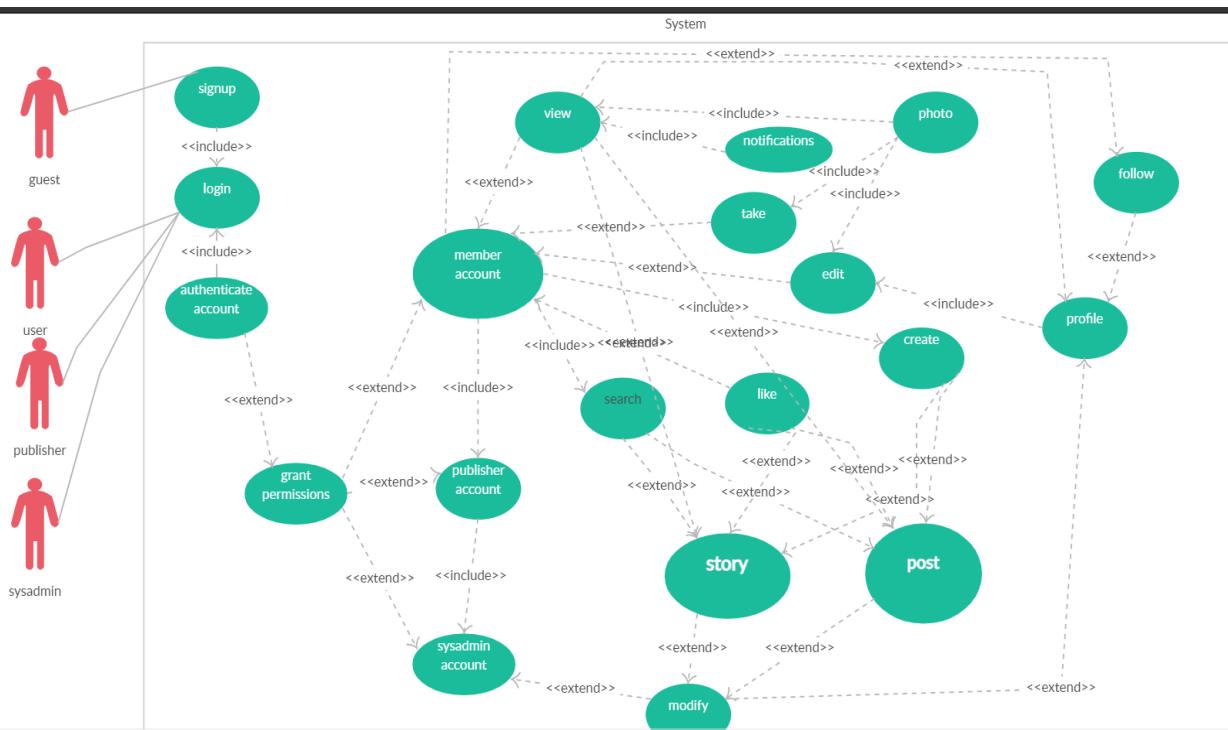
Constructing

UML models can be directly connected to a variety of programming language through mapping a model from UML to a programming language like JAVA or C++ or VB. Forward Engineering and Reverse Engineering is possible through UML.

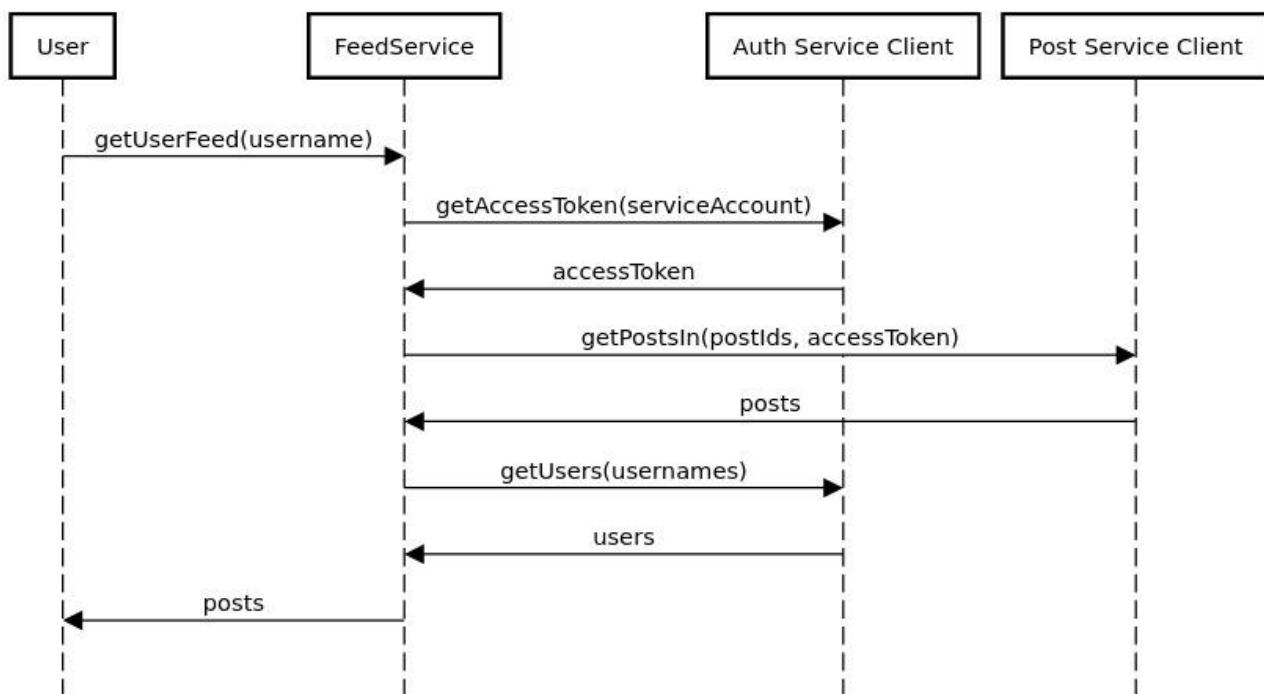
Documenting

The Deliverables of a project apart from coding are some Artifacts, which are critical in controlling, measuring and communicating about a system during its developing requirements, architecture, desire, source code, project plans, tests, prototypes releasers, etc...

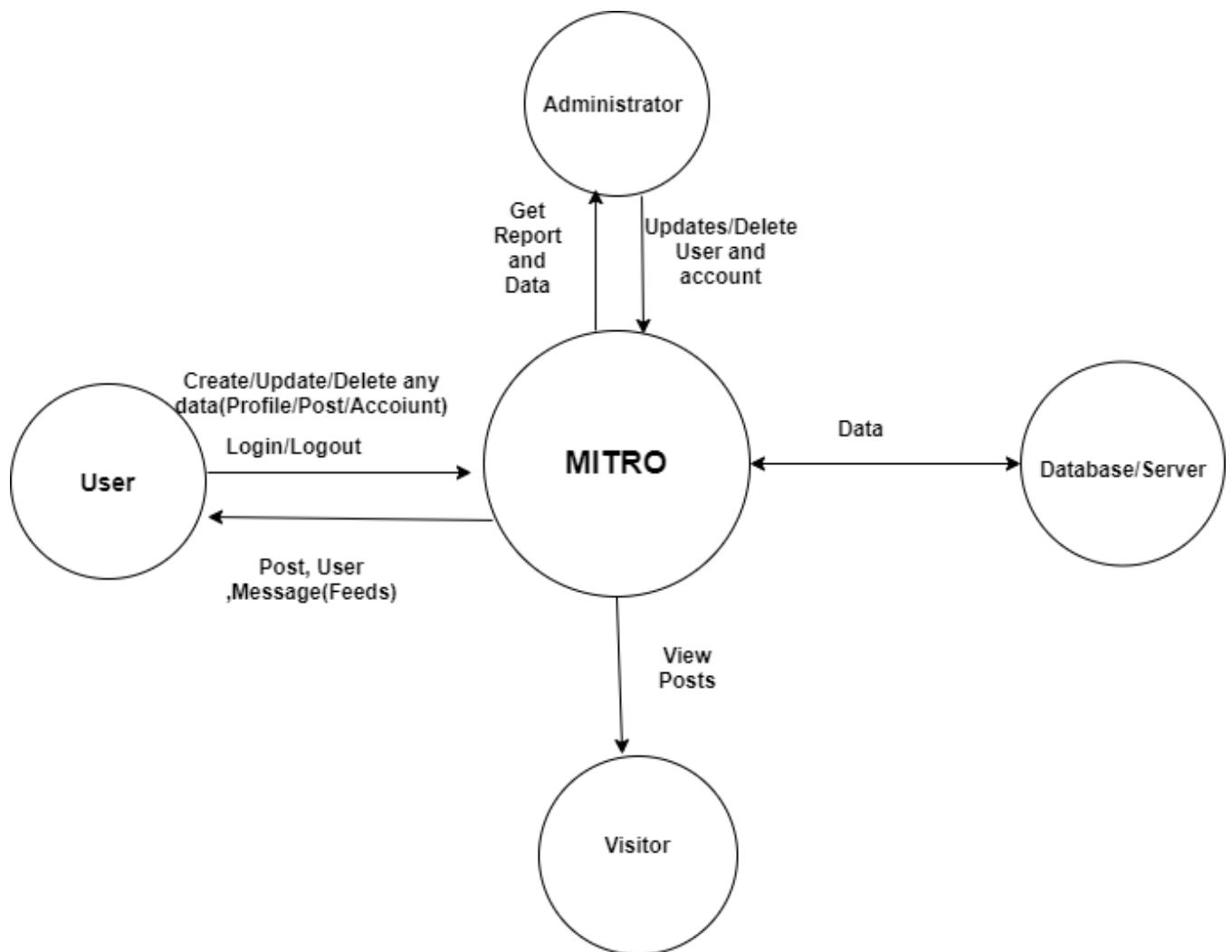
1. Use Case Diagram



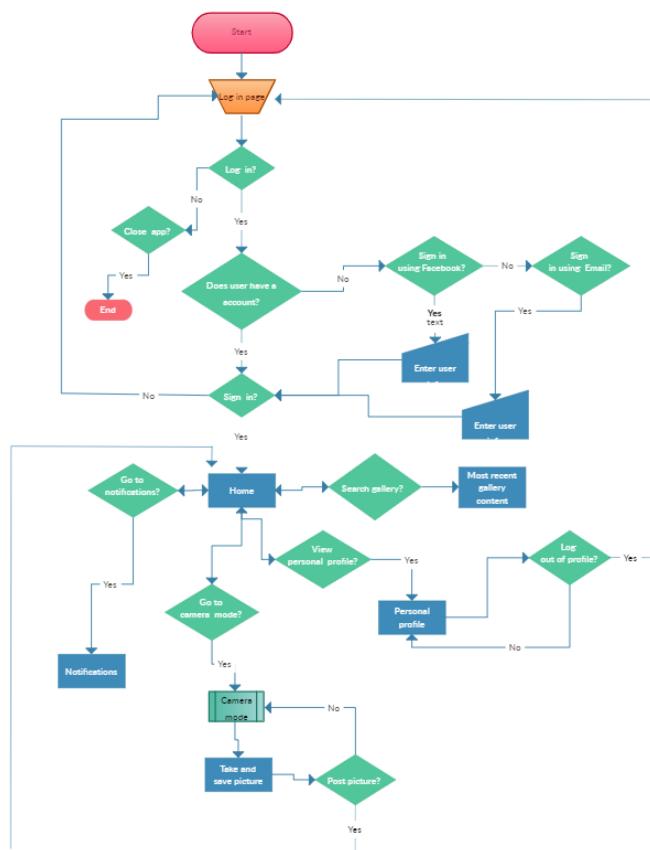
2. Sequence Diagram



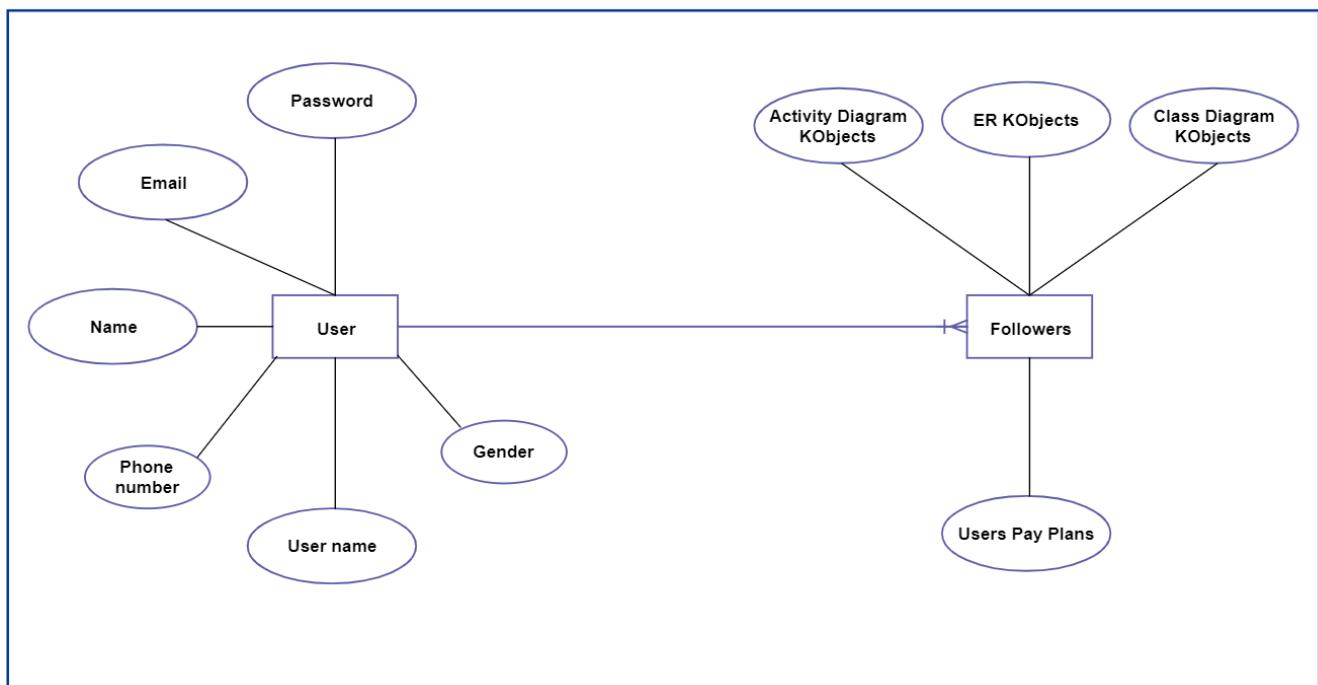
3. 0-Level DFD



4. 1-Level DFD



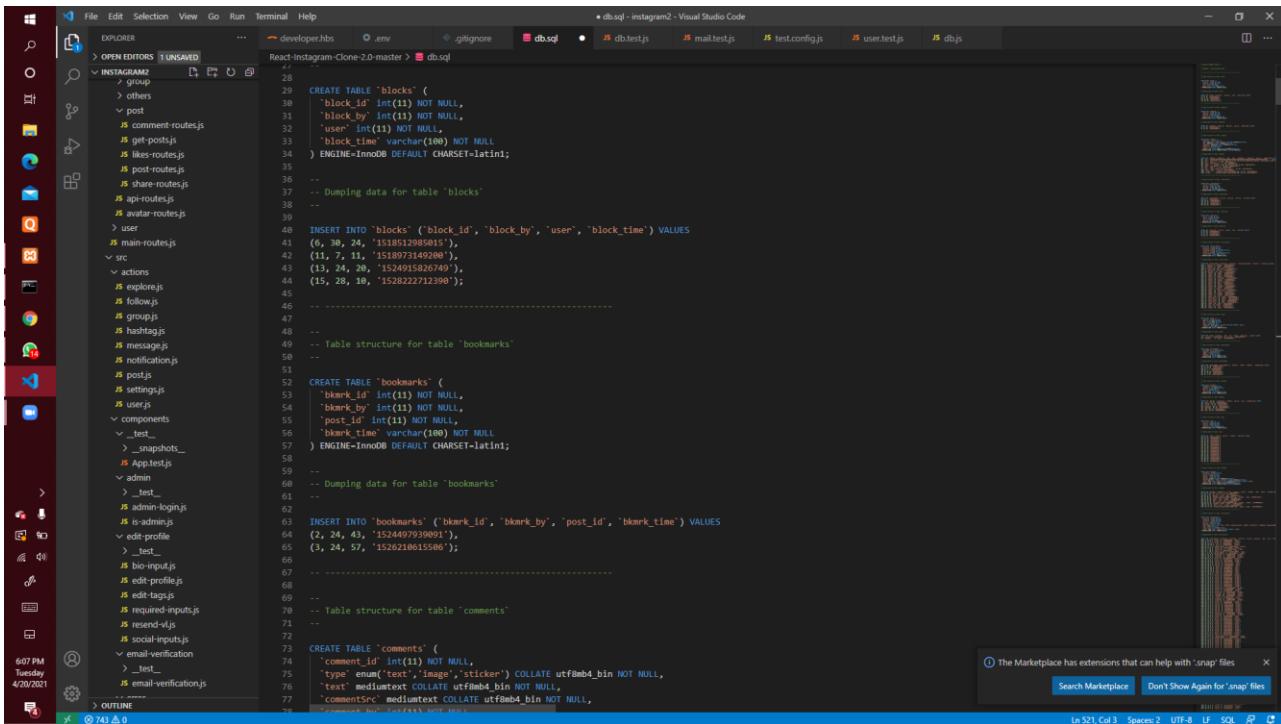
5. ER Diagram



Progress till Date & The Remaining work

Project Ideation	Done
Synopsis	Done
Technologies to Learn	Done
UML Diagram	Done
Mid Term Report	Done
Coding	Done
Testing and Implementation	Done
Final Report	Done

Coding



```

CREATE TABLE `blocks` (
  `block_id` int(11) NOT NULL,
  `block_by` int(11) NOT NULL,
  `user` int(11) NOT NULL,
  `block_time` varchar(100) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

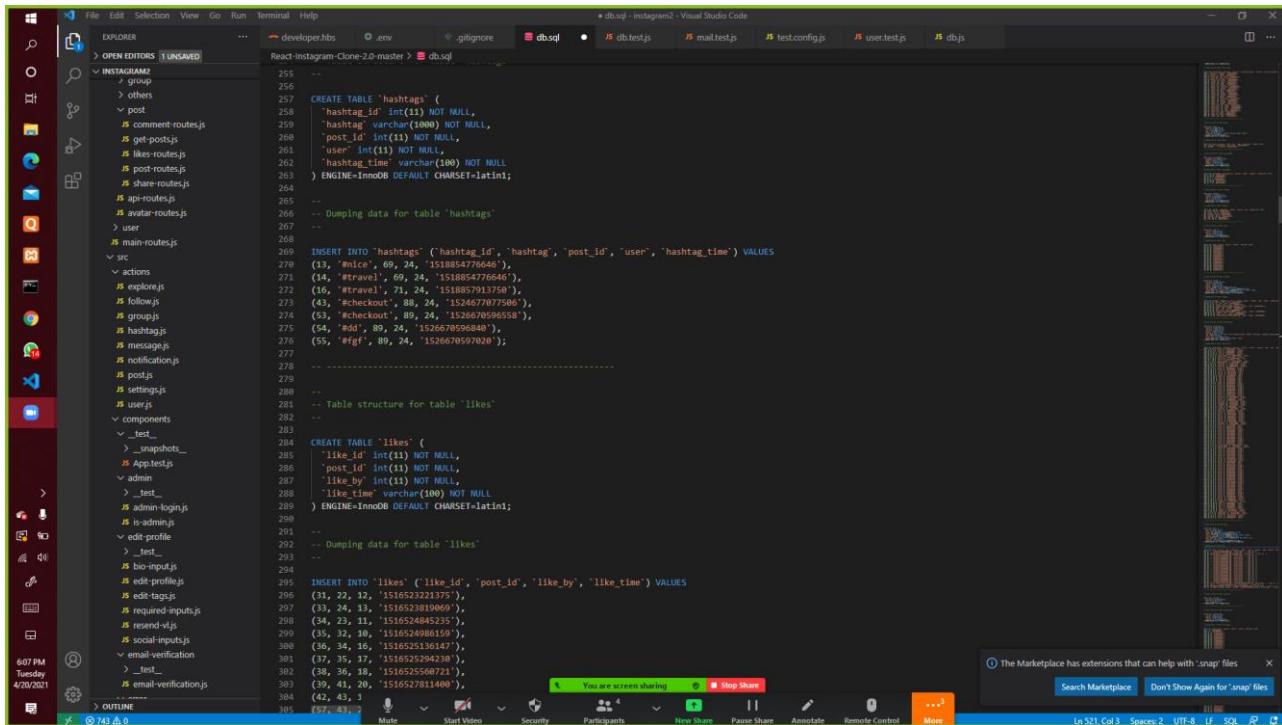
-- Dumping data for table `blocks`
INSERT INTO `blocks` (`block_id`, `block_by`, `user`, `block_time`) VALUES
(6, 30, 24, '1518512985015'),
(11, 7, 11, '1518973149200'),
(13, 24, 26, '1524915826749'),
(15, 28, 10, '1528222712398');

-- Table structure for table 'bookmarks'
CREATE TABLE `bookmarks` (
  `bmark_id` int(11) NOT NULL,
  `bmark_by` int(11) NOT NULL,
  `post_id` int(11) NOT NULL,
  `bmark_time` varchar(100) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- Dumping data for table 'bookmarks'
INSERT INTO `bookmarks` (`bmark_id`, `bmark_by`, `post_id`, `bmark_time`) VALUES
(2, 24, 43, '1524497939091'),
(3, 24, 57, '1526218615586');

-- Table structure For table 'comments'
CREATE TABLE `comments` (
  `comment_id` int(11) NOT NULL,
  `type` enum('text','image','sticker') COLLATE utf8mb4_bin NOT NULL,
  `text` mediumtext COLLATE utf8mb4_bin NOT NULL,
  `commentSrc` mediumtext COLLATE utf8mb4_bin NOT NULL,
  `commentTime` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```



```

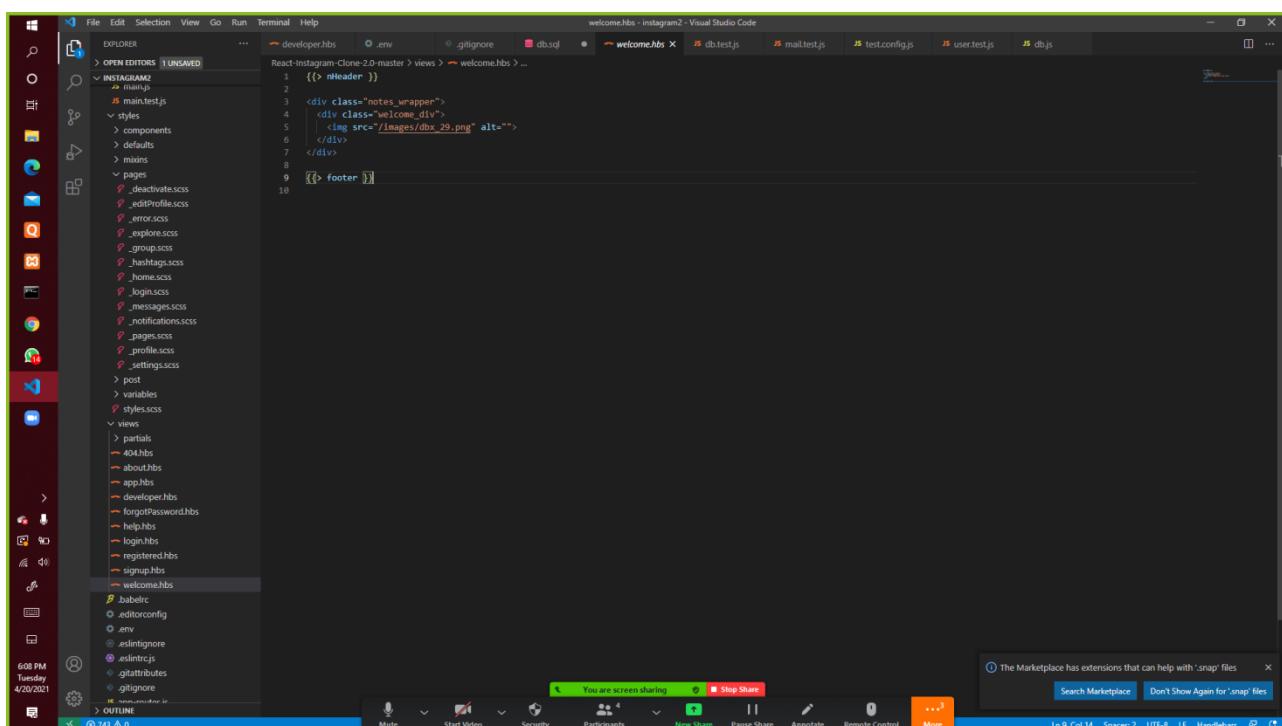
CREATE TABLE `hashtags` (
    `hashtag_id` int(11) NOT NULL,
    `hashtag` varchar(2000) NOT NULL,
    `post_id` int(11) NOT NULL,
    `user` int(11) NOT NULL,
    `hashtag_time` varchar(100) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

INSERT INTO `hashtags`(`hashtag_id`, `hashtag`, `post_id`, `user`, `hashtag_time`)
VALUES
(13, '#ice', 69, 24, '1518854776646'),
(14, '#travel', 69, 24, '1518854776646'),
(16, '#travel', 71, 24, '1518857913750'),
(43, '#checkout', 88, 24, '1524677077586'),
(53, '#checkout', 89, 24, '1526670596558'),
(54, '#dd', 89, 24, '1526670596580'),
(55, '#gf', 89, 24, '1526670597920');

CREATE TABLE `likes` (
    `like_id` int(11) NOT NULL,
    `post_id` int(11) NOT NULL,
    `like_user` int(11) NOT NULL,
    `like_time` varchar(100) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

INSERT INTO `likes`(`like_id`, `post_id`, `like_by`, `like_time`)
VALUES
(31, 22, 12, '1518522321275'),
(32, 22, 12, '1518522321275'),
(29, 23, 11, '1518524845235'),
(35, 32, 10, '1518524806159'),
(30, 34, 16, '151852536147'),
(37, 35, 17, '1518525294230'),
(38, 36, 18, '1518525560721'),
(39, 41, 20, '1518527811400'),
(42, 43, 1, '1518527811400');

```



```

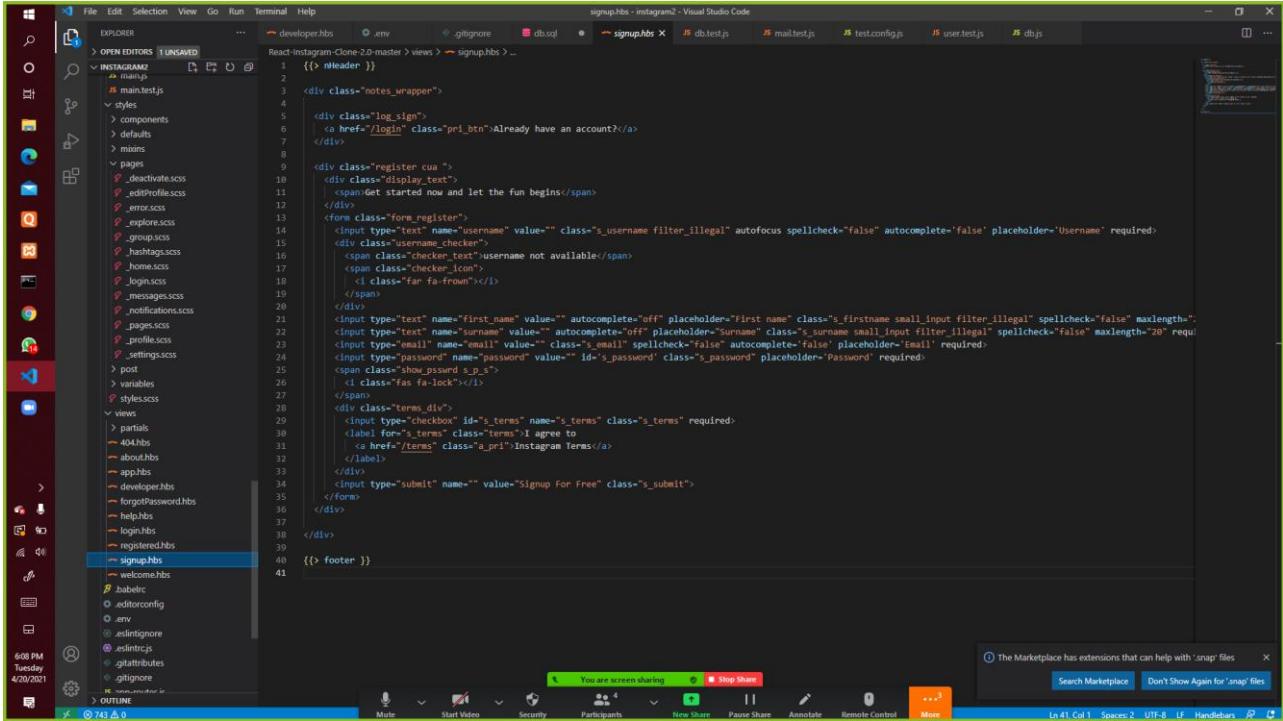
{{> 'notes_wrapper' }}





{{> 'footer' }}

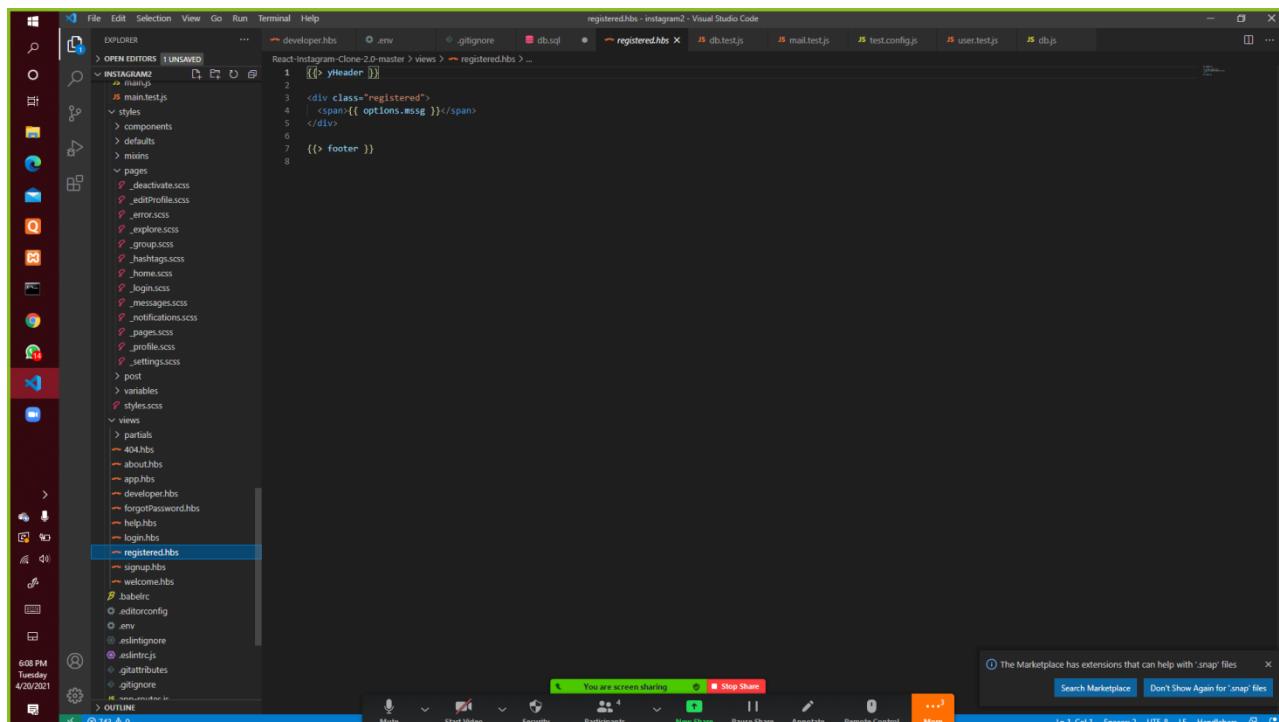
```



```

1  {{> header }}
2
3  <div class="notes_wrapper">
4
5    <div class="log_sign">
6      <a href="/login" class="pri_btn">Already have an account?</a>
7    </div>
8
9    <div class="register cum_a">
10       <div class="display_text">
11         <span>Get started now and let the fun begins</span>
12       </div>
13       <form class="form_register">
14         <input type="text" name="username" value="" class="s_username filter_illegal" autofocus spellcheck="false" autocomplete="false" placeholder="Username" required>
15         <div class="username_checker">
16           <span class="checker_text">username not available:</span>
17           <span class="checker_icon">
18             <i class="far fa-frown"></i>
19           </span>
20         </div>
21
22         <input type="text" name="firstname" value="" autocomplete="off" placeholder="First name" class="s_firstname small_input filter_illegal" spellcheck="false" maxlength="20" required>
23         <input type="text" name="surname" value="" autocomplete="off" placeholder="Surname" class="s_surname small_input filter_illegal" spellcheck="false" maxlength="20" required>
24         <input type="email" name="email" value="" class="s_email" spellcheck="false" autocomplete="false" placeholder="Email" required>
25         <input type="password" name="password" value="" id="s_password" class="s_password" placeholder="Password" required>
26         <span class="show_password_s p_s">
27           <i class="fas fa-lock"></i>
28         </span>
29         <div class="terms_div">
30           <input type="checkbox" id="s_terms" name="s_terms" class="s_terms" required>
31           <label for="s_terms" class="terms">I agree to<br/>
32             <a href="#terms" class="a_pri">Instagram Terms</a>
33           </label>
34         </div>
35         <input type="submit" name="" value="Signup For Free" class="s_submit">
36       </form>
37     </div>
38   </div>
39
40 {{> footer }}
41

```

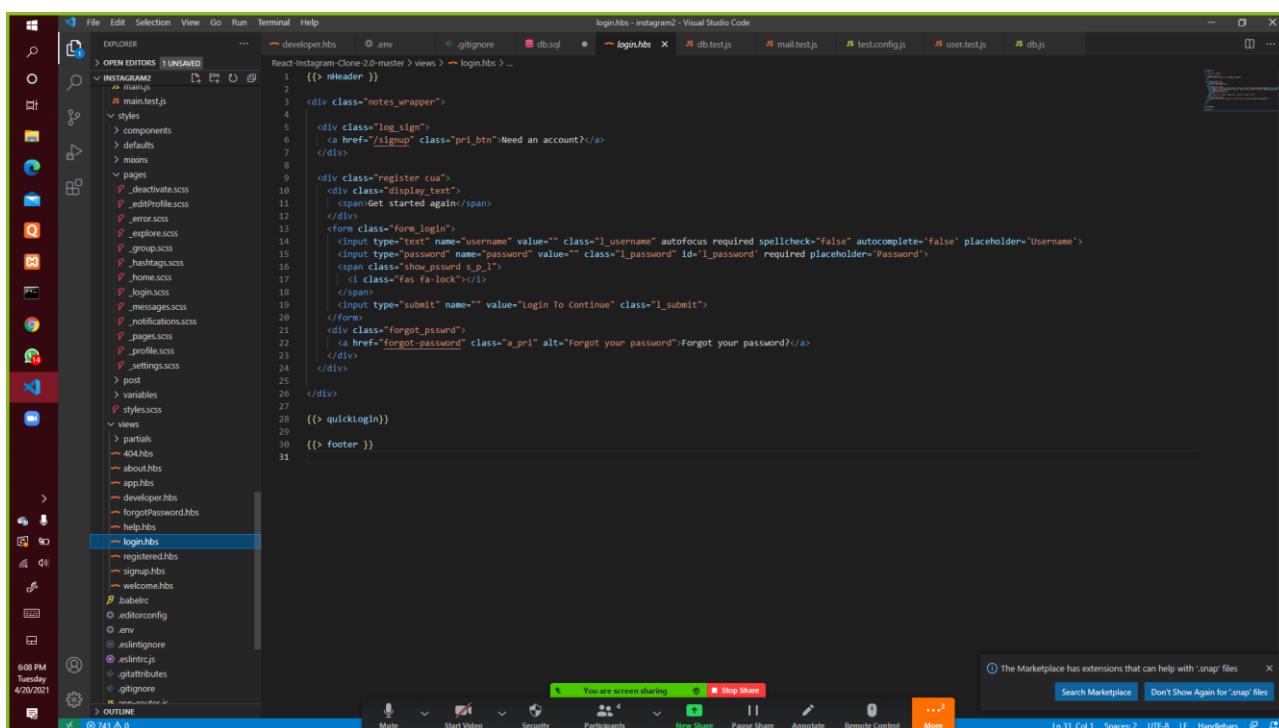


```

1 <div class="yHeader">
2   <div>
3     <div class="registered">
4       | <span>{{ options.msg }}</span>
5     </div>
6   </div>
7   {{> footer }}
8

```

The screenshot shows the Visual Studio Code interface with the registered.hbs file open in the center. The sidebar on the left shows the project structure for 'INSTAGRAM2' with various files like main.js, styles.css, and pages. The bottom status bar indicates the file is 140 lines long.



```

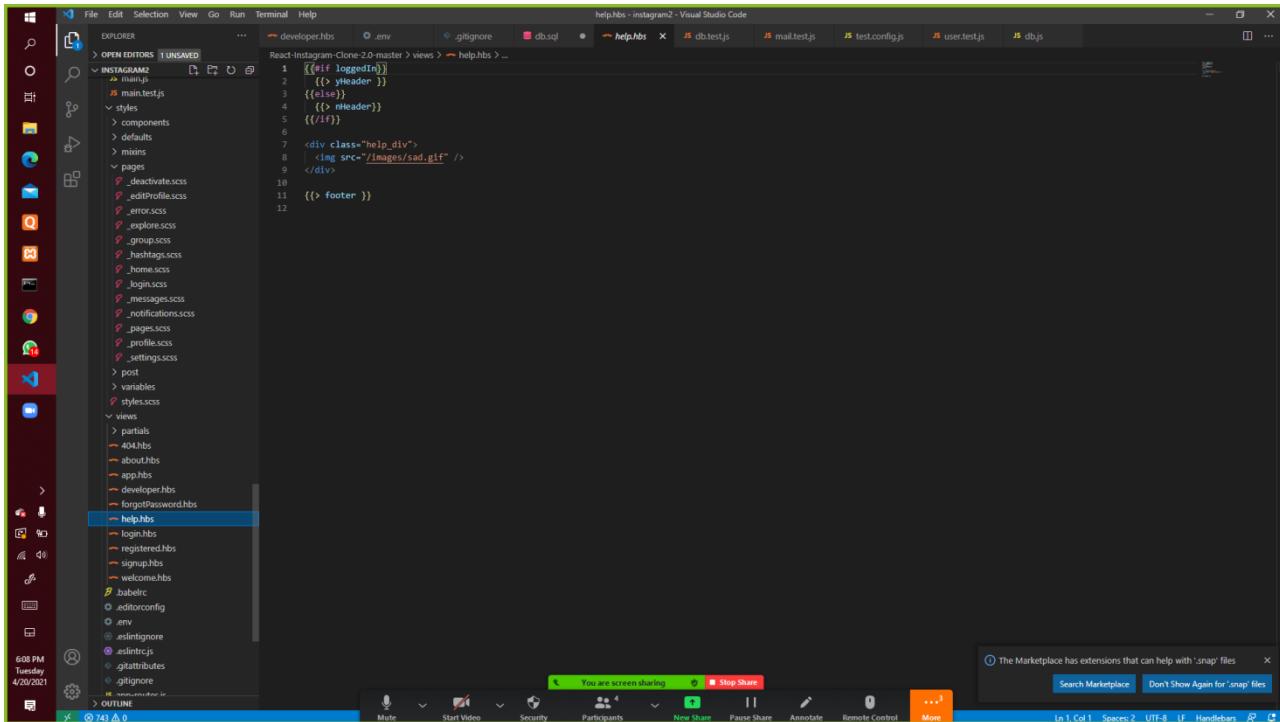
1 <> notes_wrapper>
2   <div class="log_sign">
3     | <a href="/signup" class="pri_btn">Need an account?</a>
4   </div>
5
6   <div class="register cuu">
7     <div class="display_text">
8       <span>Get started again</span>
9     </div>
10    <form class="form_login">
11      <input type="text" name="username" value="" class="l_username" autofocus required spellcheck="false" autocomplete="false" placeholder="Username">
12      <input type="password" name="password" value="" class="l_password" id="l_password" required placeholder="Password">
13      <span class="show_pswrd s_p_1">
14        | <i class="fas fa-lock"></i>
15      </span>
16      <input type="submit" name="" value="Login To Continue" class="l_submit">
17    </form>
18    <div class="forgot_password">
19      <a href="/forgot_password" class="a_pei" alt="Forgot your password?>Forgot your password?</a>
20    </div>
21  </div>
22  {{> quickLogin}}
23
24  {{> footer }}
25
26
27
28
29
30
31

```

The screenshot shows the Visual Studio Code interface with the login.hbs file open. The code defines a notes wrapper section with a log_sign class containing a link to /signup. It then defines a register class with a form_login section containing input fields for username and password, and a submit button. Below this is a forgot_password section with a link to /forgot_password. The bottom status bar indicates the file is 743 lines long.

Third Year Mini Project Report

B.Tech. (CSE)-Session 2020-2021



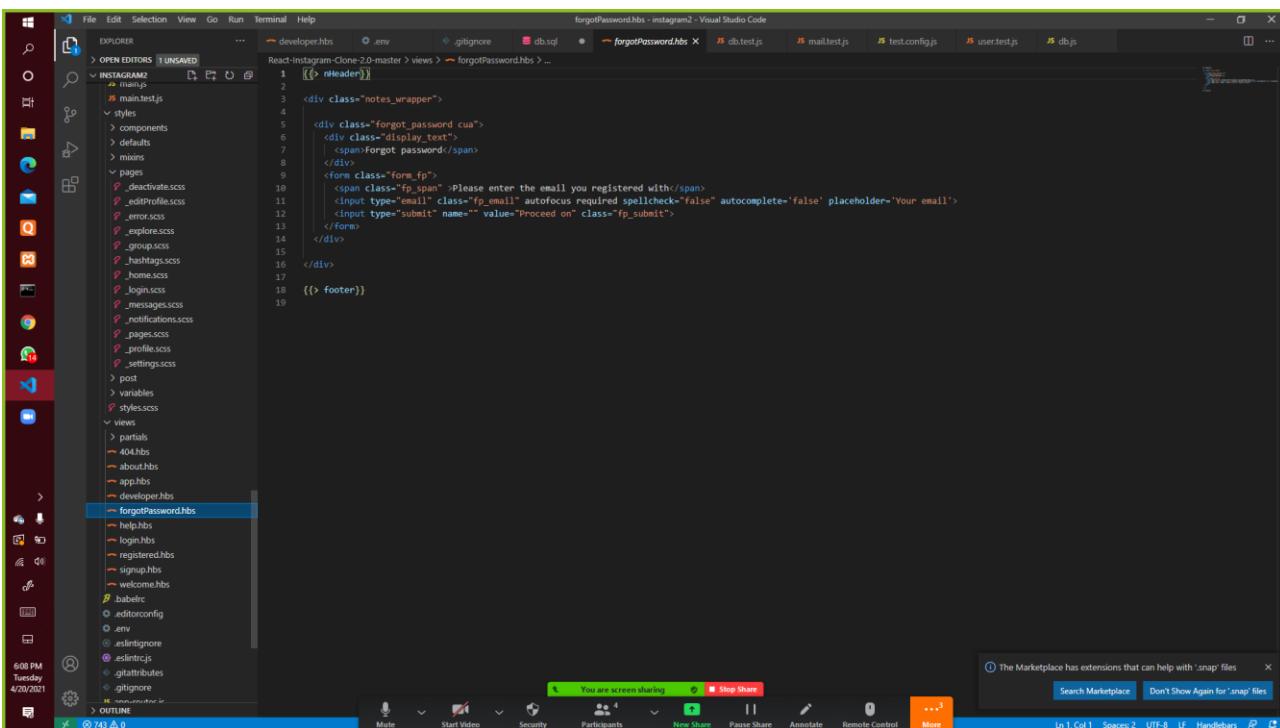
The screenshot shows the Visual Studio Code interface with the 'help.hbs' file open in the center editor pane. The code is a HBS template for a help page:

```

1  {{#if loggedin}}
2    {{> header}}
3  {{else}}
4    {{> header}}
5  {{/if}}
6
7  <div class="help_div">
8    | 

The left sidebar shows the project structure for 'React-Instagram-Clone-2.0-master' with files like main.test.js, styles.css, components, defaults, mixins, pages, deactivate.scss, editProfile.scss, error.scss, explore.scss, group.scss, hashtags.scss, home.scss, login.scss, messages.scss, notifications.scss, pages.scss, profile.scss, settings.scss, post, variables, and views.


```



The screenshot shows the Visual Studio Code interface with the 'forgotPassword.hbs' file open in the center editor pane. The code is a HBS template for a forgot password page:

```

1  {{> header}}
2
3  <div class="notes_wrapper">
4
5    <div class="forgot_password_cua">
6      <div class="display_text">
7        | <span>Forgot password</span>
8      </div>
9      <form class="fp_form">
10        <span class="fp_span">Please enter the email you registered with:</span>
11        <input type="email" class="fp_email" autofocus required spellcheck="false" autocomplete="off" placeholder="Your email">
12        <input type="submit" name="" value="Proceed on" class="fp_submit">
13      </form>
14    </div>
15  </div>
16
17 {{> footer}}
18
19

```

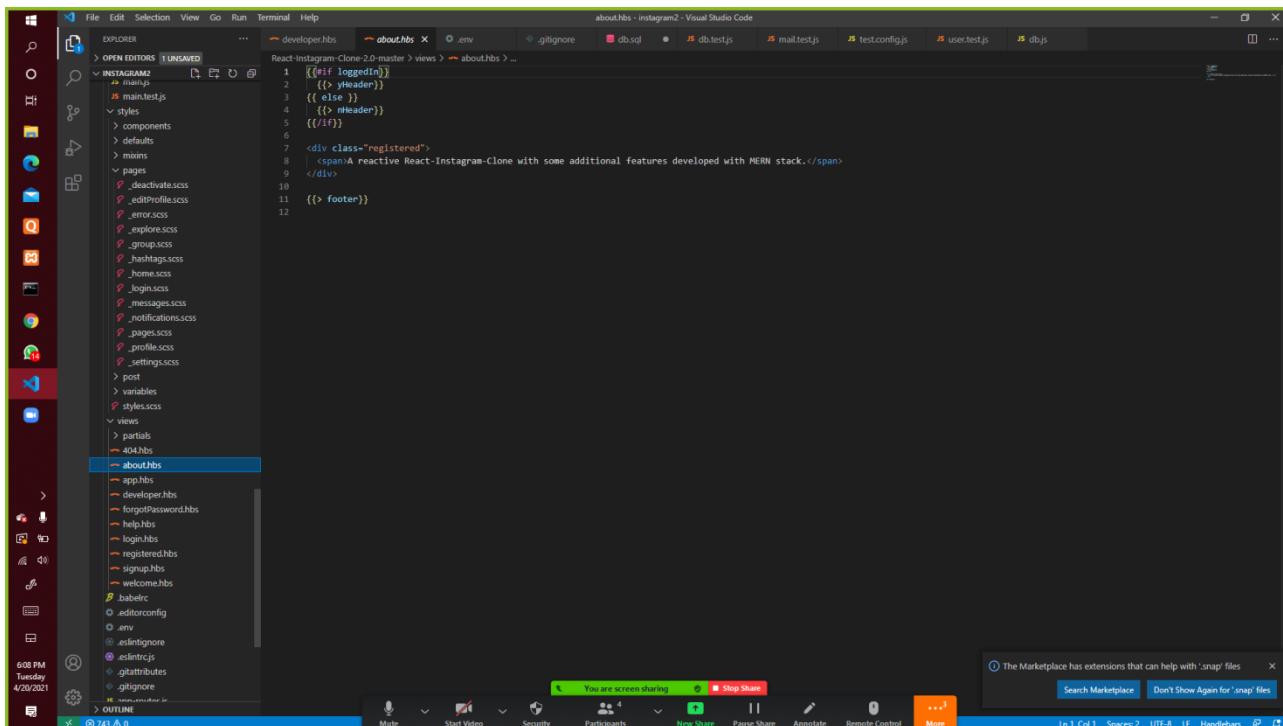
The left sidebar shows the project structure for 'React-Instagram-Clone-2.0-master' with files like main.test.js, styles.css, components, defaults, mixins, pages, deactivate.scss, editProfile.scss, error.scss, explore.scss, group.scss, hashtags.scss, home.scss, login.scss, messages.scss, notifications.scss, pages.scss, profile.scss, settings.scss, post, variables, and views.



A screenshot of Microsoft Visual Studio Code (VS Code) interface. The title bar shows "developer.hbs - Instagram2 - Visual Studio Code". The left sidebar (Explorer) lists project files: ".env", ".gitignore", "db.js", "forgotPassword.hbs", "db.test.js", "mailtest.js", "test.config.js", "user.test.js", and "db.js". Below these are "INSTAGRAM2" and "main.test.js" with a "UNSAVED" indicator. Under "INSTAGRAM2", there are "styles", "components", "defaults", "mixins", "pages" (containing deactivate.scss, editProfile.scss, error.scss, explore.scss, group.scss, hashtags.scss, home.scss, login.scss, messages.scss, notifications.scss, pages.scss, profile.scss, settings.scss), "post", "variables", "styles.css", "views", "partials" (containing 404.hbs, about.hbs, app.hbs), and "developer.hbs" (which is currently selected). The main editor area displays the "developer.hbs" file content, which includes logic for header rendering based on user status and a section for displaying developer profiles. The bottom status bar shows "608 PM Tuesday 4/20/2021" and "You are screen sharing". A floating message in the bottom right corner says "The Marketplace has extensions that can help with 'snap' files".

Third Year Mini Project Report

B.Tech. (CSE)-Session 2020-2021



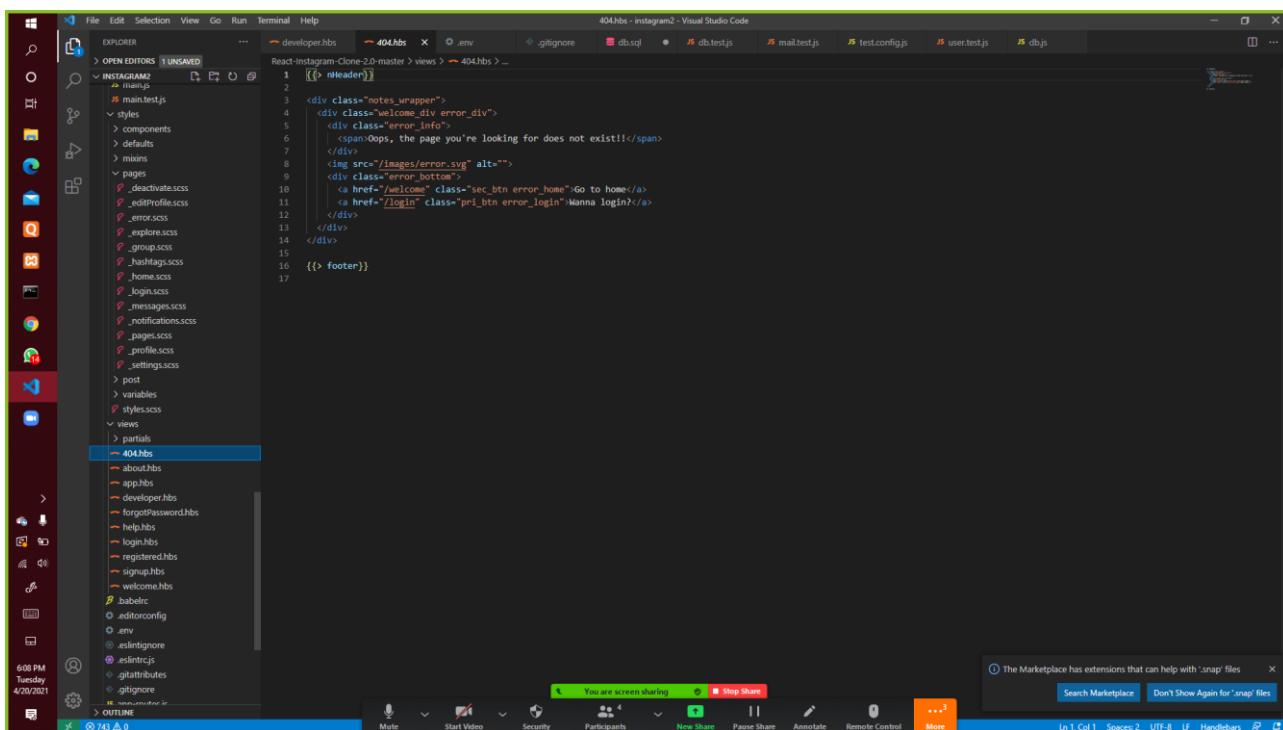
The screenshot shows the Visual Studio Code interface with the file `about.hbs` open in the center. The code is as follows:

```

1  {{#if loggedIn}}
2    {{> yHeader}}
3    {{> mHeader}}
4    {{> nHeader}}
5  {{/if}}
6
7  <div class="registered">
8    | <span>A reactive React-Instagram-Clone with some additional features developed with MERN stack.</span>
9  </div>
10
11 {{> footer}}
12

```

The left sidebar shows the project structure under `INSTAGRAM2`, including `main.js`, `styles`, `components`, `mixins`, `pages`, `partials`, and `views`. The `views` folder contains files like `404.hbs`, `about.hbs`, `app.hbs`, etc.



The screenshot shows the Visual Studio Code interface with the file `404.hbs` open in the center. The code is as follows:

```

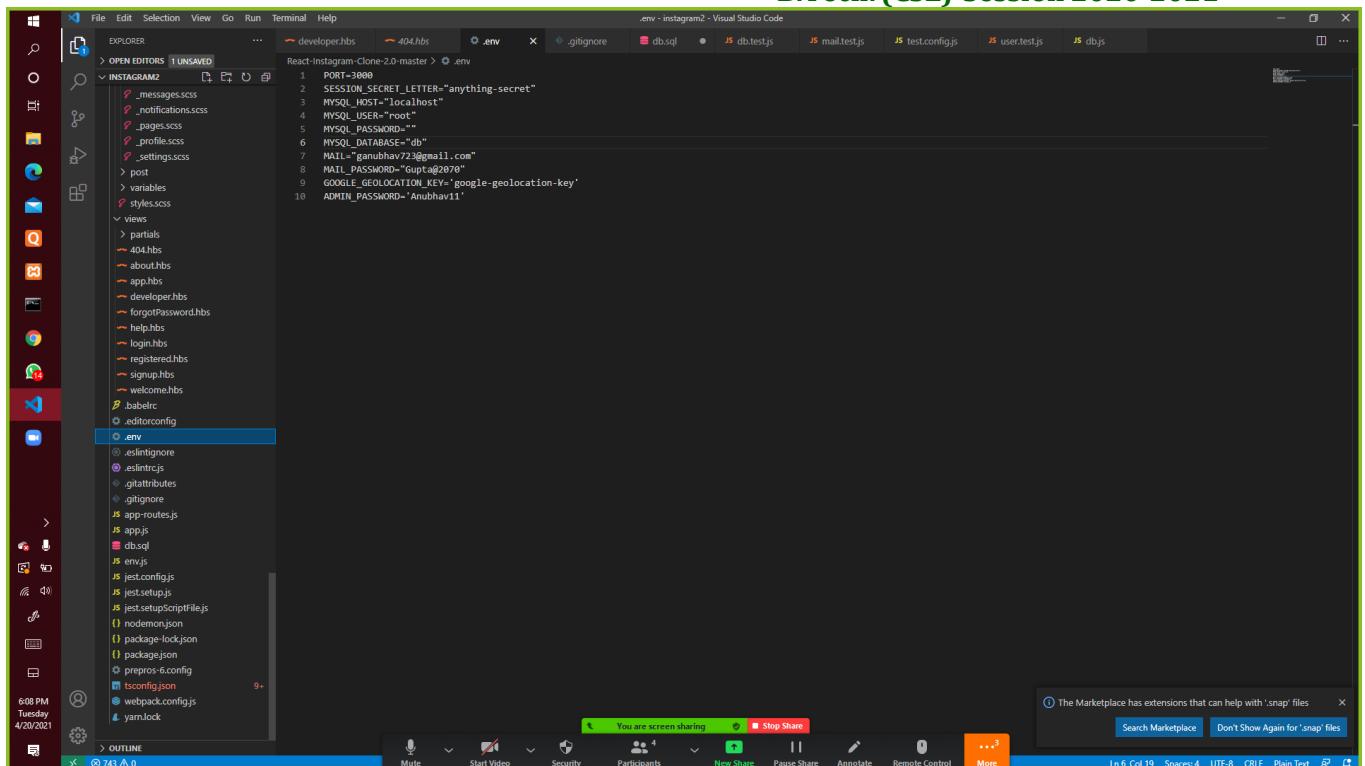
1  {{#if !header}}
2
3  <div class="notes_wrapper">
4    <div class="welcome_div error_div">
5      <div class="error_info">
6        Ooops, the page you're looking for does not exist!</span>
7      </div>
8      
9      <div class="error_bottom">
10         | <a href="/" class="sec_btn error_home">Go to home</a>
11         | <a href="/login" class="pri_btn error_login">Wanna login?</a>
12     </div>
13   </div>
14 </div>
15
16 {{> footer}}
17

```

The left sidebar shows the project structure under `INSTAGRAM2`, including `main.js`, `styles`, `components`, `mixins`, `pages`, `partials`, and `views`. The `views` folder contains files like `404.hbs`, `about.hbs`, `app.hbs`, etc.

Third Year Mini Project Report

B.Tech. (CSE)-Session 2020-2021



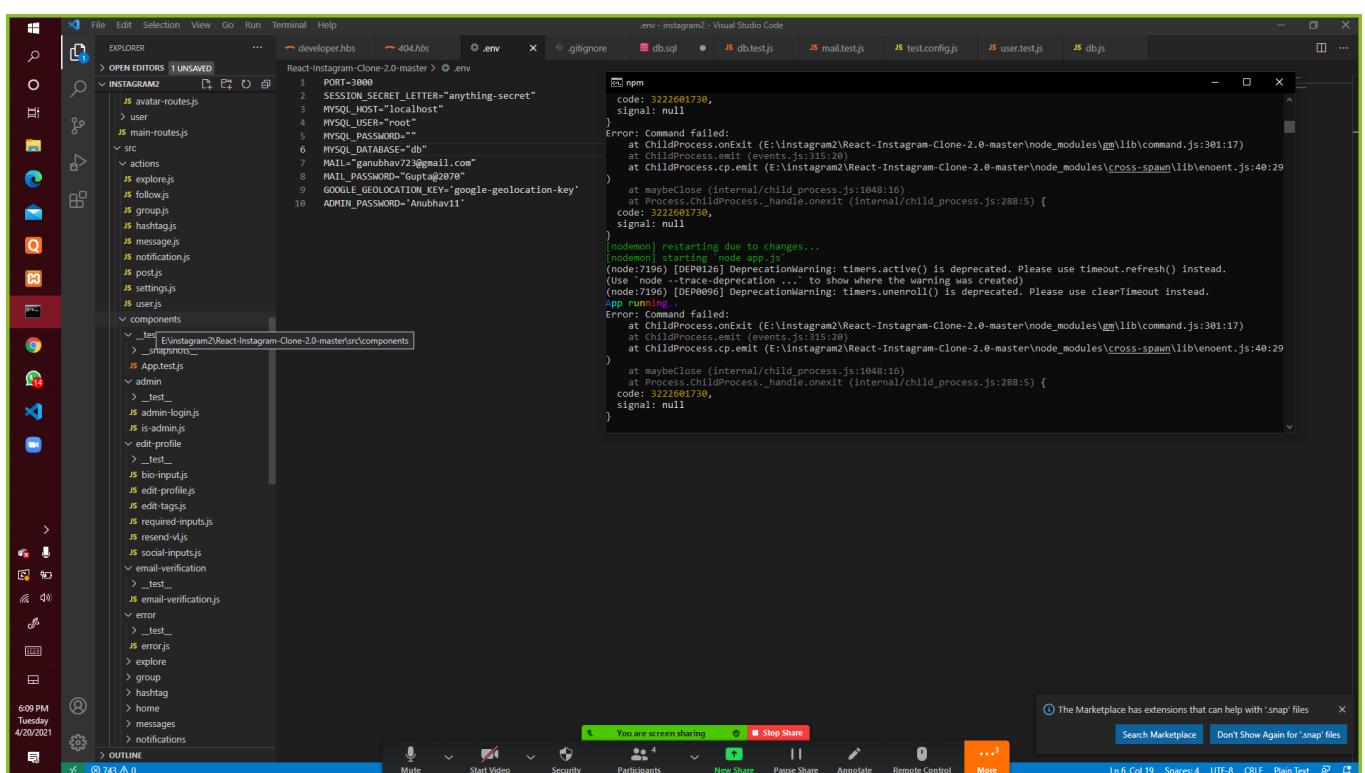
The screenshot shows the Visual Studio Code interface with the following details:

- Title Bar:** .env - Instagram2 - Visual Studio Code
- File Explorer:** Shows the project structure for "INSTAGRAM2" with files like developer.hbs, 404.hbs, .env, .gitignore, db.sql, db.test.js, test.config.js, user.test.js, and dbjs.
- Code Editor:** Displays the content of the ".env" file:

```

1 PORT=3000
2 SESSION_SECRET=LETTER="anything-secret"
3 MYSQL_HOST="localhost"
4 MYSQL_USER="root"
5 MYSQL_PASSWORD=""
6 MYSQL_DATABASE="db"
7 MAIL="ganubhav72@gmail.com"
8 MAIL_PASSWORD="Gupta@2070"
9 GOOGLE_GEOLOCATION_KEY="google-geolocation-key"
10 ADMIN_PASSWORD="Anubhav11"

```
- Bottom Status Bar:** Shows the file path (React-Instagram-Clone-2.0-master > .env), line 6, column 19, spaces 4, and encoding as UTF-8.
- Bottom Taskbar:** Includes icons for Mute, Start Video, Security, Participants, New Share, Pause Share, Annotate, Remote Control, and More.
- Bottom Right:** A message box: "The Marketplace has extensions that can help with '.snap' files" with options to "Search Marketplace" or "Don't Show Again for '.snap' files".



This screenshot is similar to the first one but includes a terminal window on the right side displaying the command-line output of the node application:

```

$ npm
  code: 3222801730,
  signal: null
}
Error: Command failed:
at ChildProcess._exit (E:\Instagram2\React-Instagram-Clone-2.0-master\node_modules\pm\lib\command.js:301:17)
at ChildProcess.emit (events.js:315:20)
at ChildProcess.cp.emit (E:\Instagram2\React-Instagram-Clone-2.0-master\node_modules\cross-spawn\lib\enoent.js:40:29
) at maybeClose (internal/child_process.js:1048:16)
at Process.ChildProcess._handle.onexit (internal/child_process.js:288:5) {
  code: 3222801730,
  signal: null
}
[nodemod] restarting due to changes...
[nodemod] starting node app.js
(node:7196) [DEP0026] DeprecationWarning: timers.active() is deprecated. Please use timeout.refresh() instead.
(node:7196) [DEP0096] DeprecationWarning: timers.unenroll() is deprecated. Please use clearTimeout instead.
App running.
Error: Command failed:
at ChildProcess._exit (E:\Instagram2\React-Instagram-Clone-2.0-master\node_modules\pm\lib\command.js:301:17)
at ChildProcess.emit (events.js:315:20)
at ChildProcess.cp.emit (E:\Instagram2\React-Instagram-Clone-2.0-master\node_modules\cross-spawn\lib\enoent.js:40:29
) at maybeClose (internal/child_process.js:1048:16)
at Process.ChildProcess._handle.onexit (internal/child_process.js:288:5) {
  code: 3222801730,
  signal: null
}

```

Third Year Mini Project Report

B.Tech. (CSE)-Session 2020-2021

```

user.js - instagram2 - Visual Studio Code
File Edit Selection View Go Run Terminal Help
developer.hbs .env JS user.js .gitignore db.sql JS db.test.js JS mail.test.js JS test.config.js JS user.test.js JS dbjs
React-Instagram-Clone-2.0-master > src > actions > JS user.js > ...
1 import { dispatchHelper } from '../utils/utils'
2
3 export const getUserDetails = username =>
4   dispatchHelper('GET_USER_DETAILS', 'get-user-details', { username })
5
6 export const deleteTag = tag => {
7   return {
8     type: 'DELETE_TAG',
9     payload: tag,
10    }
11  }
12
13 export const addTag = tag => {
14   return {
15     type: 'ADD_TAG',
16     payload: tag,
17   }
18 }
19
20 export const getMutualUsers = username =>
21   dispatchHelper('GET_MUTUAL_USERS', 'get-mutual-users', { username })
22

```

The Marketplace has extensions that can help with '.snap' files

Search Marketplace Don't Show Again for '.snap' files

6:09 PM Tuesday 4/20/2021

You are screen sharing Stop Share

Mute Start Video Security Participants New Share Pause Share Annotate Remote Control More

Ln 1 Col 1 Spaces 2 UTF-8 LF JavaScript

```

settings.js - instagram2 - Visual Studio Code
File Edit Selection View Go Run Terminal Help
developer.hbs .env JS settings.js .gitignore db.sql JS db.test.js JS mail.test.js JS test.config.js JS user.test.js JS dbjs
React-Instagram-Clone-2.0-master > src > actions > JS settings.js > ...
1 import { dispatchHelper } from '../utils/utils'
2
3 export const getBlockedUsers = () =>
4   dispatchHelper('GET_BLOCKED_USERS', 'get-blocked-users')
5
6 export const unblockUser = block_id => {
7   return {
8     type: 'UNBLOCK_USER',
9     payload: block_id,
10    }
11  }
12

```

The Marketplace has extensions that can help with '.snap' files

Search Marketplace Don't Show Again for '.snap' files

6:09 PM Tuesday 4/20/2021

You are screen sharing Stop Share

Mute Start Video Security Participants New Share Pause Share Annotate Remote Control More

Ln 1 Col 1 Spaces 2 UTF-8 LF JavaScript

Third Year Mini Project Report

B.Tech. (CSE)-Session 2020-2021

```

File Edit Selection View Go Run Terminal Help
OPEN EDITORS 1 UNSAVED
REACT-INSTAGRAM-CLONE-2.0-MASTER > SRC > ACTIONS > post.js > ...
JS post.js x .env .gitignore db.sql JS db.test.js JS mail.test.js JS test.config.js JS user.test.js JS dbjs
54 }
55 }
56 export const getPostLikes = post =>
57   dispatchHelper('GET_POST_LIKES', 'get-post-likes', { post })
58
59 export const removeLike = like_id =>
60   dispatchHelper('REMOVE_LIKE', 'remove-like', { like_id })
61
62 export const getPostTags = post =>
63   dispatchHelper('GET_POST_TAGS', 'get-post-tags', { post })
64
65 export const untag = user => {
66   return {
67     type: 'UNTAG',
68     payload: user,
69   }
70 }
71
72 export const getUsersToShare = post =>
73   dispatchHelper('GET_USERS_TO_SHARE', 'get-users-to-share', { post })
74
75 export const getPostSharers = post =>
76   dispatchHelper('GET_POST_SHARERS', 'get-post-sharers', { post })
77
78 export const unbookmark = post => {
79   return {
80     type: 'UNBOOKMARK',
81     payload: post,
82   }
83 }
84
85 export const removeShare = share_id => {
86   return {
87     type: 'REMOVE_SHARE',
88     payload: share_id,
89   }
90 }
91
92 export const comment = comment => {
93   return {
94     type: 'COMMENT',
95     payload: comment,
96   }
97 }
98
99 export const deleteComment = comment_id => {
100   return {
101     type: 'DELETE_COMMENT',
102     payload: comment_id,
103   }
104 }

```

The Marketplace has extensions that can help with '.snap' files

Search Marketplace Don't Show Again for '.snap' files

6:09 PM Tuesday 4/20/2021

You are screen sharing Stop Share

Mute Start Video Security Participants New Share Pause Share Annotate Remote Control More

Ln 1 Col 1 Spaces 2 UTF-8 LF JavaScript

```

File Edit Selection View Go Run Terminal Help
OPEN EDITORS 1 UNSAVED
REACT-INSTAGRAM-CLONE-2.0-MASTER > SRC > ACTIONS > post.js > ...
JS post.js x .env .gitignore db.sql JS db.test.js JS mail.test.js JS test.config.js JS user.test.js JS dbjs
54 }
55 }
56 export const getPostLikes = post =>
57   dispatchHelper('GET_POST_LIKES', 'get-post-likes', { post })
58
59 export const removeLike = like_id =>
60   dispatchHelper('REMOVE_LIKE', 'remove-like', { like_id })
61
62 export const getPostTags = post =>
63   dispatchHelper('GET_POST_TAGS', 'get-post-tags', { post })
64
65 export const untag = user => {
66   return {
67     type: 'UNTAG',
68     payload: user,
69   }
70 }
71
72 export const getUsersToShare = post =>
73   dispatchHelper('GET_USERS_TO_SHARE', 'get-users-to-share', { post })
74
75 export const getPostSharers = post =>
76   dispatchHelper('GET_POST_SHARERS', 'get-post-sharers', { post })
77
78 export const unbookmark = post => {
79   return {
80     type: 'UNBOOKMARK',
81     payload: post,
82   }
83 }
84
85 export const removeShare = share_id => {
86   return {
87     type: 'REMOVE_SHARE',
88     payload: share_id,
89   }
90 }
91
92 export const comment = comment => {
93   return {
94     type: 'COMMENT',
95     payload: comment,
96   }
97 }
98
99 export const deleteComment = comment_id => {
100   return {
101     type: 'DELETE_COMMENT',
102     payload: comment_id,
103   }
104 }

```

The Marketplace has extensions that can help with '.snap' files

Search Marketplace Don't Show Again for '.snap' files

6:09 PM Tuesday 4/20/2021

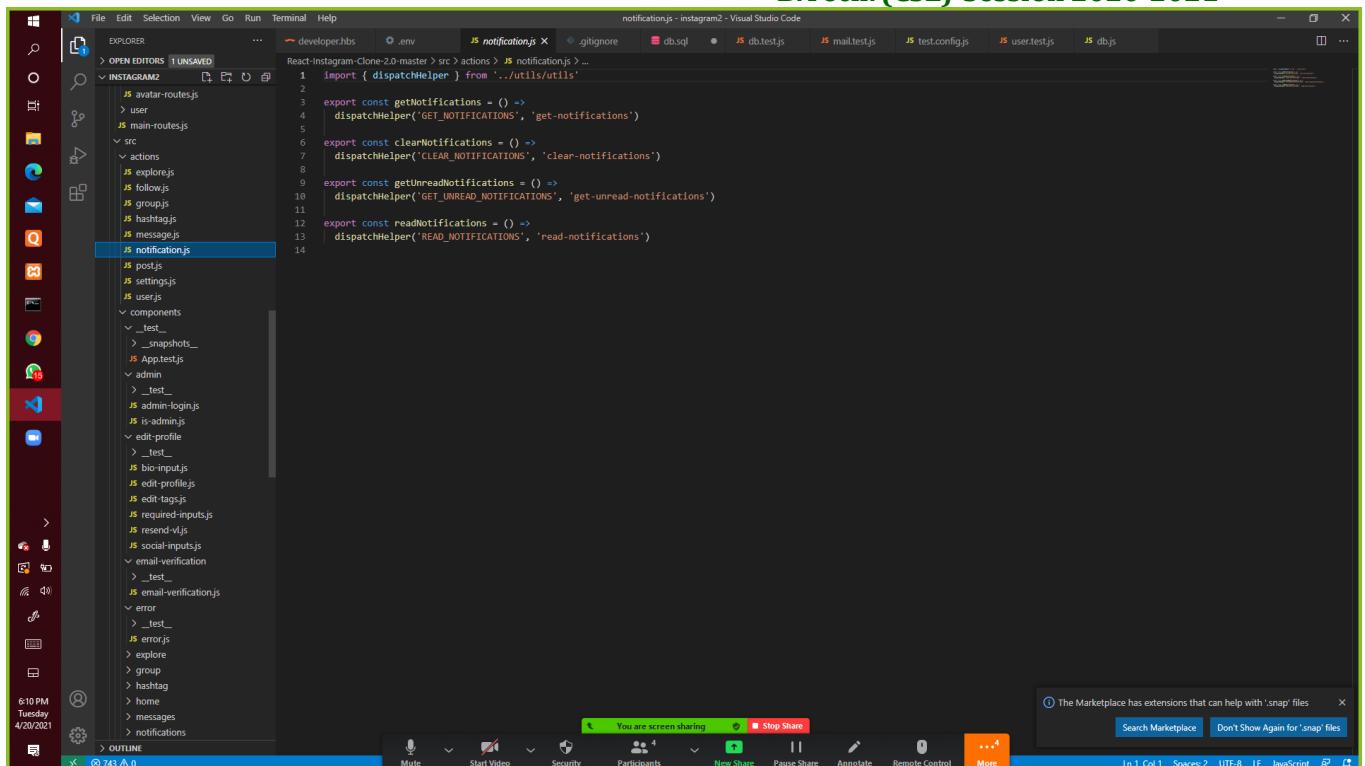
You are screen sharing Stop Share

Mute Start Video Security Participants New Share Pause Share Annotate Remote Control More

Ln 1 Col 1 Spaces 2 UTF-8 LF JavaScript

Third Year Mini Project Report

B.Tech. (CSE)-Session 2020-2021

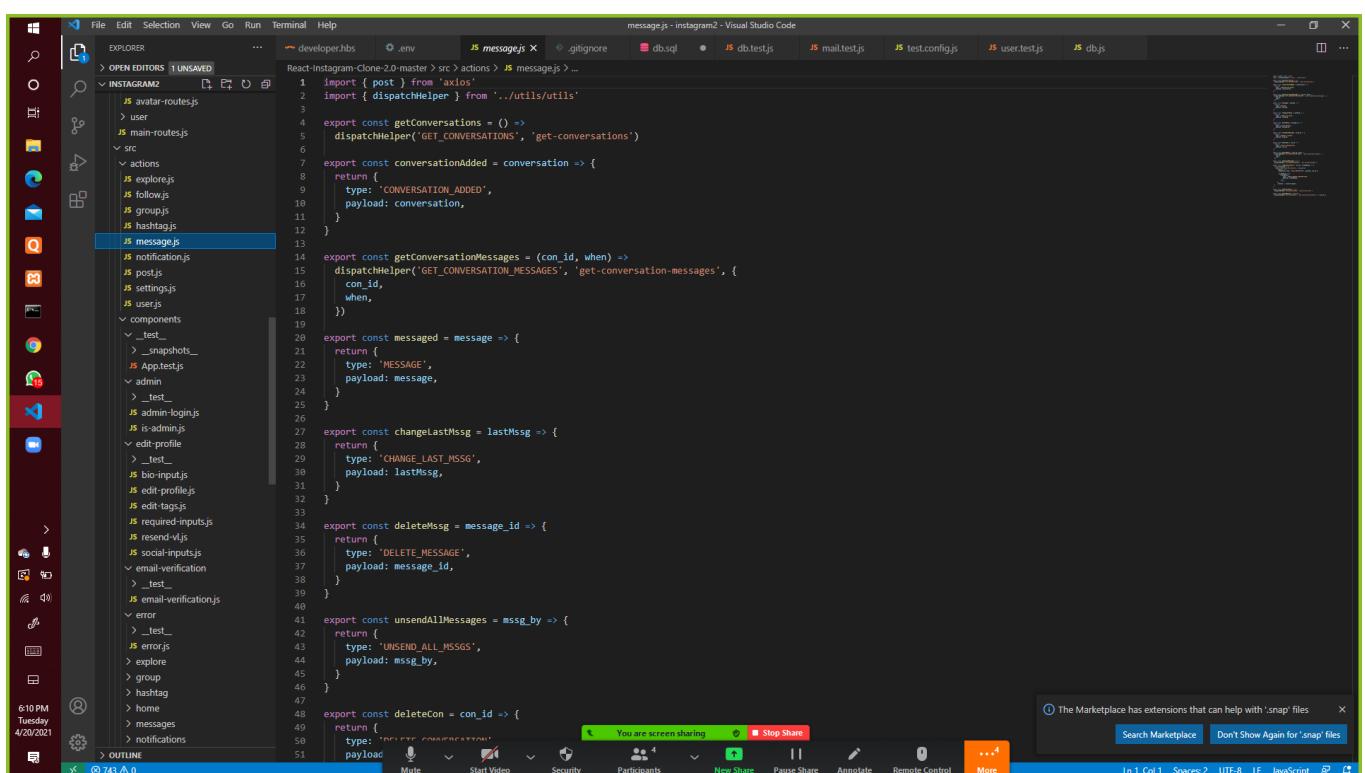


The screenshot shows the Visual Studio Code interface with the 'notification.js' file open in the editor. The code handles various notification actions like getting notifications, clearing notifications, and reading notifications.

```

1 import { dispatchHelper } from '../utils/utils'
2
3 export const getNotifications = () =>
4   dispatchHelper('GET_NOTIFICATIONS', 'get-notifications')
5
6 export const clearNotifications = () =>
7   dispatchHelper('CLEAR_NOTIFICATIONS', 'clear-notifications')
8
9 export const getUnreadNotifications = () =>
10  dispatchHelper('GET_UNREAD_NOTIFICATIONS', 'get-unread-notifications')
11
12 export const readNotifications = () =>
13  dispatchHelper('READ_NOTIFICATIONS', 'read-notifications')
14

```



The screenshot shows the Visual Studio Code interface with the 'message.js' file open in the editor. The code defines functions for getting conversations, adding conversations, getting conversation messages, sending messages, changing last message, deleting messages, and unsend all messages.

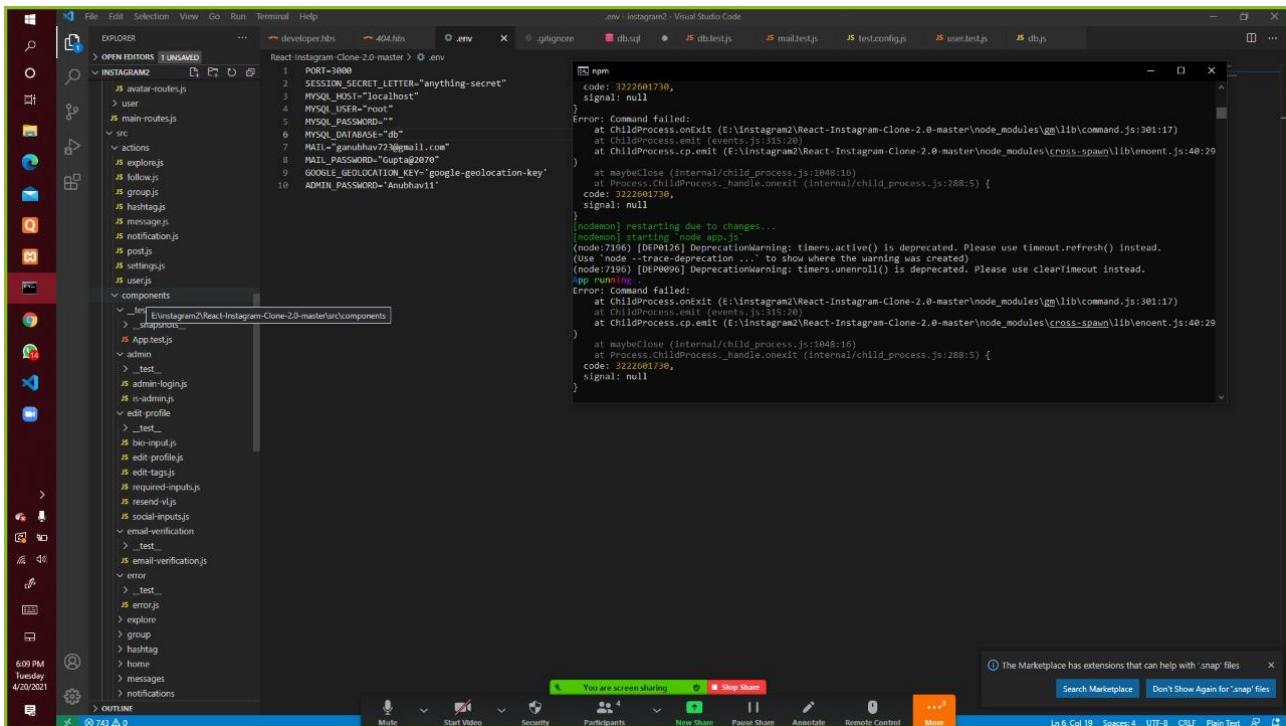
```

1 import { post } from 'axios'
2 import { dispatchHelper } from '../utils/utils'
3
4 export const getConversations = () =>
5   dispatchHelper('GET_CONVERSATIONS', 'get-conversations')
6
7 export const conversationAdded = conversation => {
8   return {
9     type: 'CONVERSATION_ADDED',
10    payload: conversation,
11  }
12}
13
14 export const getConversationMessages = (con_id, when) =>
15  dispatchHelper('GET_CONVERSATION_MESSAGES', 'get-conversation-messages', {
16    con_id,
17    when,
18  })
19
20 export const messaged = message => {
21   return {
22     type: 'MESSAGE',
23     payload: message,
24   }
25 }
26
27 export const changeLastMssg = lastMssg => {
28   return {
29     type: 'CHANGE_LAST_MSSG',
30     payload: lastMssg,
31   }
32 }
33
34 export const deleteMssg = message_id => {
35   return {
36     type: 'DELETE_MESSAGE',
37     payload: message_id,
38   }
39 }
40
41 export const unsendAllMessages = mssg_by => {
42   return {
43     type: 'UNSEND_ALL_MESSAGES',
44     payload: mssg_by,
45   }
46 }
47
48 export const deleteCon = con_id => {
49   return {
50     type: 'DELETE_CONVERSATION',
51     payload
52   }
53 }

```

Implementation

Command Line Server Running



```

PORT=3000
SECRET.LETTER="anything-secret"
MYSQL_HOST="localhost"
MYSQL_USER="root"
MYSQL_PASSWORD=""
MYSQL_DATABASE="db"
MAIL="ganubhav72@gmail.com"
MAIL_PASSWORD="Guptaj2870"
GOOGLE_GEOLOCATION_KEY="google-geolocation-key"
ADMIN_PASSWORD="Anubhav11"

npm
  code: 3222601730,
  signal: null
)
Error: Command failed:
  at ChildProcess.onExit (E:\Instagram2\React-Instagram-Clone-2.0-master\node_modules\gm\lib\command.js:301:17)
  at ChildProcess.emit (events.js:315:20)
  at ChildProcess.cp.emit (E:\Instagram2\React-Instagram-Clone-2.0-master\node_modules\cross-spawn\lib\enoent.js:40:29)
  at maybeClose (internal/child_process.js:1048:16)
  at Process.ChildProcess._handle.onexit (internal/child_process.js:288:5) {
  code: 3222601730,
  signal: null
}
[nodemon] restarting due to changes...
[nodemon] starting `node .` with babel
(node:7196) [DEP0116] DeprecationWarning: timers.active() is deprecated. Please use timeout.refresh() instead.
(Use `node --trace-deprecation ...` to show where the warning was created)
(node:7196) [DEP0006] DeprecationWarning: timers.unenroll() is deprecated. Please use clearTimeout instead.
npm run dev
Error: Command failed:
  at ChildProcess.onExit (E:\Instagram2\React-Instagram-Clone-2.0-master\node_modules\gm\lib\command.js:301:17)
  at ChildProcess.emit (events.js:315:20)
  at ChildProcess.cp.emit (E:\Instagram2\React-Instagram-Clone-2.0-master\node_modules\cross-spawn\lib\enoent.js:40:29)
  at maybeClose (internal/child_process.js:1048:16)
  at Process.ChildProcess._handle.onexit (internal/child_process.js:288:5) {
  code: 3222601730,
  signal: null
}

```



DATABASE

Showing rows 0 - 18 (19 total). Query took 0.0004 seconds.

SELECT * FROM `users`

Profile Edit inline | Edit | Explain SQL | Create PHP code | Refresh

	id	username	firstname	surname	email	password	bio	joined	email_verified	account_type
<input type="checkbox"/>	7	ghalib	Mirza	Ghalib	ghalib@gmail.com	\$2a\$10\$e3ZgKsWaa6rUopG1CUBm8OoCMKVqzSw79bfuUrCV0...		1514718748562	yes	private
<input type="checkbox"/>	8	coldplay	cold	play	coldplay@gmail.com	\$2a\$10\$zvPMQDJkIOY00UhSlrlEfufiaeTwXk2.VD4ixp.q1x2Rj...		1515918435853	no	public
<input type="checkbox"/>	10	noddy	your	noddy	noddy@gmail.com	\$2a\$10\$Frxkj9047TnMo.g9JTe5cwFakoJc4/w9kba3LeAd...		1516454412744	no	public
<input type="checkbox"/>	11	nobita	nobita	nobi	nobita@gmail.com	\$2a\$10\$nzMi2G0547CtUfuof4zKEOWhpwUWKzQwV67JpRqaqqS...		1516522466189	no	public
<input type="checkbox"/>	12	pikachu	pikachu	pikachu	pikachu@gmail.com	\$2a\$10\$ylnEwjuqkczB5hOlLe4dVXXk56KTetE4arAj...		1516522598741	no	public
<input type="checkbox"/>	13	iamsrk	Shahrukh	Khan	iamsrk@gmail.com	\$2a\$10\$Xn99377.3NsQOneTp4qeMuERyNzR2kI86JrpMHCs...		1516523593107	no	public
<input type="checkbox"/>	14	kinkade	Thomas	Kinkade	kinkade@gmail.com	\$2a\$10\$ivK3CvFn/dmkWzIRM9k.S2/WldOb6adf7B616.G...		1516523852484	no	public
<input type="checkbox"/>	15	sunryo	sunryo	honekawa	sunryo@gmail.com	\$2a\$10\$6GUUnk2jhH8RolloA/nLZG076.bc8PfhquEuc2kE...		1516524667640	no	public
<input type="checkbox"/>	16	zayn	Zayn	Mallie	zayn@gmail.com	\$2a\$10\$5tpkvoi8BxfmxNyTpQuNqgyXc5vn4.nslsVR4eM...		151652072573	no	private
<input type="checkbox"/>	17	rftak	Nusratfateh	Allikhan	rftak@gmail.com	\$2a\$10\$TAz2lpuYis/HfRB8LPhvZD0iKtT5fmUvcUghIEGyJwn...		1516525201164	no	public
<input type="checkbox"/>	18	ragnar	Ragnar	Lothbrok	ragnar@gmail.com	\$2a\$10\$M7ix4wFPUhAJsvXb7bw nk2G6zxeCjhXbnKtYFzJ...		1516525343645	no	public
<input type="checkbox"/>	19	jonsnow	jon	snow	jonsnow@gmail.com	\$2a\$10\$9Nb4hfjgg MKKLtExuWhratlT2UCoeSpq3./VMM...		1516527326854	no	public
<input type="checkbox"/>	20	gian	Takeshi	Gauda	gian@gmail.com	\$2a\$10\$K3jp44HIOKuq5y3QD4W5O5eeAOshwogEQIB/F...		1516527534985	no	public
<input type="checkbox"/>	24	takkar	iam_	takkar	takkar@gmail.com	\$2a\$10\$RjWFCWeDgm0PwCB3weerTCo0RuYzIifD2Ngy3...	Hello #world	1518016437193	no	private
<input type="checkbox"/>	27	taylor_swift	taylor	swift	taylor_swift@gmail.com	\$2a\$10\$nmQRsp0WCoV86dAD24nJ.tLr5XG31ejULQOMVkBpj...		1518018283428	no	public
<input type="checkbox"/>	28	selena	selena	gomez	selenagomez@gmail.com	\$2a\$10\$ifdYIKQdtacXln09nLuENyJlsfZlJq24tqzN...		151801809165	no	public
<input type="checkbox"/>	29	steve_jobs	steve	jobs	steve_jobs@gmail.com	\$2a\$10\$B05HNF3jnjk.8f7UkCI4puaLJSLpxW90mnk3Md2sA...		1518018498672	no	public
<input type="checkbox"/>	30	doraemon	iam_	doraemon	doraemon@gmail.com	\$2a\$10\$OjZgmsNPOT27skkotUetzY7lmEfDVxVPP2lsAv...		151845660501	no	public
<input type="checkbox"/>	31	anubhav11	Anubhav	Gupta	Anubhav_gupta_c18@glia.ac.in	\$2a\$10\$6egm0WjhExTyg0zGairJm6VkgE8tgRhGrcRL...		161852077365	no	public

You are screen sharing Stop Share

Mute Start Video Security Participants New Share Pause Share Annotate Remote Control More

Console

Third Year Mini Project Report B.Tech. (CSE)-Session 2020-2021

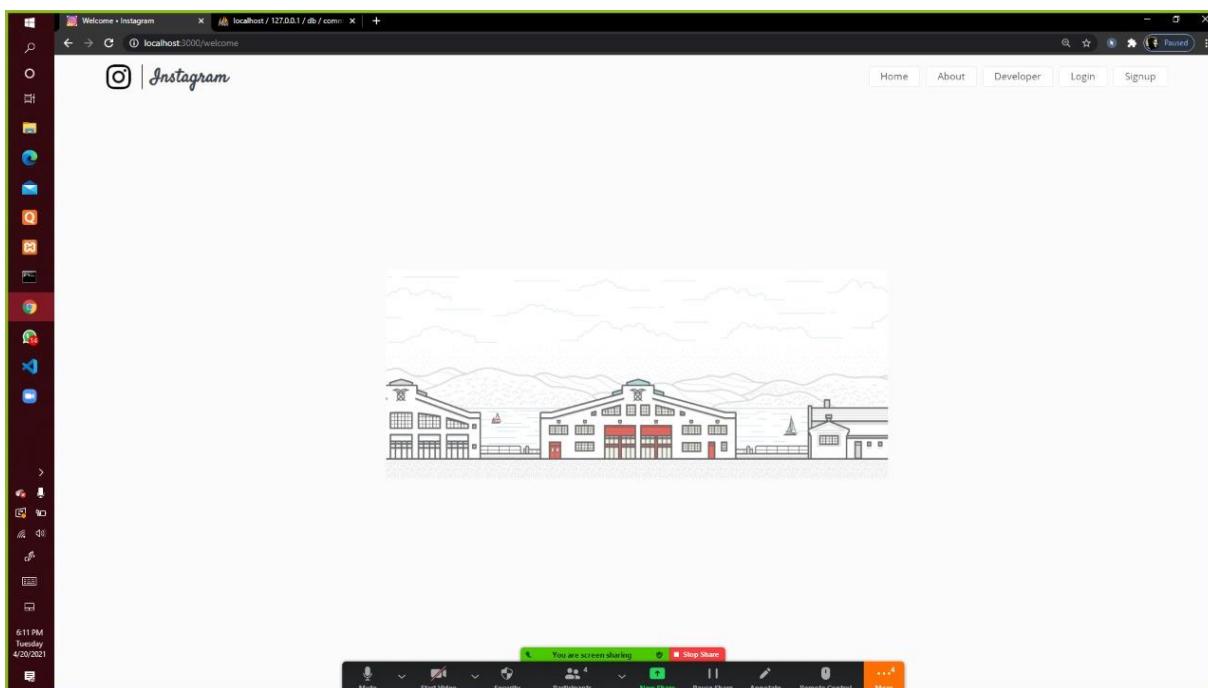
The screenshot shows the phpMyAdmin interface for the 'likes' table. The table has columns: like_id, post_id, like_by, and like_time. The data shows various likes recorded over time, with IDs ranging from 31 to 76.

like_id	post_id	like_by	like_time
31	22	12	1516523221375
33	24	13	1516523819069
34	23	11	1516524845235
35	32	10	1516524986159
36	34	16	1516525136147
37	35	17	1516525294230
38	36	18	1516525560721
39	41	20	1516527811400
42	43	18	1516528068337
57	43	23	1518016342615
58	40	24	1518016509189
59	39	24	1518016512324
63	61	27	1518018364468
64	63	30	1518510087356
65	57	24	1518515315123
67	89	28	1524245733030
69	89	18	1525862381809
71	69	24	1526210612292
75	89	24	1526663921539
76	34	31	1618920860114

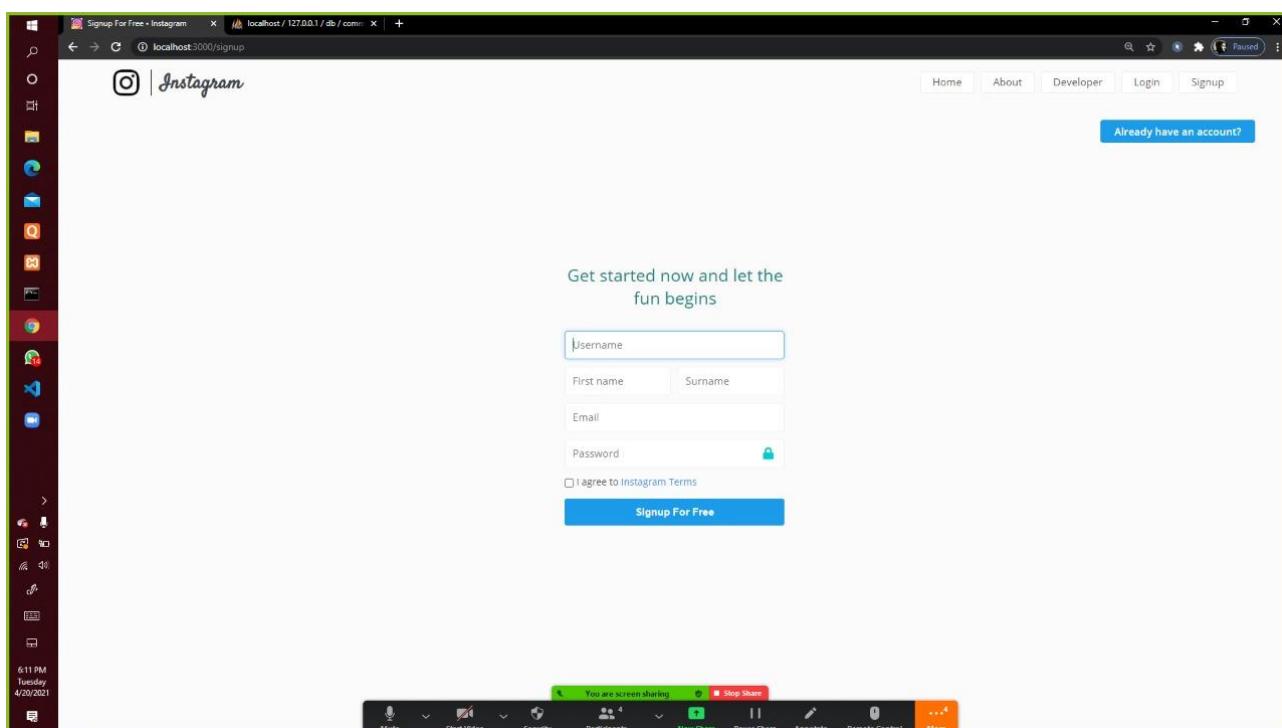
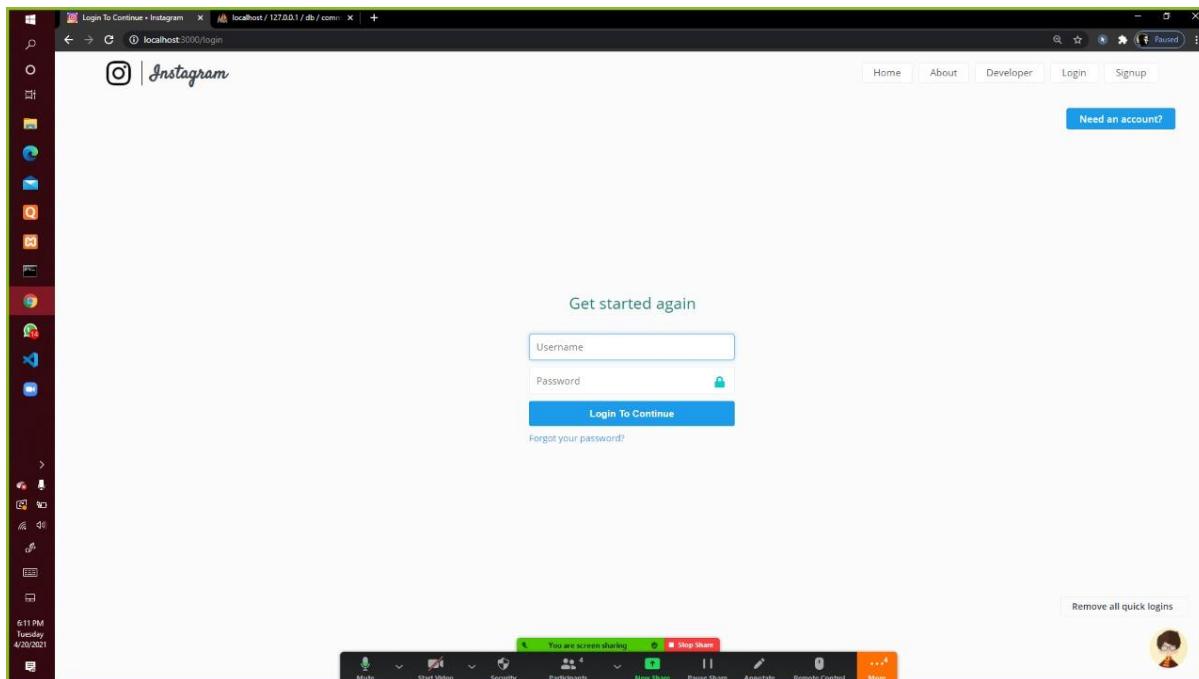
The screenshot shows the phpMyAdmin interface for the 'comments' table. The table has columns: comment_id, type, commentSrc, comment_by, post_id, and comment_time. The data shows various comments posted by users, with IDs ranging from 62 to 104.

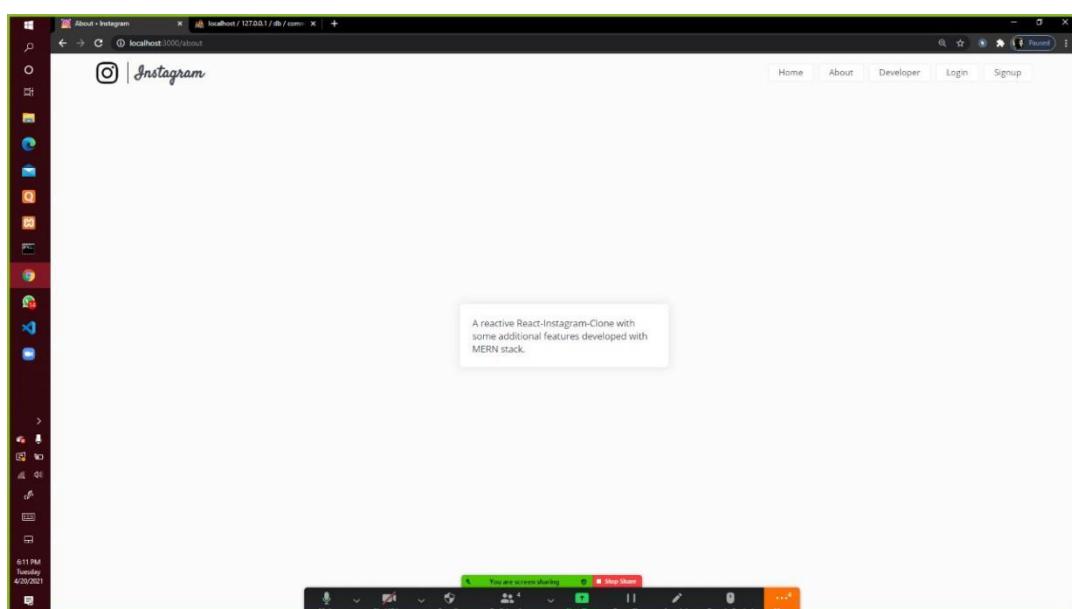
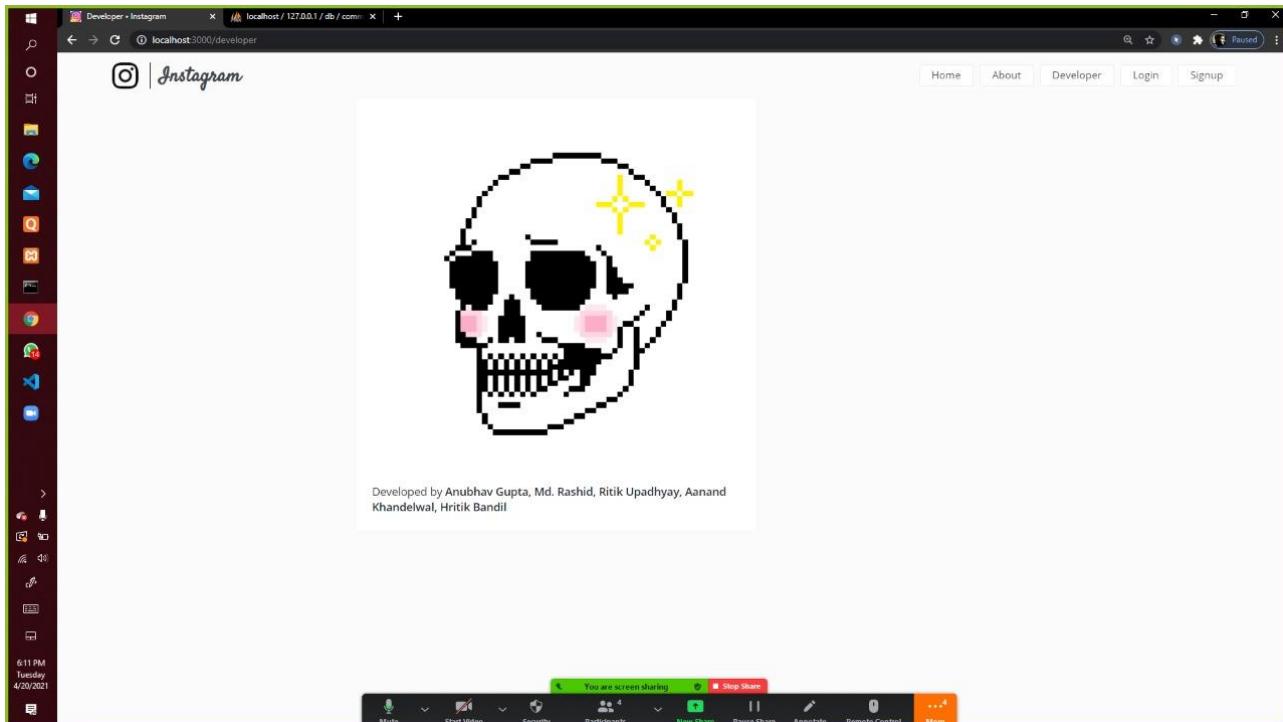
comment_id	type	commentSrc	comment_by	post_id	comment_time
62	text	mooooooooooooooooooooo...	30	57	151850978092
63	text	Lorem Ipsum is simply dummy text of the printing a...	30	57	15185092592
64	text	#Hello	24	57	151871098344
66	text	wooo @takkar	7	88	151894552477
69	image	instagram_comment_1518972851259.jpg	24	89	151897285125
70	text	thnx @ghalib	24	88	151897293273
71	text	hmmm	7	88	151897304103
73	text	https://egrex.com/23765ffffm	24	89	151911367158
102	text	mmmm	24	61	152476425044
103	sticker	instagram_comment_1527447892610.jpg	24	43	152744789261
104	sticker	instagram_comment_1527447929485.jpg	24	43	152744792948

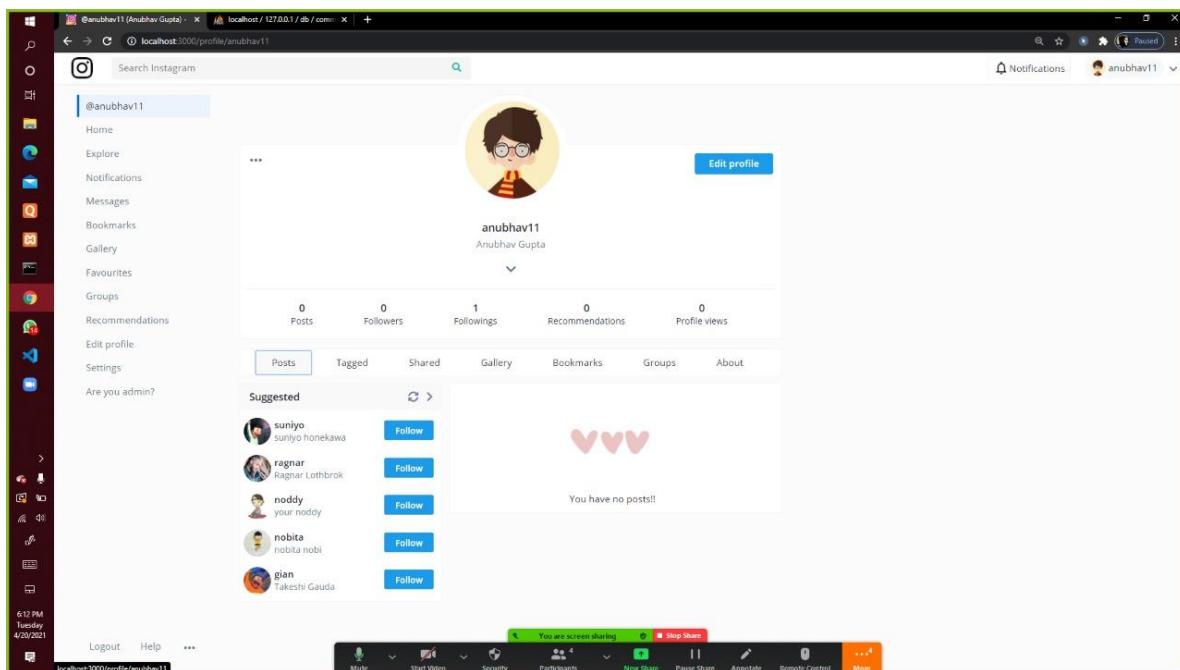
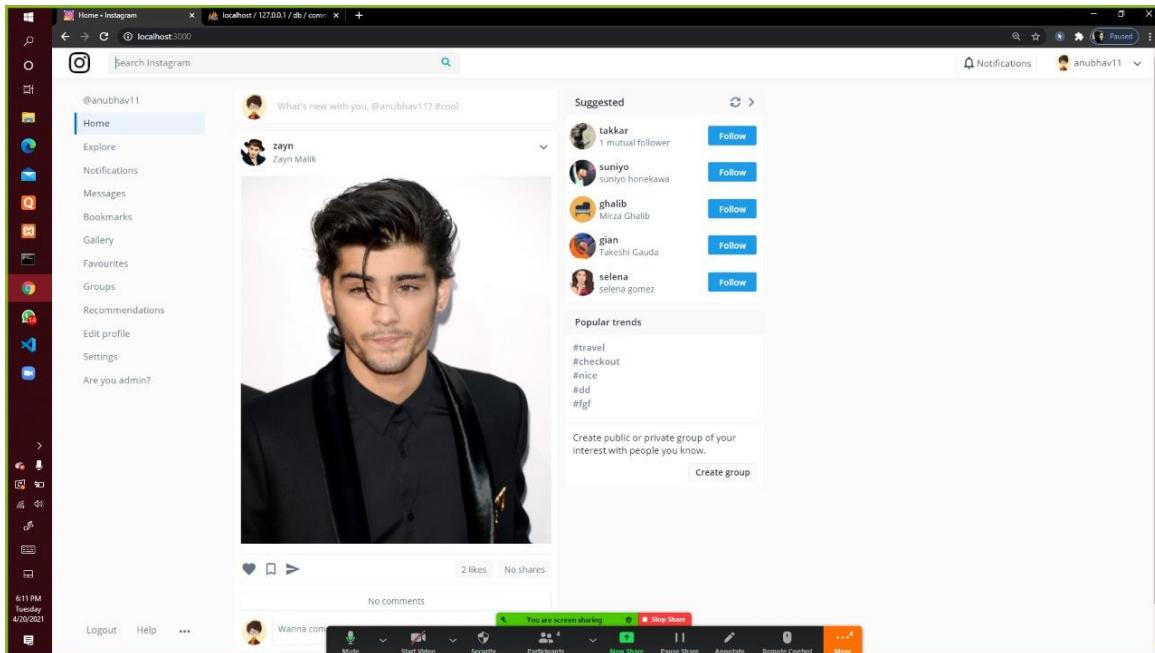
Home Page

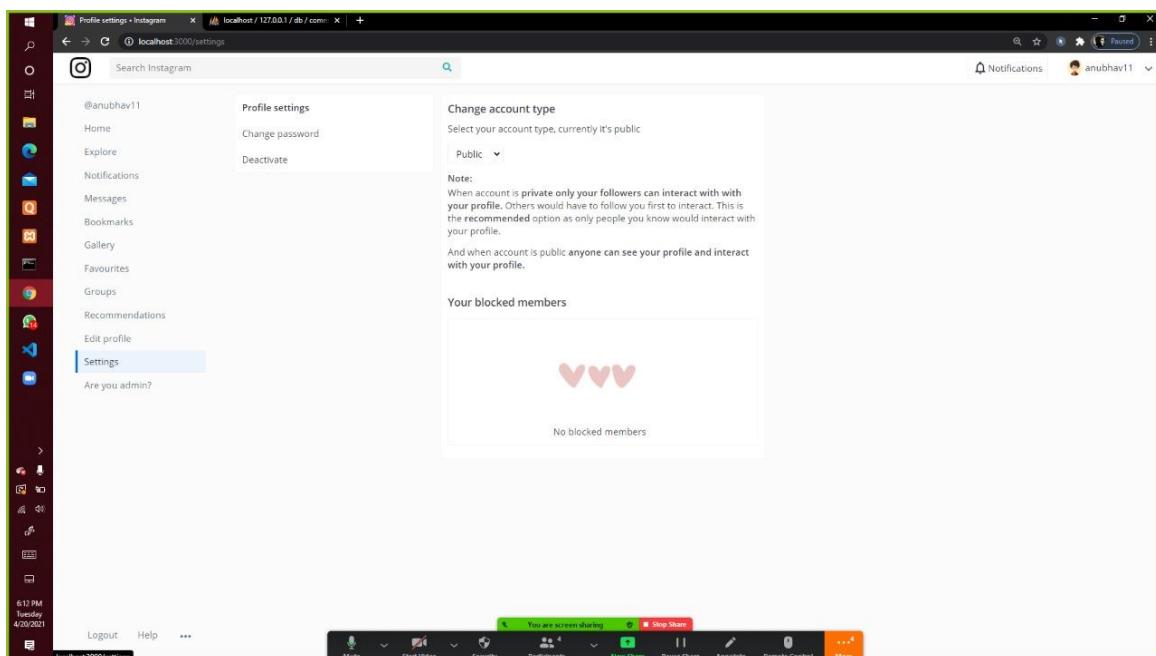
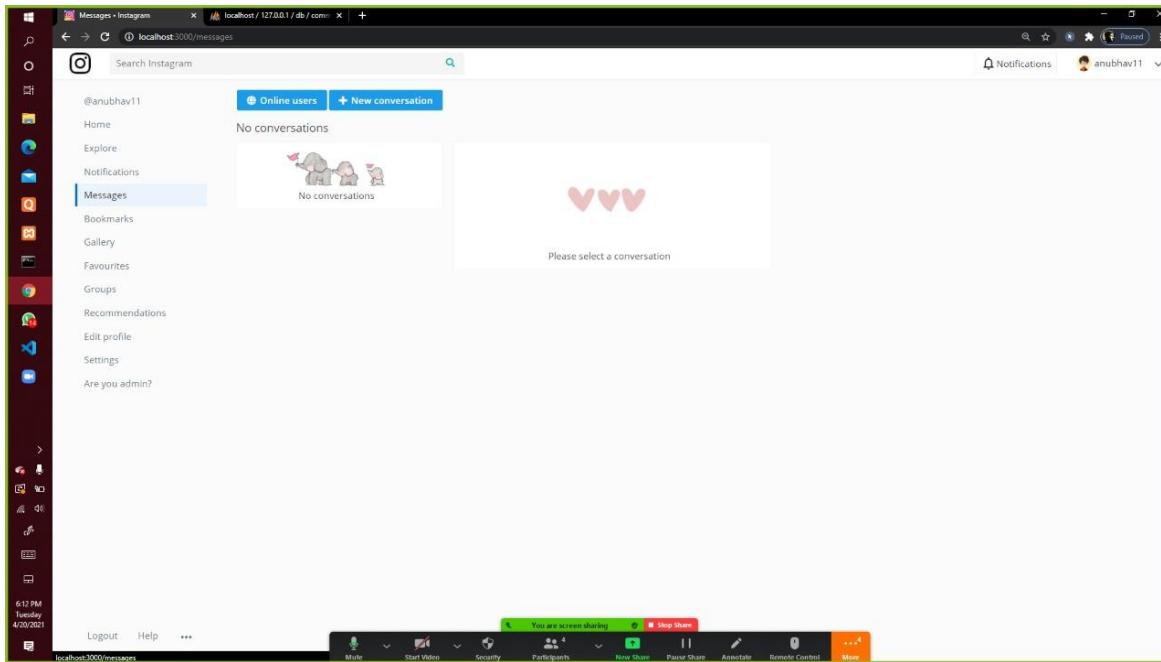


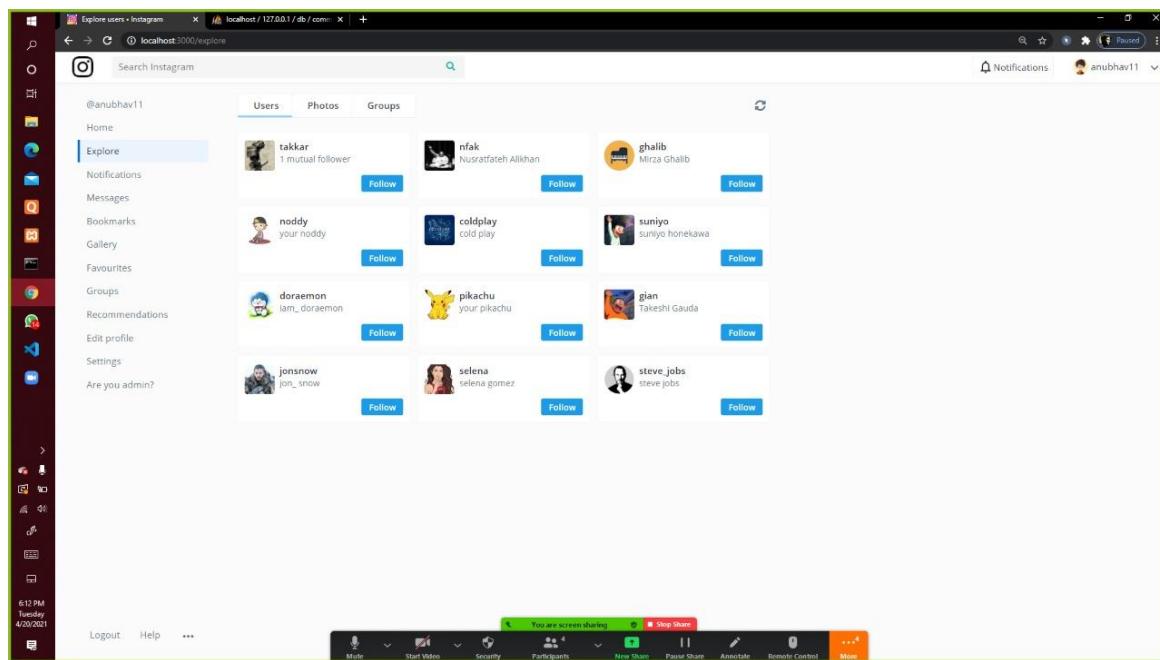
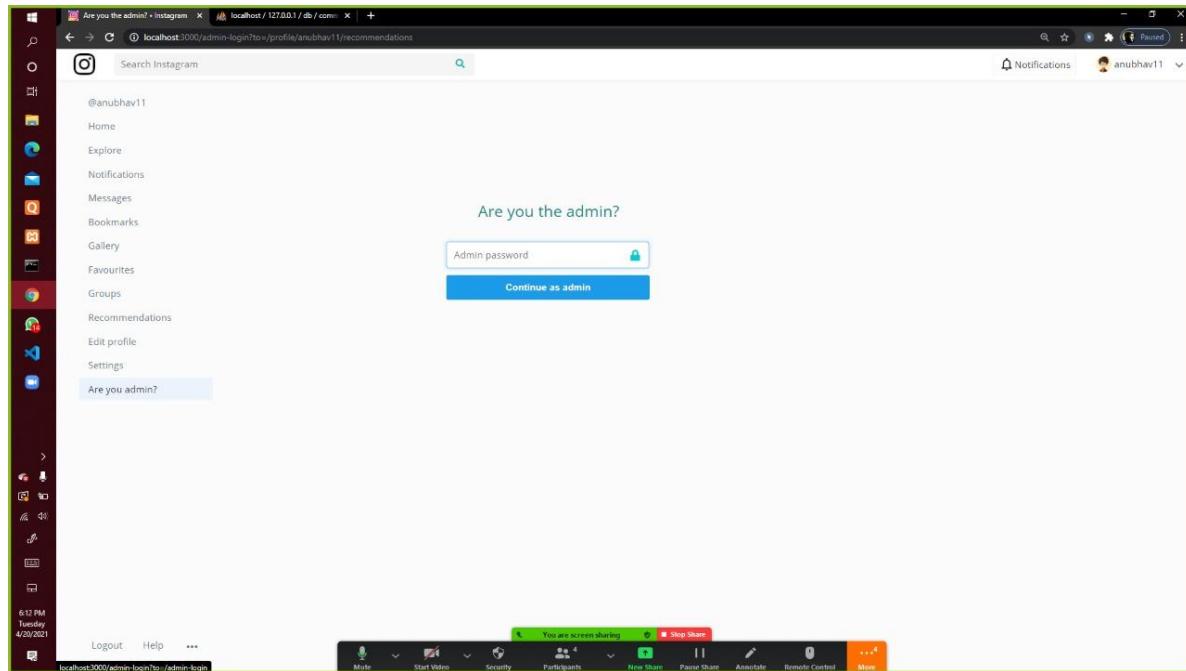
Login and Signup

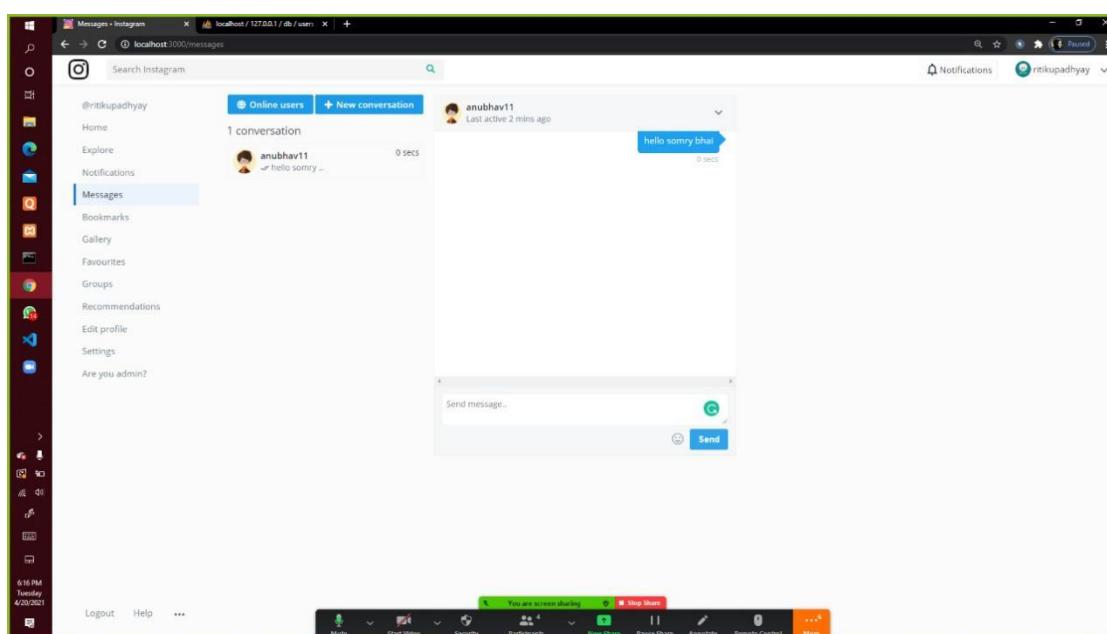
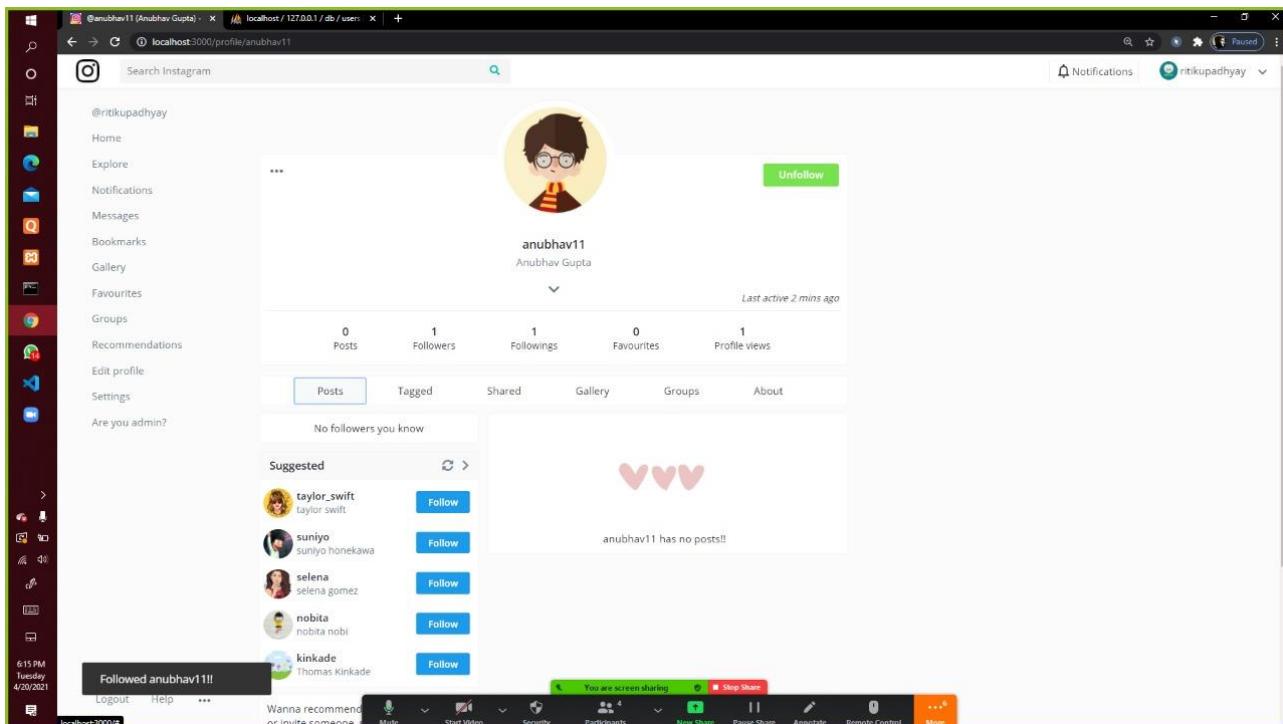


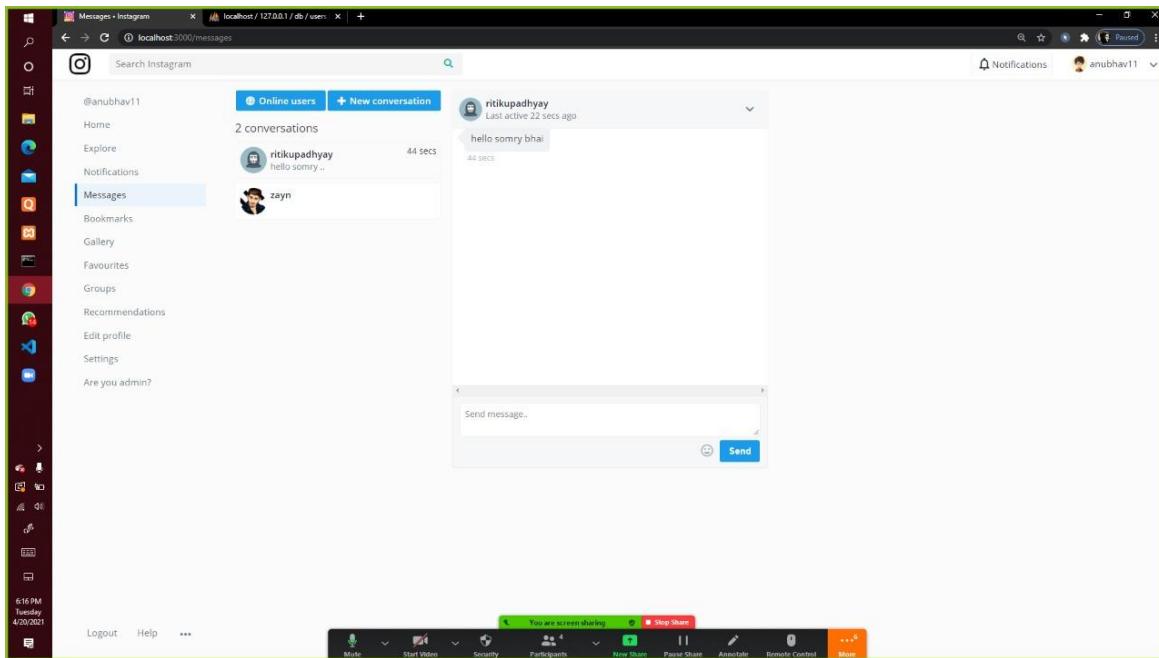




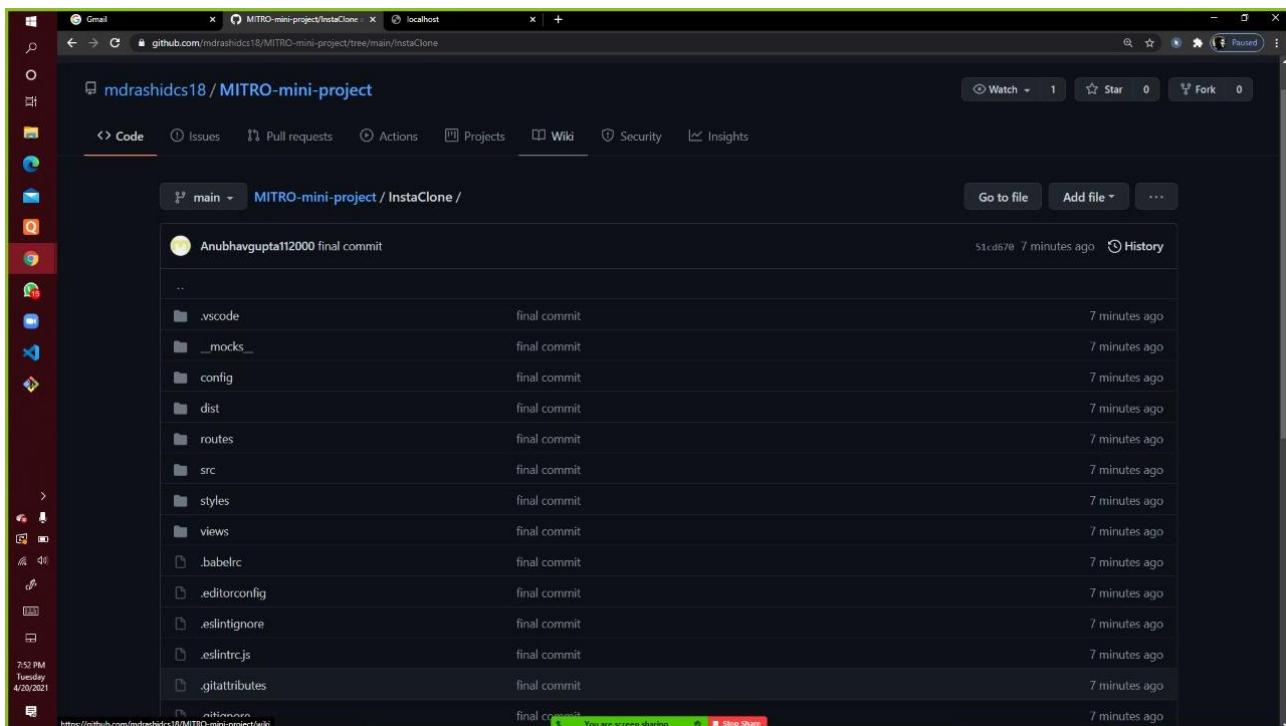








GitHub Page



Testing

Test id	Test case title	Description	Expected Outcome	Result
1.	Login and Signup	All the Users first signup and then login.	Users are logged.	Success
2.	Search and Follow	User search for their friends by typing their user names and follow them if they want.	Showed the user profile which is searched and show them a follow option.	Success
3.	Notification	If someone follows your account and send u a message u must get a notification .	Notification showed to the other user whenever he/she opened account.	Success
4.	Messaging	U can message someone u want .	Message received by the other person.	Success
5.	LogOut	After using the app when user want to close the site he must logout of their account.	Logout successfully done and user get back to the main page of the website.	Success

References

<http://www.youtube.com>

<http://www.udemy.com>

<http://www.coursera.org>

<http://www.w3schools.com>

Thank You