

**AcadIn**

**A MAJOR PROJECT  
REPORT**

Submitted in Partial Fulfillment of  
the Requirements for the Degree  
of

**BACHELOR OF TECHNOLOGY**

in

**COMPUTER SCIENCE AND ENGINEERING**

by

**Anuj Pratap Gangwar** (1901500100014)

**Prateek Singhal** (1901500100066)

**Nitish Kumar** (1901500100063)

Under the Supervision of

**Mrs. Apoorva Jain**

(Assistant Professor | Department of CSE)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**CET-IILM-AHL, GREATER NOIDA**

**Dr. A.P.J. ABDUL KALAM TECHNIAL UNIVERSITY, LUCKNOW**

**2022-2023**



## Department of Computer Science and Engineering

### **CERTIFICATE**

This is to certify that the **KCS852 Project (B.Tech final year, 8<sup>th</sup> Semester)**, titled **“AcadIn.”** submitted by **Anuj Pratap Gangwar (1901500100014), Nitish Kumar (1901500100063) and Prateek Singhal (1901500100066)** was carried out under my guidance and supervision.

To the best of my knowledge and belief, the information presented in this report is their own task and has not been submitted elsewhere.

**Dr. Shilpy Agrawal**

Head of Department

(Associate Professor | Department of CSE)

**Mrs. Apoorva Jain**

Faculty Supervisor

(Assistant Professor | Department of CSE)

## **DECLARATION**

I hereby declare that the work presented in this project report entitled “AcadIn” was carried out by us. I have not submitted the matter embodied in this report for the award of any other degree or diploma of any other University or Institute. I have given due credit to the original authors/sources for all the words, ideas, diagrams, graphics, computer programs, experiments, results, that are not my original contribution. I have used quotation marks to identify verbatim sentences and given credit to the original authors/sources. I affirm that no portion of my work is plagiarized, and the experiments and results reported in the project report are not manipulated. In the event of a complaint of plagiarism and the manipulation of the experiments and results, I shall be fully responsible and answerable.

Signature:

Name : Anuj Pratap Gangwar  
Roll No. 1901500100014

Signature:

Name : Nitish Kumar  
Roll No. 1901500100063

Signature:

Name : Prateek Singhal  
Roll No. 1901500100066

## **ACKNOWLEDGEMENT**

It gives us a great sense of pleasure to present the report of the project undertaken during B.Tech. fourth year. We owe special debt of gratitude to **Mrs. Apoorva Jain (Faculty Supervisor)**, Assistant Professor in the Department of Computer Science & Engineering, College of Engineering and Technology IILM-AHL, Greater Noida for her constant support and guidance throughout the course of our work. His sincerity, thoroughness and perseverance have been a constant source of inspiration for us.

Name : Anuj Pratap Gangwar  
Roll No. 1901500100014

Name : Nitish Kumar  
Roll No. 1901500100063

Name : Prateek Singhal  
Roll No. 1901500100066

## **TABLE OF CONTENTS**

<b>S.NO</b>	<b>TITLE</b>	<b>PAGE NO</b>
1	CHAPTER- 1   INTRODUCTION	1
2	CHAPTER- 2   SYSTEM REQUIREMENT	3
3	CHAPTER- 3   WORKING & TESTING	8
4	CHAPTER- 4   HARDWARE & SOFTWARE USED	20
5	ADVANTAGES & LIMITATIONS	21
6	CHAPTER- 5   ER MODEL	22
7	CHAPTER- 6   DFD LEVELS	23
8	CHAPTER- 7   DFD LEVELS: CONTEXT DIAGRAM	26
9	CHAPTER- 8   CONCLUSION	27
10	CHAPTER- 9   FUTURE SCOPE	28
11	REFERENCES	29

## **LIST OF FIGURES**

<b>FIG. NO.</b>	<b>TITLE</b>	<b>PAGE NO.</b>
1.1	Black Box Testing	8
1.2	SIGN UP	9
1.3	Successfully login with new account	9
1.4	Sign-Up with an existing email	10
2.1	Successfully login with correct email & password	11
2.2	Successfully login with new account	11
2.3	Login with incorrect email	12
2.4	Login with correct email but incorrect password	12
3.1	Adding experience, education and bio	13
3.2	Profile page after update	13
3.3	Updating experience, education and bio	14
3.4	Profile page after this update	14
4.1	Adding only text in the post	15
4.2	Post added as shown on Home Page	15
4.3	User adds photo along with text in the post	16
4.4	Post after comment and like	16
5.1	User searches for his or her existing friend	17
5.2	User views friend's profile	17
5.3	User can follow his friends	18

5.4	User can unfollow his friends	18
6.1	User adds a friend to chat	19
6.2	After click on add user icon	19
6.3	After clicking on user that he/she want to add	19
6.4	After clicking on user that was added	20
6.5	After sending “Hi” message to the user	20
6.6	After getting reply from the user	20
7.1	ENTITY-RELATIONSHIP MODEL	23
8.1	DFD Levels (0,1)	24
8.2	DFD Level 2	25
9.1	DFD LEVELS: Context Diagram	26

## **ABSTRACT**

- Keep the students of IILM CET connected through a virtual platform and thus reduce the hesitation amongst juniors to get advice/suggestions from their seniors, share achievements, facilitating more interactions thus bringing the student community closer.
- It will bring the student community of IILM CET closer and allows like-minded students to collaborate like any other social media platform but one specifically designed for their needs.
- The proposed system is specifically designed for the students of IILM CET and can be later customized for different institutions.
- Users can create accounts, follow other users, share posts and discuss. There is filter in app to filter out different students according to their streams. Every student can post their achievements, work experience, project, etc. to share with other people.



## **CHAPTER- 1**

### **INTRODUCTION**

This project is a web-based application aimed at making a platform where college students can use it to share their work-based or learning experiences and achievements with others through texts, photos and videos.

- Keep the students of IILM college connected through a virtual platform and thus reduce the hesitation amongst juniors to get advice/suggestions from their seniors, share achievements, facilitating more interactions thus bringing the student community closer.
- This way, the app not only creates new relationship between students but also allows others to see what they are doing and how they did it and take help from it.
- It will bring the student community of IILM closer and allows like-minded students to collaborate like any other social media platform but one specifically designed for their needs.
- Users can create accounts, follow other users, share posts and discuss. There is filter in app to filter out different students according to their streams. Every student can post their achievements, work experience, project, etc. to share with other people.

#### **1.1 Existing System Current Model:**

Currently a variety of social media apps for students such as LinkedIn facilitate the same.

Users can create their profile, post achievements, like, comment and share posts, have 1-1 conversations with anyone, follow anyone and view their profiles.

#### **1.2 Limitations of Existing System:**

The above platform is built keeping only limited forms of interactions in mind, like

1-1 conversations, and creating and viewing profiles. It's not user friendly for discussions for students. They also have unwanted advertisements which serve as a distraction.

### **1.3 Proposed System:**

The proposed system is specifically designed for the students of IIITG and can be later customized for different institutions. Its functionalities include

- Sign Up / Sign In -> Students can sign up for posting or seeing content on the application or if already signed up, they can sign in to their profile.
- Posts -> Users can create their own post or view/interact with others posts by liking or commenting on their post.
- Chat -> Any user can chat with their followers/connections.

## **CHAPTER- 2**

### **SYSTEM REQUIREMENT SPECIFICATION**

The requirements specification is a technical specification of requirements for the software products. It is the first step in the requirements analysis process it lists the requirements of a particular software system including functional, performance and security requirements. The requirements also provide usage scenarios from a user, an operational and an administrative perspective. The purpose of software requirements specification is to provide a detailed overview of the software project, its parameters and goals. This describes the project target audience and its user interface, hardware and software requirements. It defines how the client, team and audience see the project and its functionality.

The application requirements can be divided into Functional and Non-Functional requirements. Functional requirements define the capabilities and functions that a system must be able to perform successfully. Non-Functional requirements Define the qualities and criteria that can be used to judge the operation of a system.

#### **2.1 Functional Requirements**

- I. Users must have a valid User ID and password to login thus creating their individual profiles.
- II. Users can update their profiles by adding which year they are in, their skills, profile picture, bio etc.
- III. Like and Comment on the posts.
- IV. Follow/Connect with other students.
- V. Chat with friends

#### **2.2 Non-Functional Requirements**

- I. Secure access of confidential data (user's details) .
- II. Better component design to get efficiency at peak time.
- III. Flexible service-based architecture will be highly desirable for future extension.

## **2.3 TECHNOLOGIES USED**

Web application development is the creation of application programs that reside on remote servers and are delivered to the user's device over the Internet. A web application (web app) does not need to be downloaded and is instead accessed through a network. An end user can access a web application through a web browser such as Google Chrome, Safari, or Mozilla Firefox. A majority of web applications can be written in JavaScript, Cascading Style Sheets (CSS), and HTML5.

Web application development will typically have a short development life-cycle lead by a small development team. Front-end development for web applications is accomplished through client-side programming. Client refers to a computer application such as a web browser. Client-side programming will typically utilize HTML, CSS and JavaScript. HTML programming will instruct a browser how to display the on-screen content of web pages, while CSS keeps displayed information in the correct format. JavaScript will run JavaScript code on a web page, making some of the content interactive.

Server-side programming powers the client-side programming and is used to create the scripts that web applications use. Scripts can be written in multiple scripting languages such as Ruby, Java and Python. Server-side scripting will create a custom interface for the end-user and will hide the source code that makes up the interface.

A database such as MySQL or MongoDB can be used to store data in web application development.

### **2.3.1 Node.JS**

Node.js is a cross-platform, open-source server environment that can run on Windows, Linux, Unix, macOS, and more. Node.js is a back-end JavaScript runtime environment, runs on the V8 JavaScript Engine, and executes JavaScript code outside a web browser.

Node.js lets developers use JavaScript to write command line tools and for server-side scripting. The ability to run JavaScript code on the server is often used to generate dynamic web page content before the page is sent to the user's web browser.

Consequently, Node.js represents a "JavaScript everywhere" paradigm, unifying web-application development around a single programming language, as opposed to using different languages for the server- versus client-side programming.

Node.js has an event-driven architecture capable of asynchronous I/O. These design choices aim to optimize throughput and scalability in web applications with many input/output operations, as well as for real-time Web applications (e.g., real-time communication programs and browser games).

The Node.js distributed development project was previously governed by the Node.js Foundation, and has now merged with the JS Foundation to form the OpenJS Foundation. OpenJS Foundation is facilitated by the Linux Foundation's Collaborative Projects program.

### **2.3.2 Socket.IO**

Socket.IO is an event-driven library for real-time web applications. It enables real-time, bi-directional communication between web clients and servers. It consists of two parts: a client-side library that runs in the browser, and a server-side library for Node.js. Both components have a nearly identical API. Socket.IO provides the ability to implement real-time analytics, binary streaming, instant messaging, and document collaboration. Notable users include Microsoft Office, Yammer and Zendesk. Socket.IO handles the connection transparently and will automatically upgrade to WebSocket if possible. This means that the developer does not need to know how to use the WebSocket protocol in order to use Socket.IO.

Socket.IO is not a WebSocket library with fallback options to other real-time protocols. It is a custom real-time transport protocol implementation on top of other real-time protocols. A Socket.IO implementing server cannot connect to a non-Socket.IO WebSocket client. A Socket.IO implementing client cannot talk to a non-Socket.IO WebSocket or Long Polling Comet server. Socket.IO requires using the Socket.IO libraries on both client and server side.

### **2.3.3 HTML**

The Hypertext Markup Language or HTML is the standard markup language for documents designed to be displayed in a web browser. It is often assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for its appearance.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting

structural semantics for text such as headings, paragraphs, lists, links, quotes, and other items. HTML elements are delineated by *tags*, written using angle brackets. Tags such as `<img />` and `<input />` directly introduce content into the page. Other tags such as `<p>` and `</p>` surround and provide information about document text and may include sub-element tags. Browsers do not display the HTML tags but use them to interpret the content of the page.

HTML can embed programs written in a scripting language such as JavaScript, which affects the behavior and content of web pages. The inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), former maintainer of the HTML and current maintainer of the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997. A form of HTML, known as HTML5, is used to display video and audio, primarily using the `<canvas>` element, together with JavaScript.

#### **2.3.4 Cascading Style Sheets (CSS)**

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML or XML (including XML dialects such as SVG, MathML or XHTML). CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

CSS is designed to enable the separation of content and presentation, including layout, colors, and fonts. This separation can improve content accessibility; provide more flexibility and control in the specification of presentation characteristics; enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, which reduces complexity and repetition in the structural content; and enable the .css file to be cached to improve the page load speed between the pages that share the file and its formatting.

Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobile device.

#### **2.3.5 JavaScript**

JavaScript, often abbreviated as JS, is a programming language that is one of the core technologies of the World Wide Web, alongside HTML and CSS. As of 2022, 98% of websites use JavaScript on the client side for webpage behavior, often incorporating third-party libraries. All major web browsers have a dedicated JavaScript engine to execute the code on users' devices.

JavaScript is a high-level, often just-in-time compiled language that conforms to the ECMAScript standard. It has dynamic typing, prototype-based object-orientation, and first-class functions. It is multi-paradigm, supporting event-driven, functional, and imperative programming styles. It has application programming interfaces (APIs) for working with text, dates, regular expressions, standard data structures, and the Document Object Model (DOM).

The ECMAScript standard does not include any input/output (I/O), such as networking, storage, or graphics facilities. In practice, the web browser or other runtime system provides JavaScript APIs for I/O.

JavaScript engines were originally used only in web browsers, but are now core components of some servers and a variety of applications. The most popular runtime system for this usage is Node.js.

Although Java and JavaScript are similar in name, syntax, and respective standard libraries, the two languages are distinct and differ greatly in design.

### **2.3.6 Mongo DB**

MongoDB is a source-available cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas. MongoDB is developed by MongoDB Inc. and licensed under the Server Side Public License (SSPL) which is deemed non-free by several distributions. MongoDB is a member of the MACH Alliance.

### **2.3.7 Express.js**

Express.js, or simply Express, is a back end web application framework for building RESTful APIs with Node.js, released as free and open-source software under the MIT License. It is designed for building web applications and APIs. It has been called the de facto standard server framework for Node.js.

Express is the back-end component of popular development stacks like the MEAN, MERN or MEVN stack, together with the MongoDB database software and a JavaScript front-end framework or library.

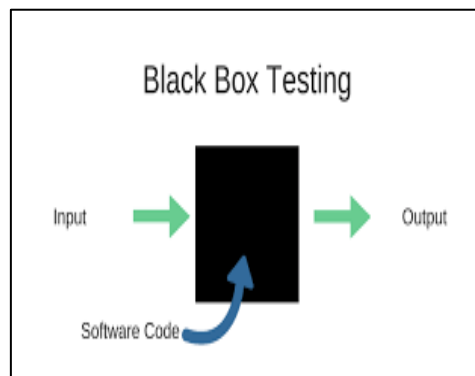
### **2.3.8 Git**

Git is a distributed version control system that tracks changes in any set of computer files, usually used for coordinating work among programmers collaboratively developing source code during software development. Its goals include speed, data integrity, and support for distributed, non-linear workflows (thousands of parallel branches running on different computers).

## CHAPTER- 3

### WORKING & TESTING

We are doing **black box testing** on this project. **Black Box Testing** is a software testing method in which the functionalities of software applications are tested without having knowledge of internal code structure, implementation details and internal paths. Black Box Testing mainly focuses on input and output of software applications and it is entirely based on software requirements and specifications. It is also known as Behavioral Testing.



(Fig. 1.1)

We performed manual System testing on a complete integrated system to evaluate the compliance of the system with all the requirements.

We tested the design and behavior of the system and also the expectations of the users.

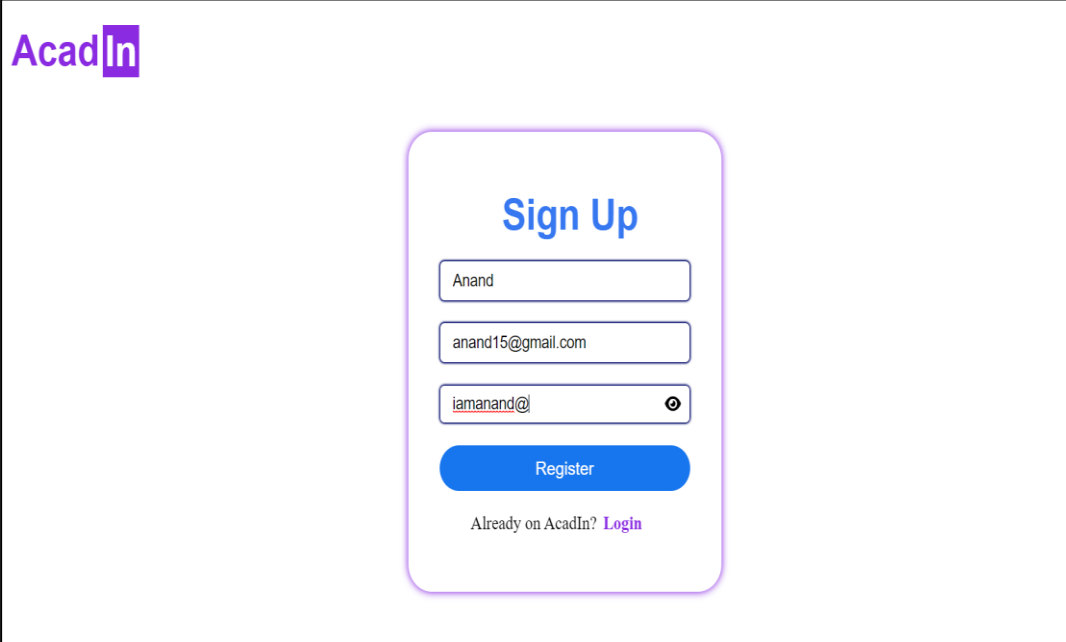
**1. Sign Up:** Users must first register themselves on AcadIn to create their individual accounts using their username, email id and a password.

**Case 1:** Successful Sign Up with New Email

Expected Output: Account Created and successfully logged in.

Actual Output:



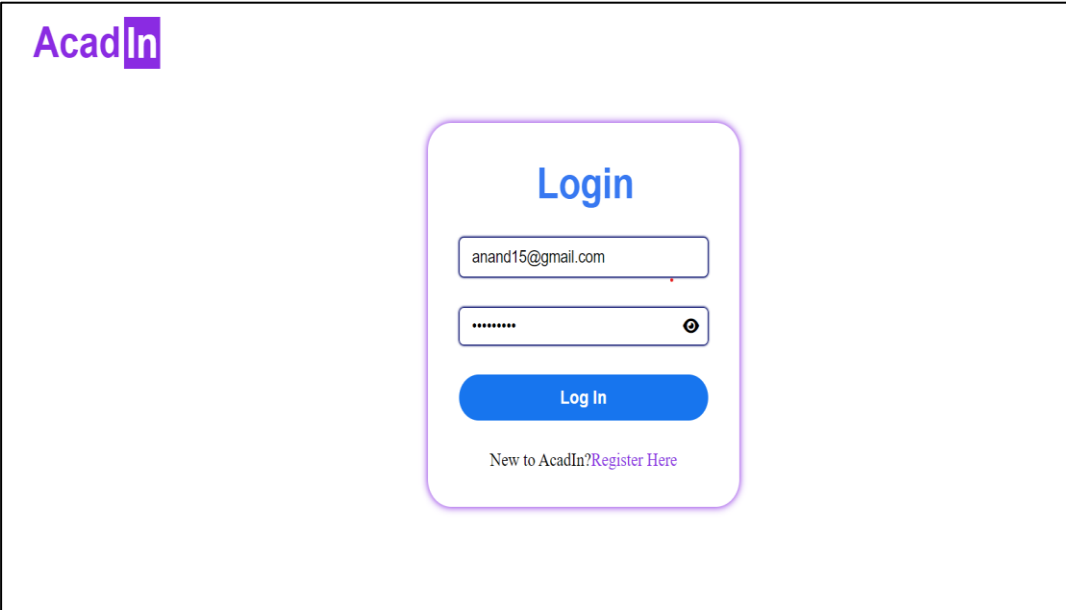


The image shows the 'Sign Up' form on the AcadIn website. The AcadIn logo is in the top left corner. The form is centered and has a light blue border. It contains the following elements:

- Sign Up** (Title)
- (Name field)
- (Email field)
- (Password field) with a toggle icon on the right.
- (Register button)
- Already on AcadIn? [Login](#) (Link)

(Fig. 1.2)

Successful Login with new account:



The image shows the 'Login' form on the AcadIn website. The AcadIn logo is in the top left corner. The form is centered and has a light blue border. It contains the following elements:

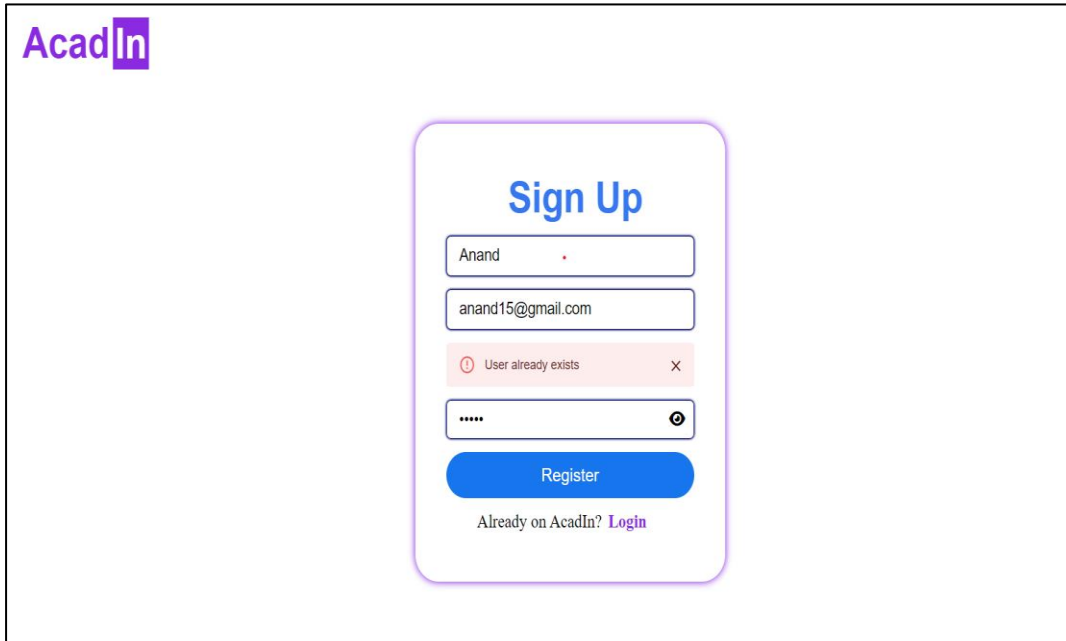
- Login** (Title)
- (Email field)
- (Password field) with a toggle icon on the right.
- (Log In button)
- New to AcadIn? [Register Here](#) (Link)

(Fig. 1.3)

**Case 2:** Sign Up with an existing email

Expected Output: User Already Exist.

Actual Output:

The image shows a web application interface for 'AcadIn'. In the top left corner is the 'AcadIn' logo. Centered on the page is a 'Sign Up' form. The form has a title 'Sign Up' in blue. It contains three input fields: a username field with 'Anand' and a red asterisk, an email field with 'anand15@gmail.com', and a password field with four dots and an eye icon. Below the email field is a red error message box that says 'User already exists' with a red circle icon and a close 'X' button. At the bottom of the form is a blue 'Register' button. Below the button is a link that says 'Already on AcadIn? Login'.

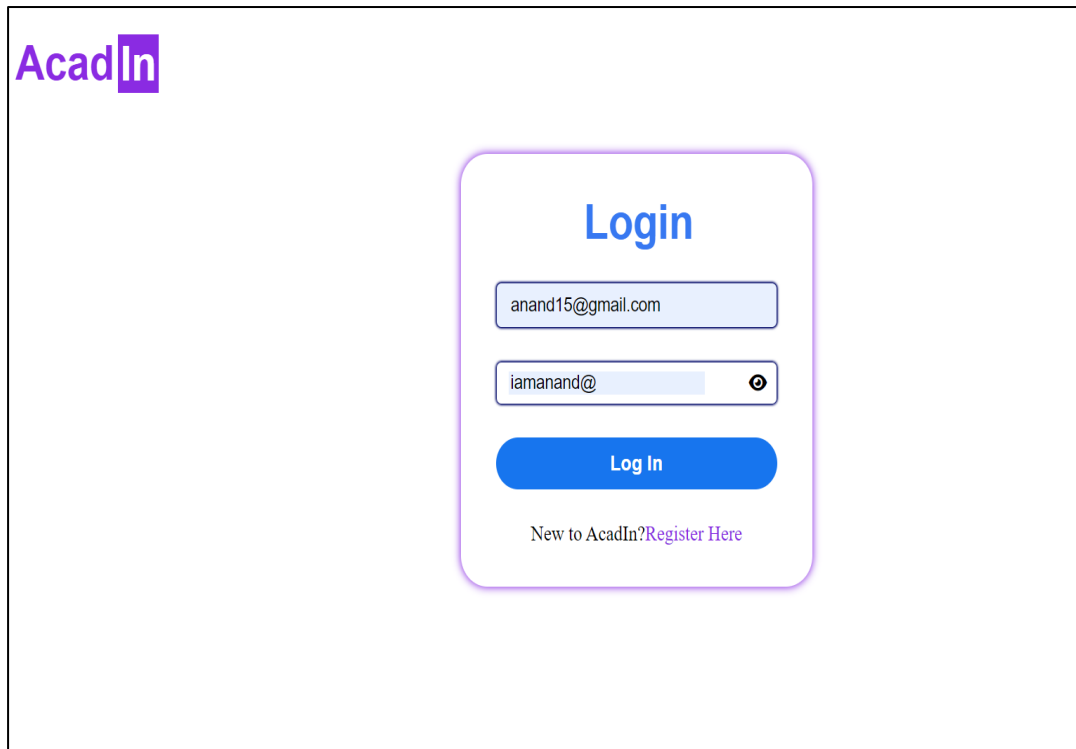
(Fig. 1.4)

**2. Sign In:** In order to successfully login to their account, users must provide the correct username and password.

**Case 1:** Successful Login with correct Email and password

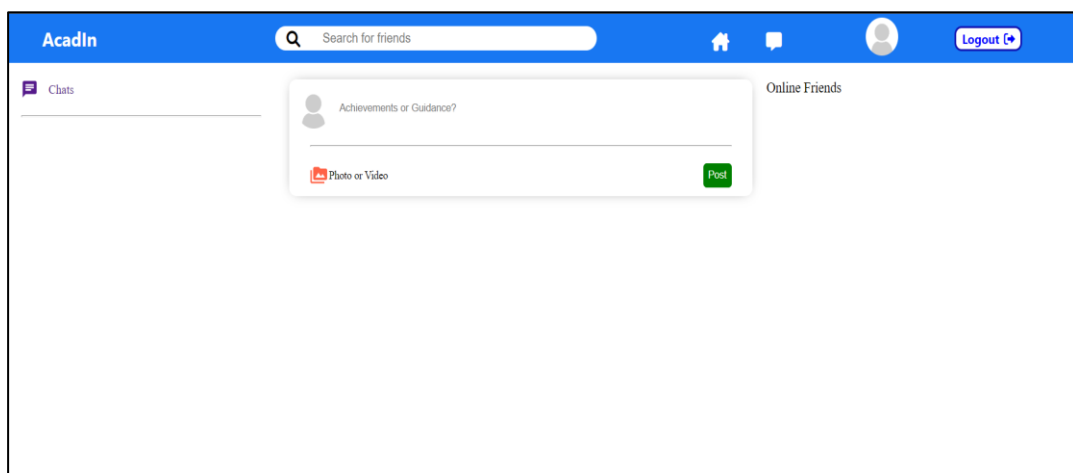
Expected output - Successful login

Actual output:



(Fig. 2.1)

Successful Login with new account:

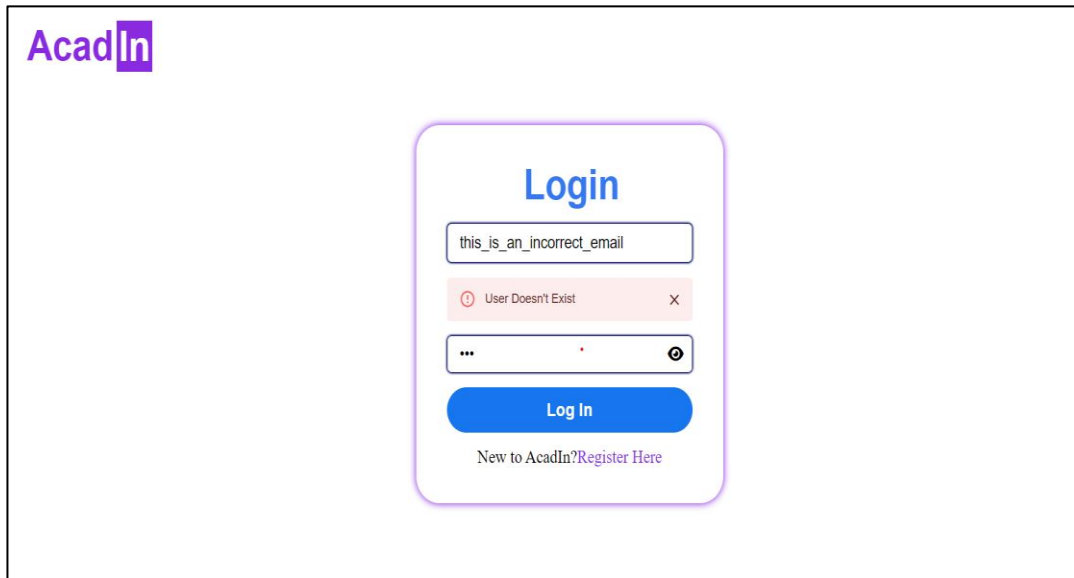


(Fig. 2.2)

**Case 2: Login with Incorrect Email**

Expected output - User Doesn't Exist

Actual output -



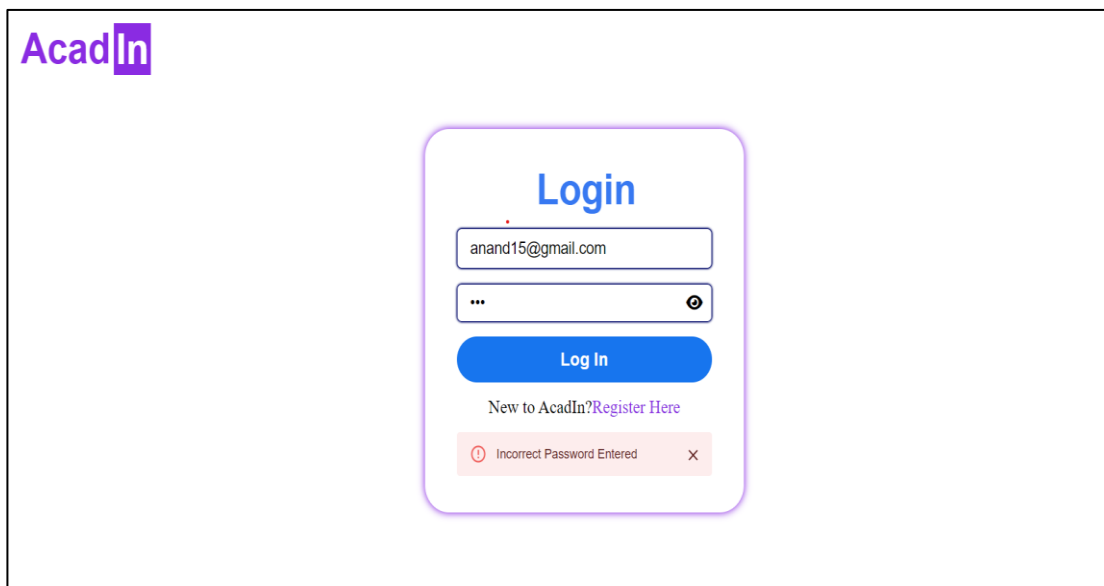
The screenshot shows the AcadIn login interface. At the top left is the AcadIn logo. The main content is a rounded rectangle with a purple border. Inside, the word "Login" is at the top. Below it is a text input field containing "this\_is\_an\_incorrect\_email". Under the input field is a red error message box that says "User Doesn't Exist" with a red circle icon and a close button (X). Below the error box is a password input field with a red asterisk and a toggle icon. At the bottom of the login box is a blue "Log In" button. Below the button is a link that says "New to AcadIn? Register Here".

(Fig. 2.3)

**Case 3: Login with Correct Email but Incorrect Password**

Expected output - Incorrect Password Entered

Actual output -



The screenshot shows the AcadIn login interface. At the top left is the AcadIn logo. The main content is a rounded rectangle with a purple border. Inside, the word "Login" is at the top. Below it is a text input field containing "anand15@gmail.com". Under the input field is a password input field with a red asterisk and a toggle icon. At the bottom of the login box is a blue "Log In" button. Below the button is a link that says "New to AcadIn? Register Here". At the very bottom of the login box is a red error message box that says "Incorrect Password Entered" with a red circle icon and a close button (X).

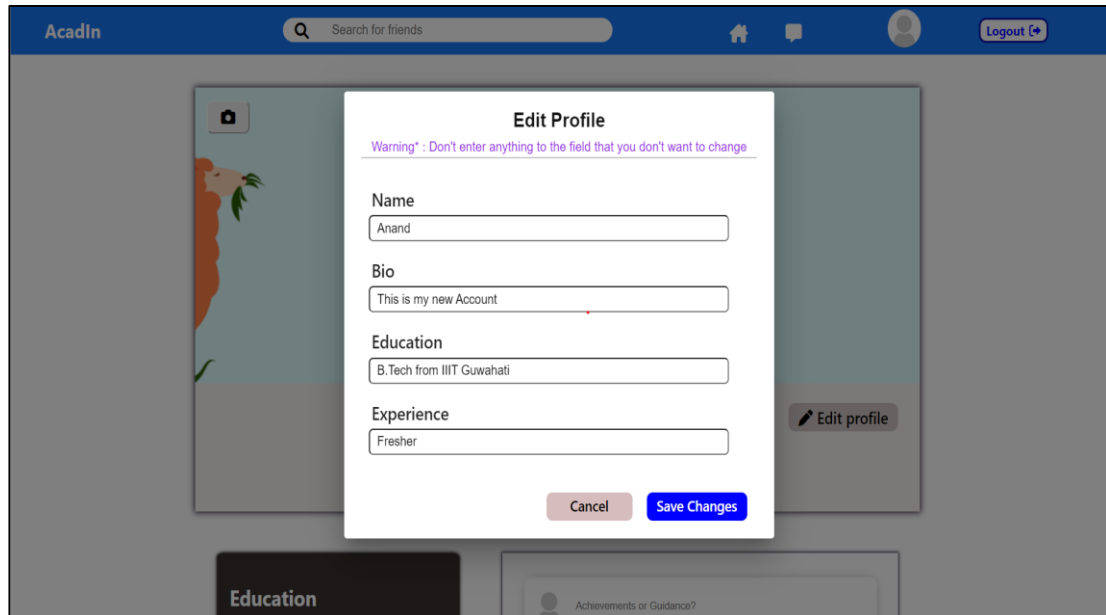
(Fig. 2.4)

**3. Profile:** Users can add education, experience, bio and profile picture to their profile.

**Case 1:** Adding experience, education and bio.

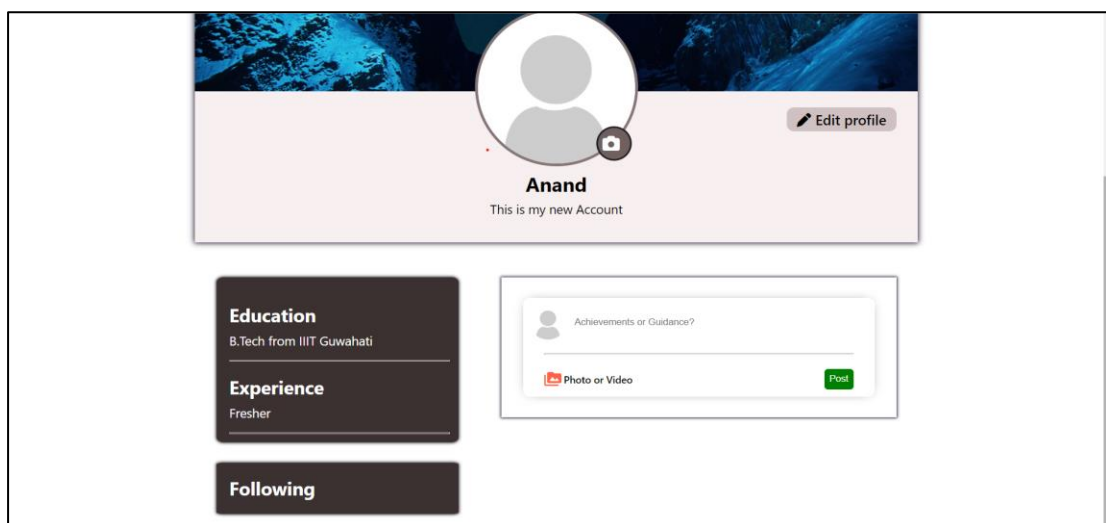
Expected Output: Added details are visible on their profile page.

Actual Output:



(Fig. 3.1)

Profile Page after this update:

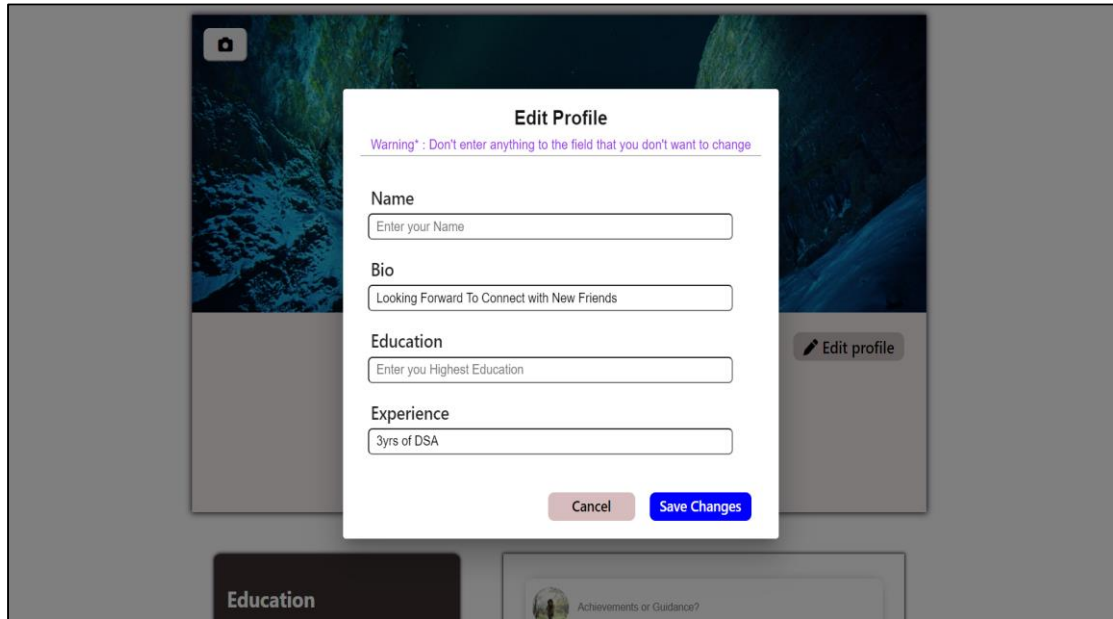


(Fig. 3.2)

**Case 2:** Updating experience, education and bio.

Expected Output: Added details are visible on their profile page.

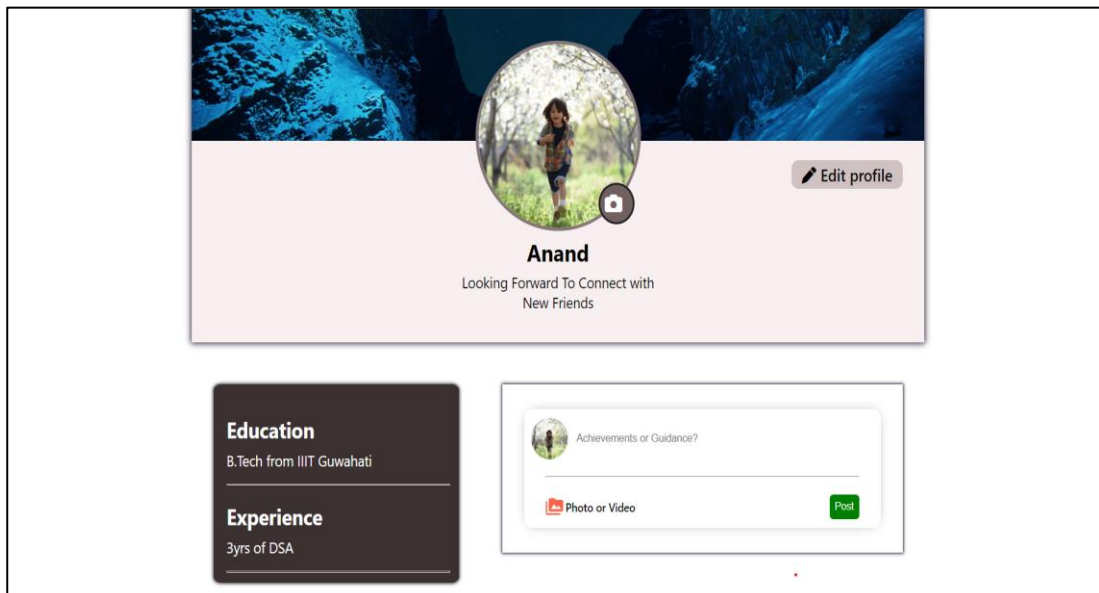
Actual Output:



The screenshot shows a modal window titled "Edit Profile" overlaid on a blurred background of a user's profile page. The modal contains a warning message: "Warning\* : Don't enter anything to the field that you don't want to change". Below this, there are four input fields: "Name" (placeholder: "Enter your Name"), "Bio" (placeholder: "Looking Forward To Connect with New Friends"), "Education" (placeholder: "Enter you Highest Education"), and "Experience" (placeholder: "3yrs of DSA"). At the bottom of the modal are two buttons: "Cancel" and "Save Changes".

(Fig. 3.3)

Profile Page after this update:



The screenshot shows the user's profile page after the update. The profile card at the top features a circular profile picture of a person in a field, the name "Anand", and the bio "Looking Forward To Connect with New Friends". To the right of the profile card is an "Edit profile" button. Below the profile card, there are two sections: "Education" and "Experience". The "Education" section shows "B.Tech from IIIT Guwahati". The "Experience" section shows "3yrs of DSA". To the right of these sections is a post creation area with a placeholder image, the text "Achievements or Guidance?", a "Photo or Video" button, and a "Post" button.

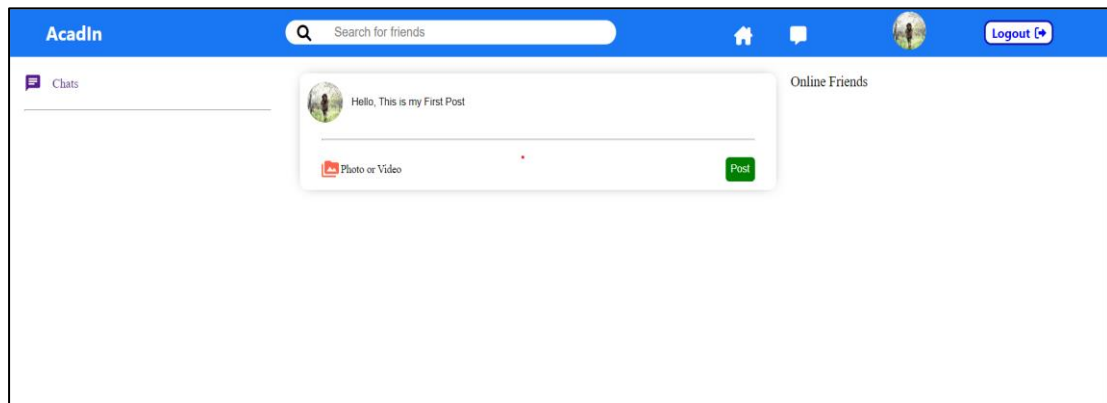
(Fig. 3.4)

**4. Post:** Users can post anything they want and that will be visible to all the people they follow.

**Case 1:** Adding only text in the post

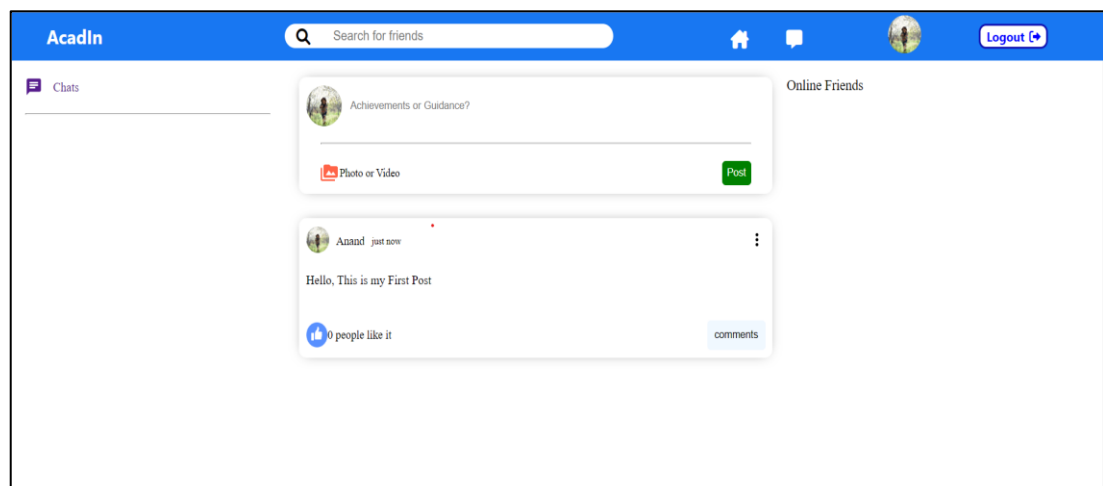
Expected Output: Post is being displayed on the home page and profile page

Actual Output:



(Fig. 4.1)

Post added as shown on Home Page:

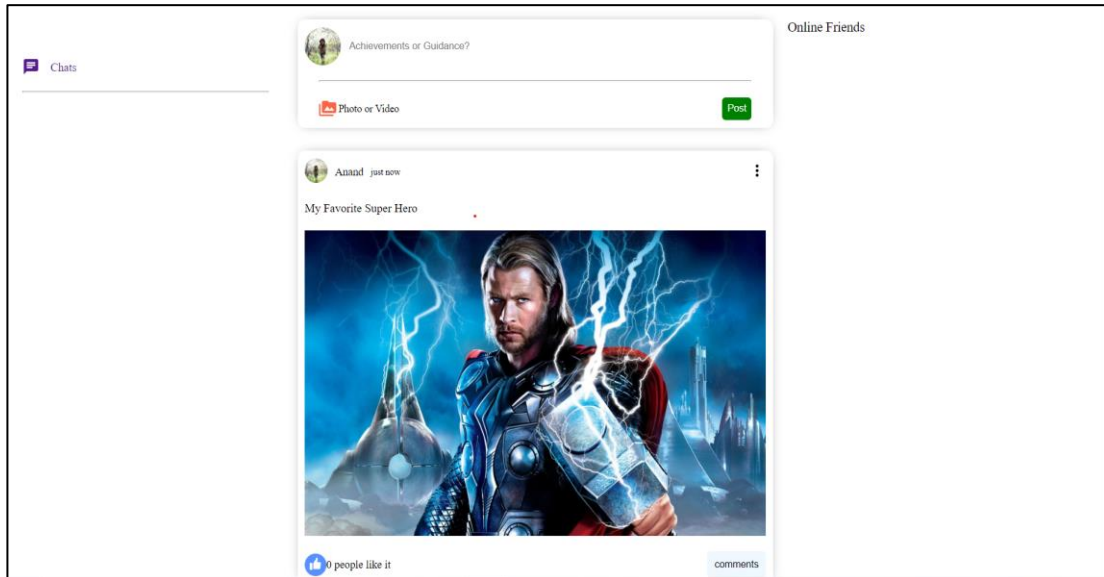


(Fig. 4.2)

**Case 2:** User adds photo along with text in the post.

Expected Output: Post is being displayed on the home page and profile page

Actual Output:



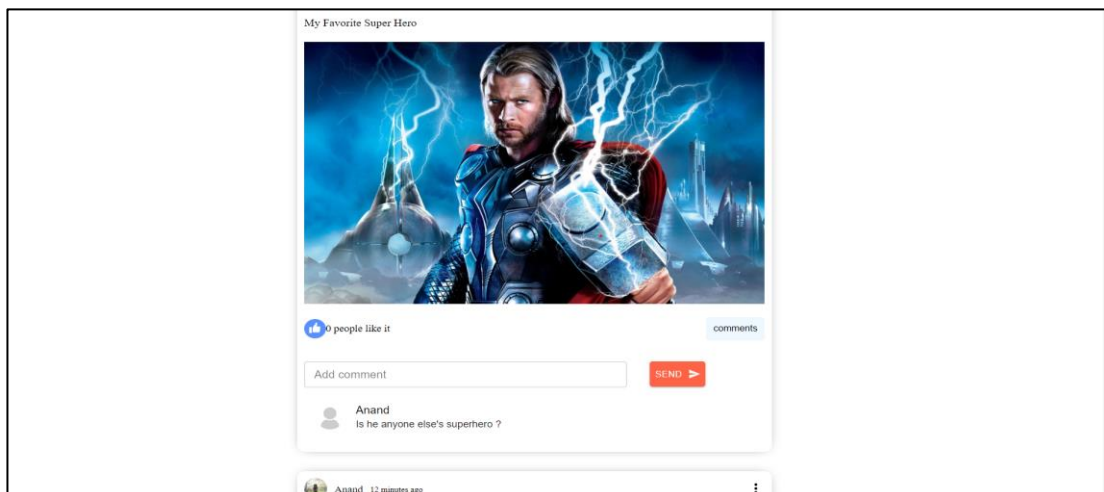
(Fig. 4.3)

**Case 3:** User can like or comment on his post.

Expected output: Like and comment gets displayed on the post.

Actual output:

Post after Comment and Like:



(Fig. 4.4)

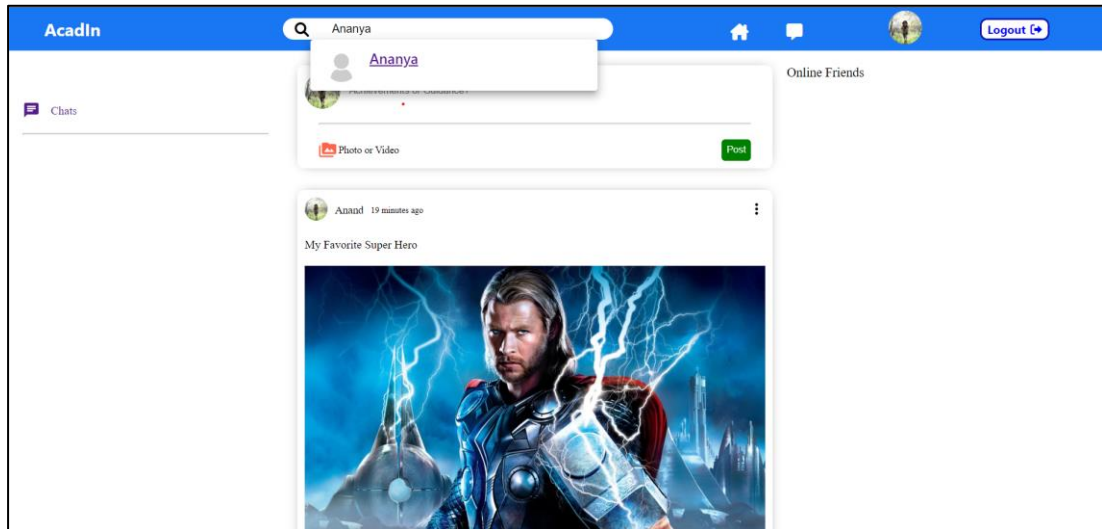


**5. Friend:** User can search for his or her friend, view their profile, follow, unfollow and can see the friend's posts after following them.

**Case 1:** User searches for his or her existing friend.

Expected Output: Shows the profile link of the Friend

Actual Output:

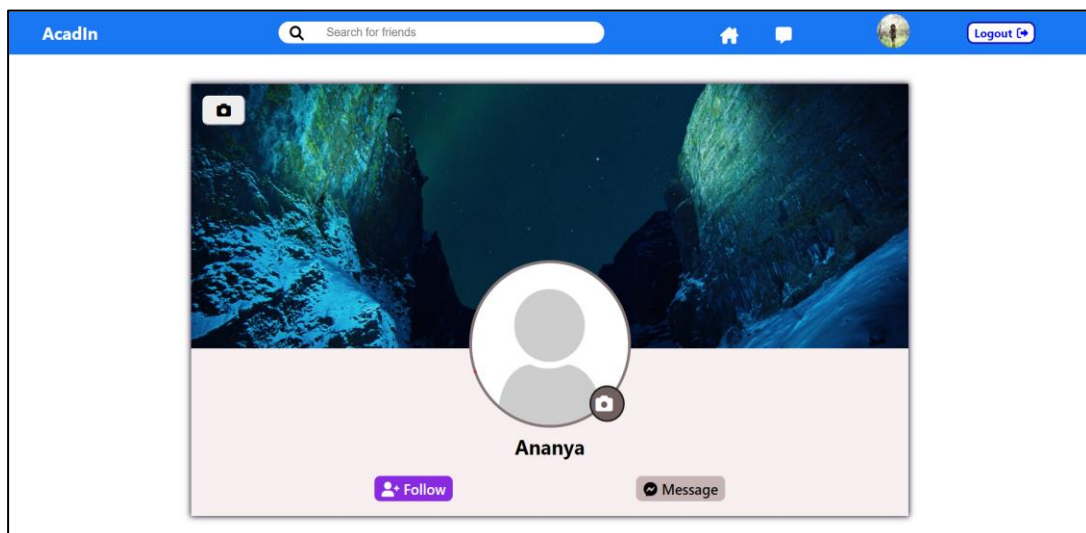


(Fig. 5.1)

**Case 2:** User views friend's profile

Expected Output: Shows the profile of the Friend

Actual Output:

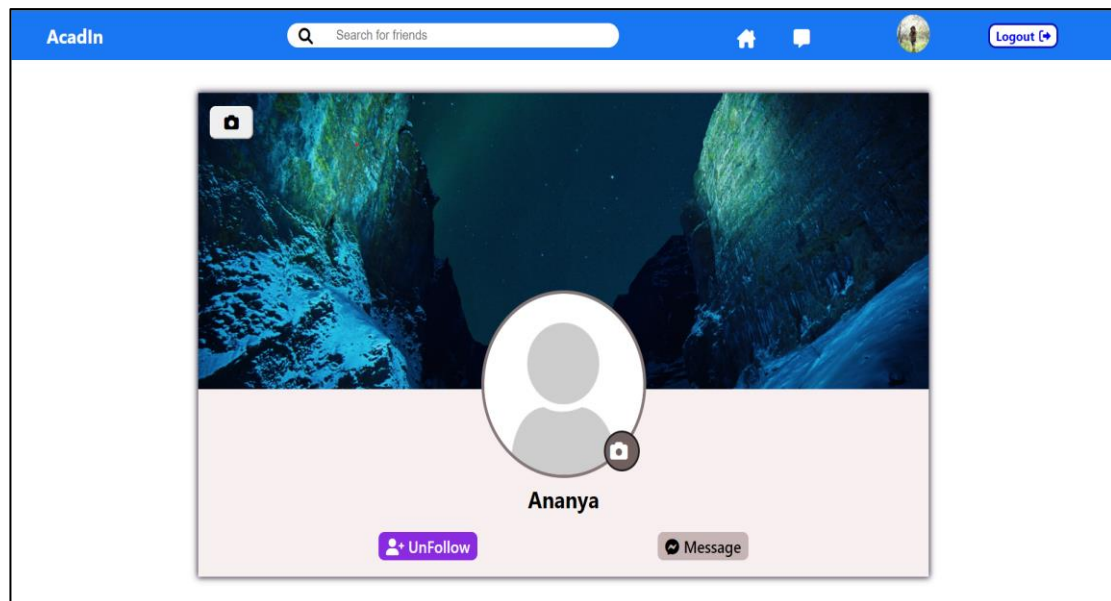


(Fig. 5.2)

**Case 3:** User can follow his or her friend.

Expected output: User follows that friend.

Actual output:

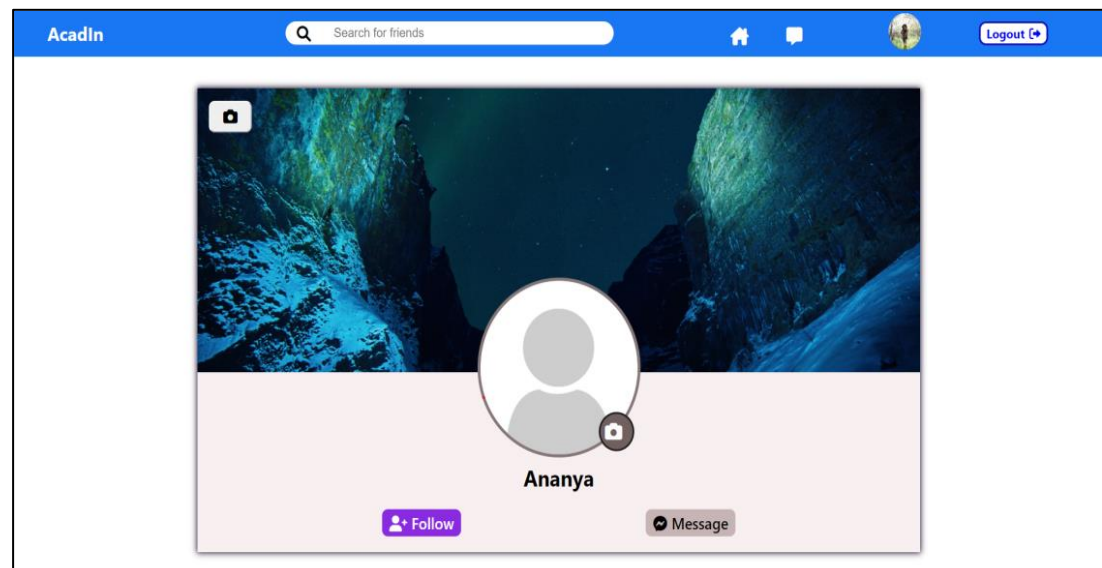


(Fig. 5.3)

**Case 4:** User can unfollow his or her friend.

Expected output: User unfollows that friend. Icon reverts back to follow.

Actual Output:



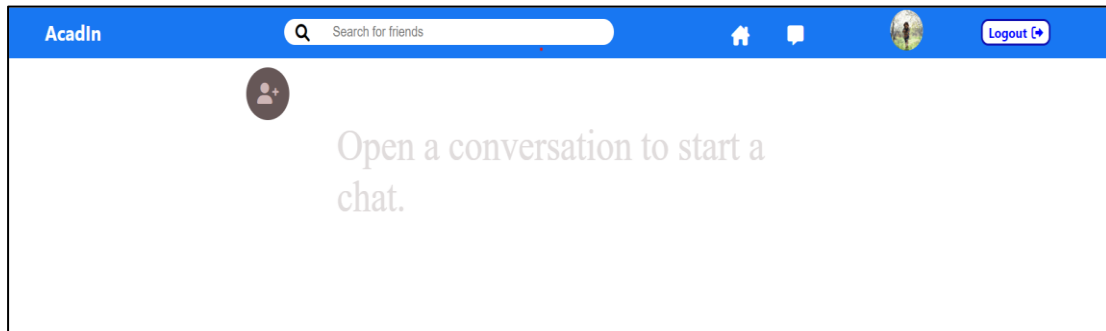
(Fig. 5.4)

**6. Chat:** Users can chat with their friends.

**Case 1:** User adds a friend to chat.

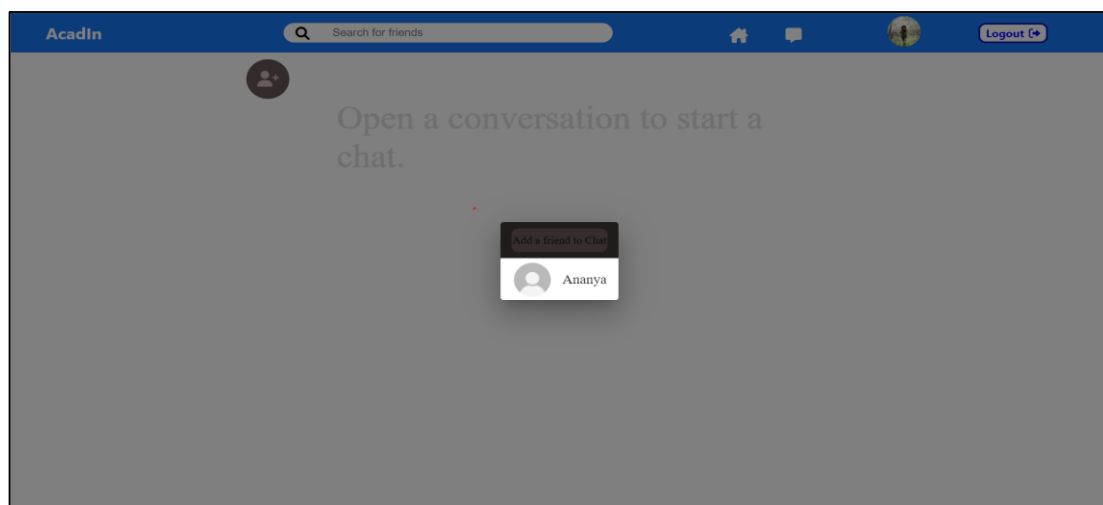
Expected output: Friend is added and now can be messaged.

Actual output:



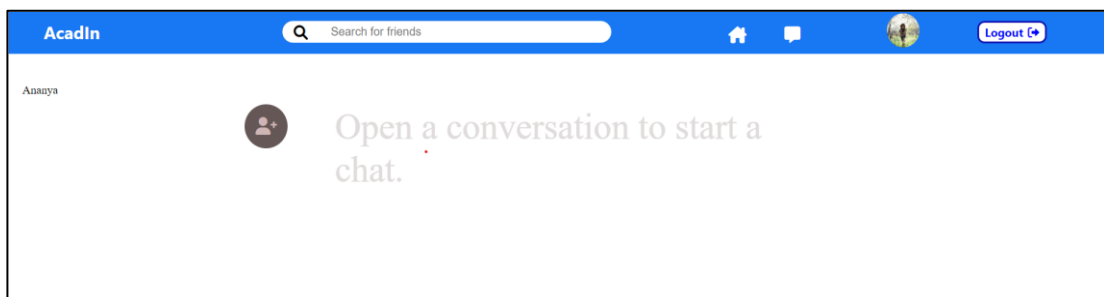
(Fig. 6.1)

After Click on Add User Icon:



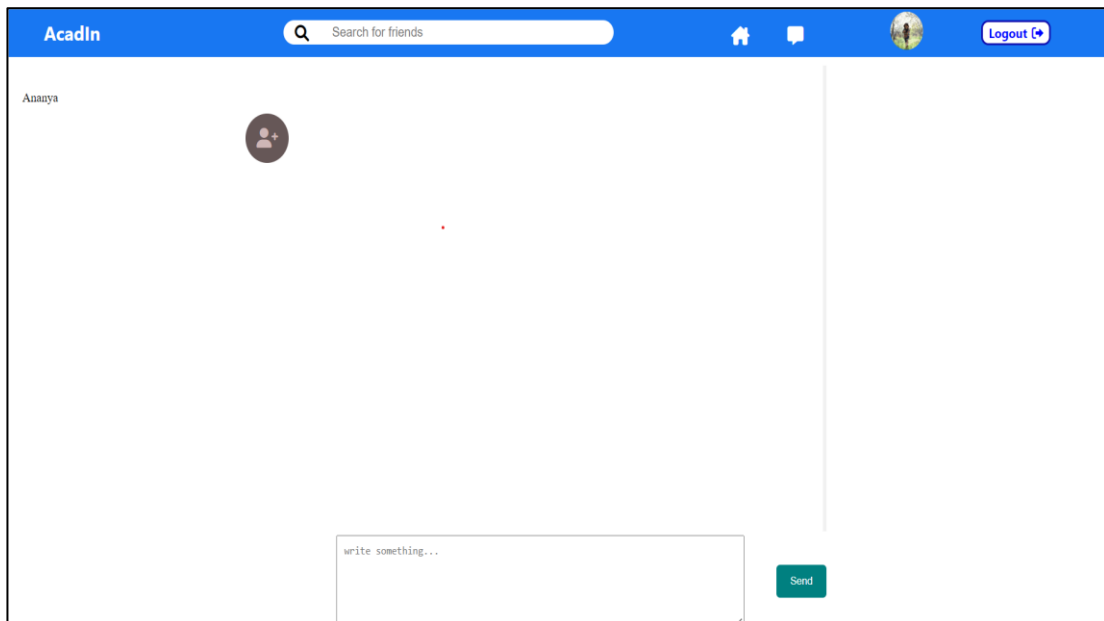
(Fig. 6.2)

After Clicking on user that he/she want to add :



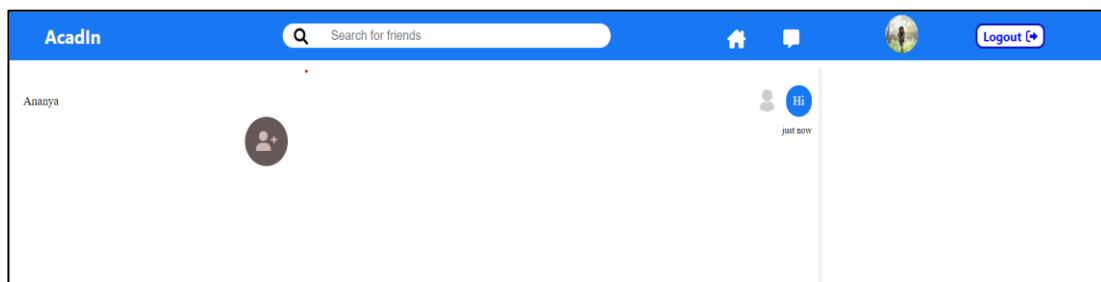
(Fig. 6.3)

After Clicking on user that was added:



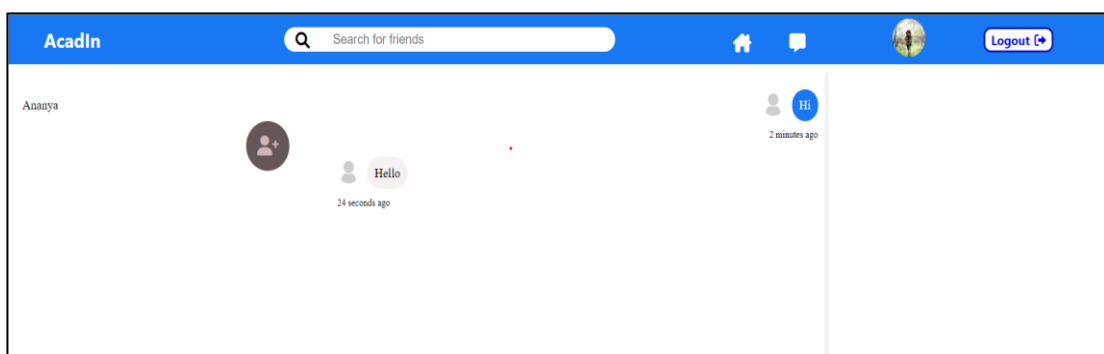
(Fig. 6.4)

After sending “Hi” message to the user:



(Fig. 6.5)

After getting reply from the user:



(Fig. 6.6)

## **CHAPTER- 4**

### **HARDWARE & SOFTWARE USED**

#### **4.1 Application Development**

This section presents the Implementation of the “AcadIn” web application. Based on the design described in the previous section, the code written and the implementation techniques used are presented.

#### **4.2 Tools**

##### **4.2.1 Software: -**

- Database Server – MongoDB
- Client - Web Browser
- Development Tools - Visual Studio Code, GitHub
- Framework - React.js, Socket.io, Express.js
- Programming Languages – JavaScript
- Environment - Node.js
- Version Control – Git

##### **4.2.2 Hardware: -**

- Microsoft Windows 7/8/10 (32 or 64 bit)
- 2 GB RAM minimum, 8 GB recommended
- 2 GB of available disk space minimum, 4 GB recommended

## **ADVANTAGES**

- Single application for students to collaborate, have fun and discuss.
- Provide students with a social media app to see what's happening in their community.

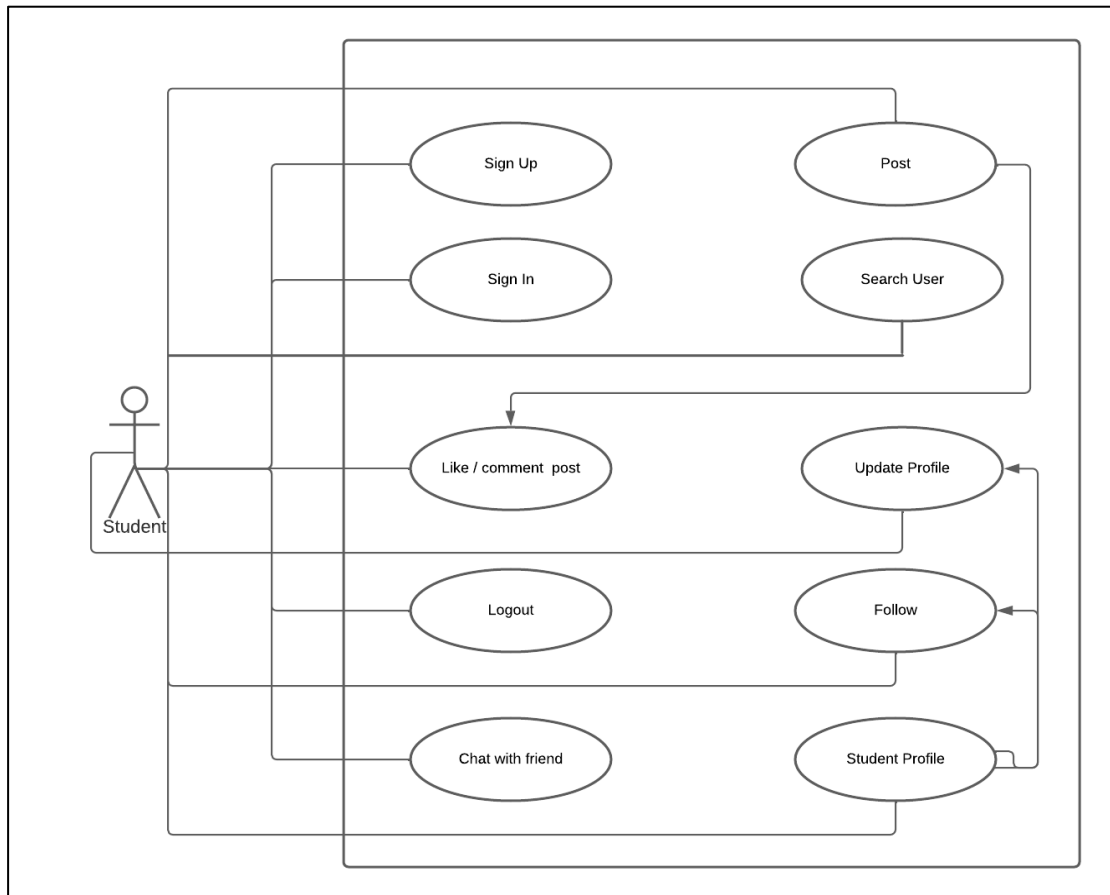
## **LIMITATIONS**

- Require GPRS connection every time.
- Tons of spam messages.
- Risk of identity theft.

## **CHAPTER- 5**

### **ENTITY - RELATIONSHIP MODEL**

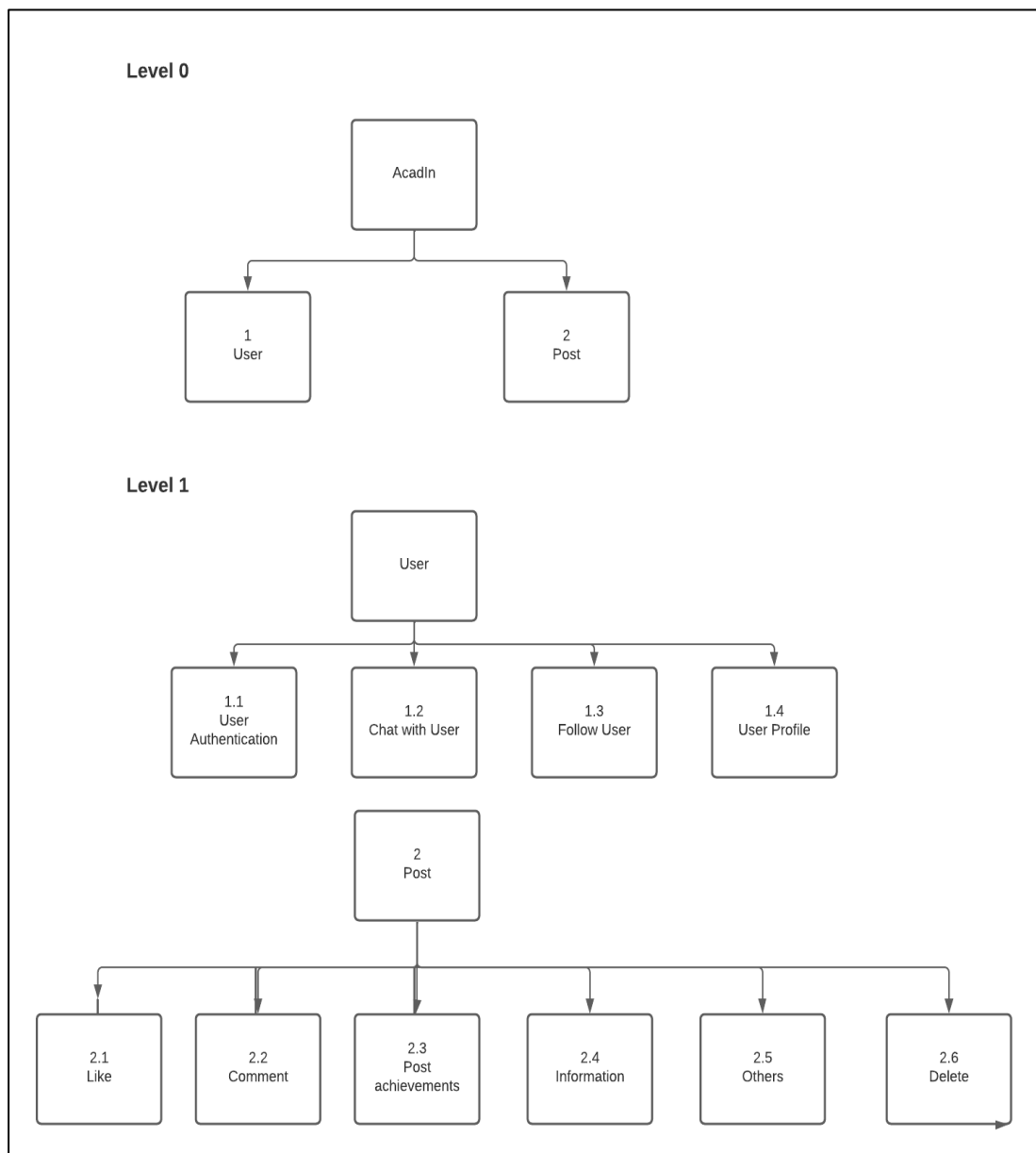
An entity–relationship model describes interrelated things of interest in a specific domain of knowledge. A basic ER model is composed of entity types and specifies relationships that can exist between entities.



(Fig. 7.1)

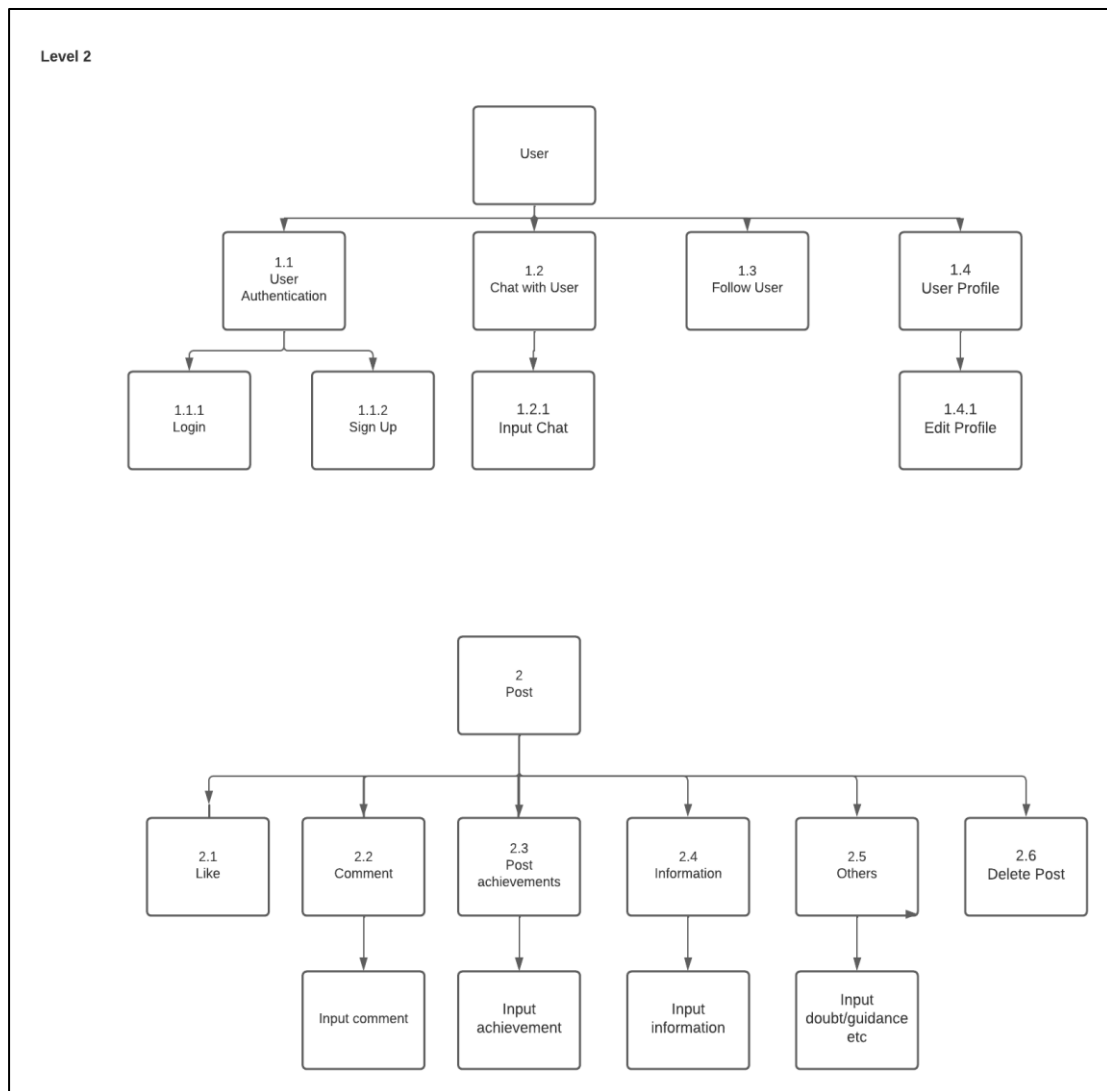
## **CHAPTER- 6**

### **DFD LEVELS**



(Fig. 8.1)

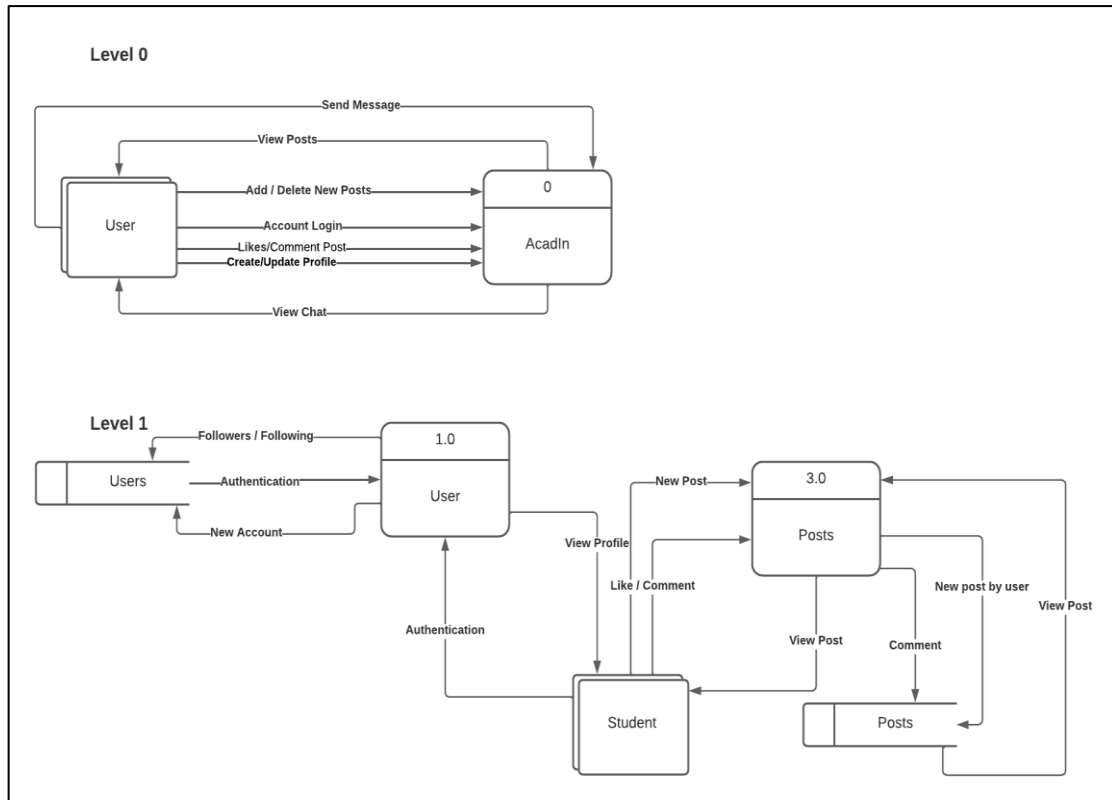




(Fig. 8.2)

## CHAPTER- 7

### DFD LEVELS: CONTEXT DIAGRAM



(Fig. 9.1)

## **CHAPTER- 8**

## **CONCLUSION**

In conclusion, the sole purpose of the project is to enable the student community of the institute come closer. It should act as a platform where students can share their knowledge and learning so that the entire community can benefit from it.

There is always some place for enhancements in any software application, however good and efficient the application may be.

In this modern age, the social media has become an important tool for communicating with others. These social platforms also are essential for businesses and their marketing strategies today. This is because they supply a cheap and effective manner for reaching bent many people in their audience. In the future, it's expected that the expansion of the social media platforms (and their features) will make it increasingly difficult for businesses to stay track of their social media metrics. However, brands have generally evolved because the social media has grown and it's expected that if these brands keep themselves updated with the newest trends, they're going to be successful in the future as well.

## **CHAPTER- 9**

### **FUTURE SCOPE**

- The proposed system is specifically designed for the students of IILM but can be customized for other institutes and organizations.
- Profile insights - Use your AcadIn insights to learn more about overall trends across your followers and your content's performance with your audience. You can also view insights for specific posts, Stories, videos, reels, and Live videos you've created to see how each one performed and how people engaged with them. Insights are free and available only for business or creator accounts.  
You can view these Insights from your profile in the AcadIn app:
  - Recent highlights.
  - Insights overview, which includes Accounts reached, Accounts engaged and Total followers.
  - Content you shared
- Who viewed your profile - The AcadIn Premium account will have the following who's viewed your profile features:
  - You'll see the entire list of viewers from the past 90 days, if you have at least one viewer in the past 90 days, you'll also see the viewer trends and insights.
  - In addition to the free account experience, you'll be able to see weekly viewer insights.
- Group conversation.
- Video conferencing.

## **REFERENCES**

1. Henriyan, D., Subiyanti, D. P., Fauzian, R., Anggraini, D., Aziz, M. V. G., & Prihatmanto, A. S. (2016, October). Design and implementation of web based real time chat interfacing server. In *2016 6th International Conference on System Engineering and Technology (ICSET)* (pp. 83-87). IEEE.
2. Királ'ová, A., & Pavlíčka, A. (2015). Development of social media strategies in tourism destination. *Procedia-Social and Behavioral Sciences*, 175, 358-366.
3. Çakır, R., & Tan, S. S. (2017). Development of educational applications on the social network of facebook and its effects on students' academic achievement. *Educational Sciences: Theory & Practice*, 17(5).
4. Satheesh, M., D'mello, B. J., & Krol, J. (2015). *Web development with MongoDB and NodeJs*. Packt Publishing Ltd.
5. Wang, R., Chen, S., & Wang, X. (2012, May). Signing me onto your accounts through facebook and google: A traffic-guided security study of commercially deployed single-sign-on web services. In *2012 IEEE Symposium on Security and Privacy* (pp. 365-379). IEEE.
6. <https://www.w3schools.com/html/>
7. <https://www.w3schools.com/css/>
8. <https://www.w3schools.com/js/>
9. <https://nodejs.org/en/docs/guides>
10. <https://expressjs.com/en/guide/routing.html>
11. <https://www.w3schools.com/mongodb/>
12. <https://github.com/belikeamitesh/AcadLink>