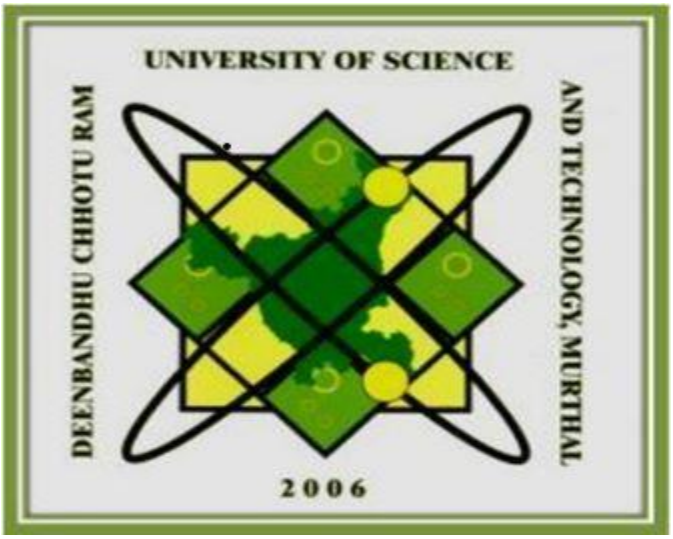
Project Report

on

Social Networking Website Leechi

Bachelor of Technology
in

Computer Science & Engineering



Submitted To
Dr. Kavita Rathi

Submitted by
Rajeev Kumar
18001001044

Department of Computer Science & Engineering
Deenbandhu Chhotu Ram University of Science and Technology

ACKNOWLEDGEMENT

A successful task makes everyone happy. Success will often be crowned to people who made it reality but the people who are behind curtain with constant guidance and encouragement that made it possible will be crowned first on the eve of success. Words are inadequate to express my deep sense of gratitude towards all those people behind the screen who guided, inspired and helped me for the completion of our project work. The successful completion of the project on "SOCIAL NETWORKING" which I have undertaken has a partial fulfillment of the requirements for the award of Bachelor of technology degree in Computer Science and Engineering. It is with profound sense of gratitude that I acknowledge my project guide Dr. Kavita Rathi for providing me with live specification and his valuable suggestion which encouraged me to complete this project successfully. I thank our Dr. Kavita Rathi for permitting us to do this project. At last but not the least I thank entire Computer Science department who rendered their full cooperation for successful completion of the project.

(Rajeev Kumar)

INDEX

S. No.	Contents	Page no.	Signature
1	Introduction to the project	4	
2	Study and analysis of Existing System	6	
3	Proposed System	7	
4	Hardware and Software Requirement	10	
5	Which Technology is Used and Why?	11	
6	Services Used	14	
7	Library Used	14	
8	Deployment Services	16	
9	Main Component & Features	17	
10	DFD &Use Case Diagram	18	
11	Snapshots	22	
12	References	26	

Introduction

SOCIAL NETWORKING is a website which provides the up-to date information of all the students registered into it. It is effectively used for knowing the people of different groups. We can update our personal details. They can wish their friends by sending E-Cards. Technology for building applications for generating dynamic web content, such as HTML,CSS, Reactjs,Saas,Material UI Lib,Node js,etc.

A website succeeds-when it meets the needs of the people who use it, when it performs flawlessly over a long period of time, when it is easy to modify and even easier to use-it can and does change things for the better. But when software fails-when its users are dissatisfied, when it is error prone, when it is difficult to change and even harder to use-bad things can and do happen.

We all want to create websites that makes things better, avoiding the bad things that lurk in the shadow of failed efforts. To succeed we need discipline when software is designed and built. Many individuals and companies still develop software haphazardly, even as they build systems to service the most advanced technologies of the day. As a result, the quality of the software that we produce suffers and bad things happen.

This project report is intended to serve as a guide to the website developed on Social Networking. I have tried to follow the principles and rules as suggested by the software engineers as far as possible, in order to make this website a successful one. The report starts with a comprehensive introduction to the project undertaken as its very First Section, It includes about the Study and analysis of Existing System used. The second part presents and discusses the theoretical Proposed System. The third section focuses on Hardware and Software Requirements. The fourth part is the Which technology and Why?. This part focuses on requirements analysis and specification, analysis issues Of Technology. In The fifth section, given about services and libraries used. The sixth section highlights about deployment adopted for this project. The seventh part suggests the main component and features. The eighth part discusses about the use case diagram,data-flow-diagram, user-flow-diagram. The ninth part shows the schema or format of storing the data in backend . The tenth section contains the snapshots of the project.

The final section is the reference part that contains a list of the websites and documentation that were referred during the development of the project and the report as well. The emphasis in this report is to document the important concepts and techniques used for the successful development of this project.I do hope that, through this report, the readers will get a real picture of what the project is all about. I also wish that may this website satisfies all the needs and requirements of the us

Study and analysis of Existing System

Existing System: The existing system for Leechi is the current social networking giant-The Facebook.

Study of Existing System: Facebook is a social networking service launched in February 2004, owned and operated by Facebook It was founded by Mark Zuckerberg with his college roommates and fellow Harvard University students Eduardo Saverin, Azel Andrew McCollum, Dustin Moskovitz and Chris Hughes. The website's membership was initially limited by the founders to Harvard students, but was expanded to other colleges in the Boston area, the Ivy League, and gradually most universities in Canada and the United States, corporations, and by September 2006, to everyone of age 13 and older with a valid email address.
Current Stats: The current presence of Facebook is all over the world. Facebook is ranked 2nd in the internet world next only to Google and have a record 1.15 billion active users till March 2013. **Platform:** Facebook is developed using C++ and PHP. The database used by Facebook is MySQL. **Issues with Existing System:** The various issues with the existing

System are as follows:

1. Compatibility of platform and database server*:* The platform used to develop Facebook is PHP while database server used is MySQL. The compatibility between PHP and MySQL server is known to have certain issues when used with large data regarding speed of operation.

2. We cannot Find nearby Friends on facebooks

Proposed System

The proposed system aims at overcoming the pitfalls of the existing system then replicating the major features of the same gradually. The major features of the proposed system are:

Each specific type of NoSQL database has different strengths, but all share fundamental characteristics that allow them to:

- Handle large volumes of data at high speed with a scale-out architecture
- Store unstructured, semi-structured, or structured data
- Enable easy updates to schemas and fields
- Be developer-friendly
- Take full advantage of the cloud to deliver zero downtime
- Map Feature With near By friends

Handle Large Volumes of Data at High Speed with a Scale-Out Architecture

SQL databases are most often implemented in a scale-up architecture, which is based on using ever larger computers with more CPUs and more memory to improve performance.

NoSQL databases were created in Internet and cloud computing eras that made it possible to more easily implement a scale-out architecture. In a scale-out architecture, scalability is achieved by spreading the storage of data and the work to process the data over a large cluster of computers. To increase capacity, more computers are added to the cluster.

This scale-out architecture is particularly painless to implement in cloud computing environments where new computers and storage can be easily added to a cluster.

The scale-out architecture of NoSQL systems provides a clear path to scalability when data volume or traffic grows. Achieving the same type of scalability with SQL databases can be expensive, require lots of engineering, or may not be feasible.

Store Unstructured, Semi-Structured, or Structured Data

Relational databases store data in structured tables that have a predefined schema. To use relational databases, a data model must be designed and then the data is transformed and loaded into the database.

When data is used in applications, the data then must be retrieved using SQL, and adapted to the form used in the application. Then when the data is written back, it must be transformed again back into the relational tables.

NoSQL databases have proven popular because they allow the data to be stored in ways that are easier to understand or closer to the way the data is used by applications. Fewer transformations are required when the data is stored or retrieved for use. Many different types of data, whether structured, unstructured, or semi-structured, can be stored and retrieved more easily.

In addition, the schemas of many NoSQL databases are flexible and under the control of the developers, making it easier to adapt the database to new forms of data. This removes bottlenecks in the development process associated with asking a database administrator to redesign a SQL database.

NoSQL databases support widely used data formats:

- Big data of all kinds -- text data as well as time-series data
- JSON files, which are nested human-readable files consisting of names and value pairs. This format can capture highly complex parent-child hierarchical structures, which can be efficiently stored in document databases
- Simple binary values, lists, maps, and strings can be handled at high speed in key-value stores
- Sparse data can be efficiently stored in columnar databases, where null values take up no room at all. They are also effective for information that does not change frequently (nonvolatile data)
- Networks of interrelated information can be stored in graph databases.

Enable Easy Updates to Schema and Fields

NoSQL databases have become popular because they store data in simple straightforward forms that can be easier to understand than the type of data models used in SQL databases.

In addition, NoSQL databases often allow developers to directly change the structure of the data.

- Document databases don't have a set data structure to start with, so a new document type can be stored just as easily as what is currently being stored.
- With key-value and column-oriented stores, new values and new columns can be added without disrupting the current structure.
- In response to new kinds of data, graph database developers add nodes with new properties and arcs with new meanings.

Developer-Friendly

Adoption of NoSQL databases has primarily been driven by uptake from developers who find it easier to create various types of applications compared to using relational databases.

Hardware Requirements:

For development:

- 8 GB Ram
- 1.5 GHz processor or more
- Internet Connection (1Mbps broadband connection is recommended for smooth operation).

For Deployment:

- 512 MB Ram
- Any processor with operating frequency not less than 1GHz.
- Internet connection (256Kbps broadband connection is recommended for smooth operation).

Software Requirements :

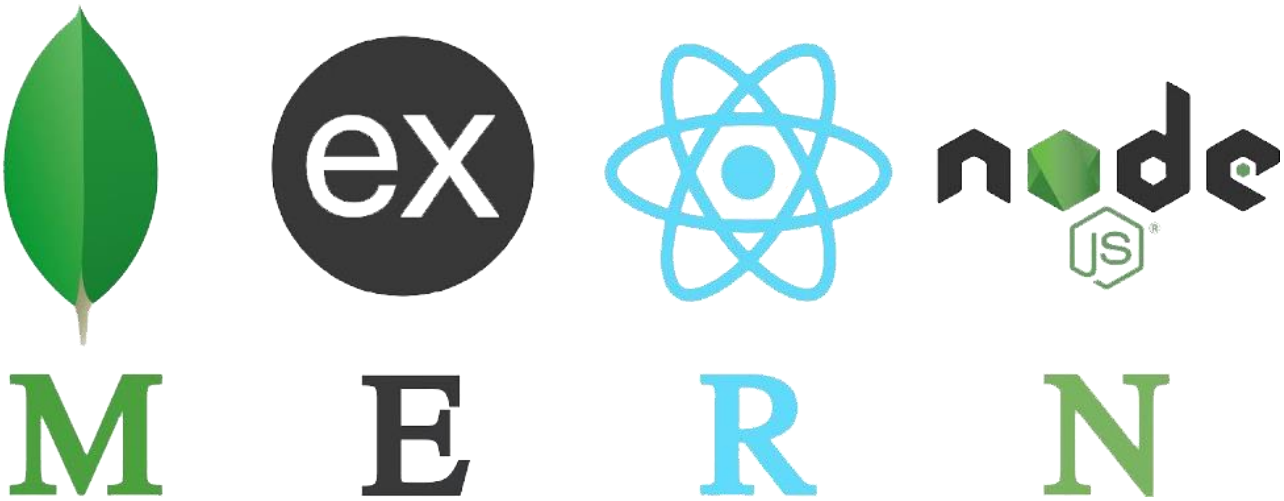
For Development:

- Visual Studio Code
- Chrome
- ColorZilla
- PostMan
- Git Lense
- Es6 Formatter

For Deployment

- Github
- Heroku
- Aws
- Gitbash

Which Technology is Used and Why?



Technology used:

Reactjs - React. js is an open-source JavaScript library that is used for **building user interfaces specifically for single-page applications**. It's used for handling the view layer for web and mobile apps. React also allows us to create reusable UI components.

Nodejs - is an open-source, cross-platform, back-end JavaScript runtime environment that runs on the V8 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—running scripts server-side to produce 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, rather than different languages for server-side and client-side scripts.

Expressjs - Express.js, or simply Express, is a back end web application framework for 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.

MongoDB - 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.

HTML - The HyperText Markup Language, or HTML is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets and scripting languages such as JavaScript

CSS - Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript

SASS - Sass is a preprocessor scripting language that is interpreted or compiled into Cascading Style Sheets. SassScript is the scripting language itself.

Redux - Redux is an open-source JavaScript library for managing and centralizing application state. It is most commonly used with libraries such as React or Angular for building user interfaces. Similar to Facebook's Flux architecture, it was created by Dan Abramov and Andrew Clark.

SocketIO - In most cases, the connection will be established with WebSocket, providing a low-overhead communication channel between the server and the client.

Why choose of MERN Stack!

a. UI rendering and performance - React JS is the best when it is about UI layer abstraction. Since React is only a library, it provides you the freedom to build the application and organize the code however you want. So, it is better than Angular in terms of UI rendering and performance.

b. Cost-Effective - As MERN Stack uses one language throughout that is Javascript so it will be beneficial for a company to hire Javascript experts only rather than hiring different specialists for different technology. This move will save a lot of time and money.

c. Open Source - All technologies that are involved in MERN are open-source. This feature allows a developer to get solutions to queries that may evolve during development, from the available open portals. As a result, it will be beneficial for a developer.

d. To switch between client and server - As everything is written in one language this is why MERN is simple and fast. And also it is easy to switch between client and server.

Service Used :

Mapbox - a plugin for react-map-gl, is a JavaScript library that allows you to add an interactive map to your website. Mapbox.js is no longer in active development. Mapbox.js natively supports raster tilesets, and can support vector tilesets using L.mapbox.styleLayer

.

EmojiPicker – for emoji sending.

Weatherbit- A weather Api.

Cloudinary - a cloud-based image and video management services. It enables users to upload, store, manage, manipulate, and deliver images and video for websites and apps. Wikipedia

LIBRARY USED

At FrontEnd:

- "react-awesome-slider": "^4.1.0",
- "react-cropper": "^1.3.0",
- "react-dom": "^16.12.0",
- "react-helmet": "^6.1.0",
- "react-hook-form": "^4.9.8",
- "react-html-parser": "^2.0.2",
- "react-image-lightbox": "^5.1.1",
- "react-infinite-scroll-component": "^5.0.4",
- "react-infinite-scroller": "^1.2.4",
- "react-loader-spinner": "^4.0.0",
- "react-map-gl": "^5.3.10",
- "react-redux": "^7.2.0",

- "react-router-dom": "^5.1.2",
- "react-scripts": "^4.0.3",
- "react-select": "^3.1.0",
- "react-slick": "^0.28.1",
- "react-toastify": "^8.1.0",
- "react-transition-group": "^4.3.0",
- "react-video-thumbnail": "^0.1.3",
- "reactjs-file-uploader": "^1.0.8",
- "redux": "^4.0.5",
- "simple-react-video-thumbnail": "^0.0.7",
- "socket.io-client": "^2.3.0",
- "styled-components": "^5.3.3",
- "sun-time": "^1.0.2",
- "validator": "^12.2.0",

At BackEnd

- "bcryptjs": "^2.4.3",
- "body-parser": "^1.19.0",
- "cloudinary": "^1.21.0",
- "cors": "^2.8.5",
- "dotenv": "^8.2.0",
- "express": "^4.17.1",
- "express-validator": "^6.4.0",
- "jsonwebtoken": "^8.5.1",
- "mongoose": "^5.7.8",
- "mongoose-unique-validator": "^2.0.3",
- "multer": "^1.4.2",
- "multer-storage-cloudinary": "^2.2.1",
- "nodemon": "^2.

For Deployment Services

Features Of Social Media and Main Components:

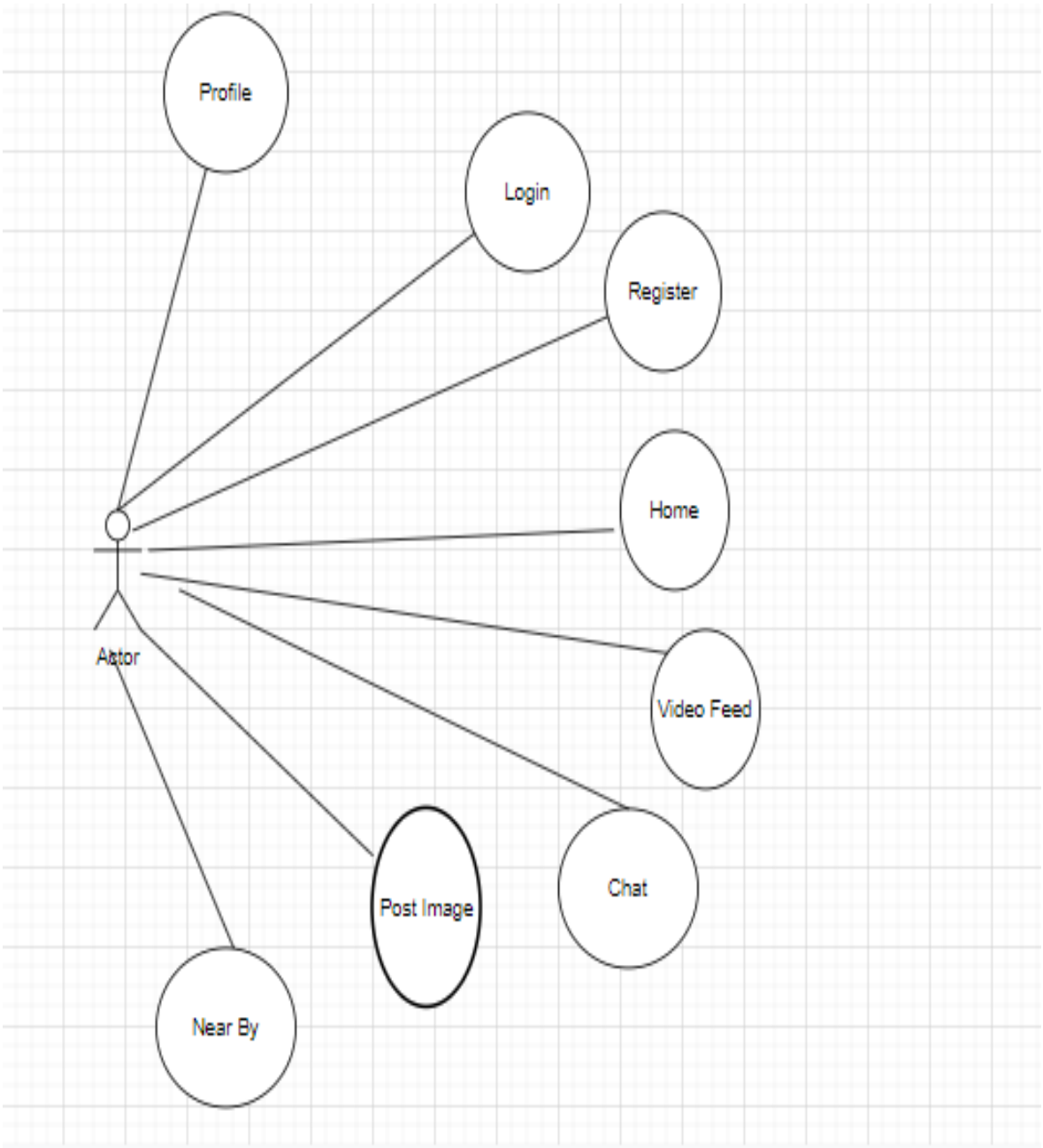
- Register
- Friends
- Chat
- Near By
- Weather
- Notification
- Account
- Setting

User features:

- register and login users.
- post images can be uploaded using camera or file system.
- pagination on every pages.
- Dark mode.
- Weather option
- Near by friends
- search other users by username.
- user suggestions menu.
- save any post to collection.
- Emoji picker.
- delete posts and comments.
- admin panel is included.
- Explore page to view other posts by random users.
- notifications page.

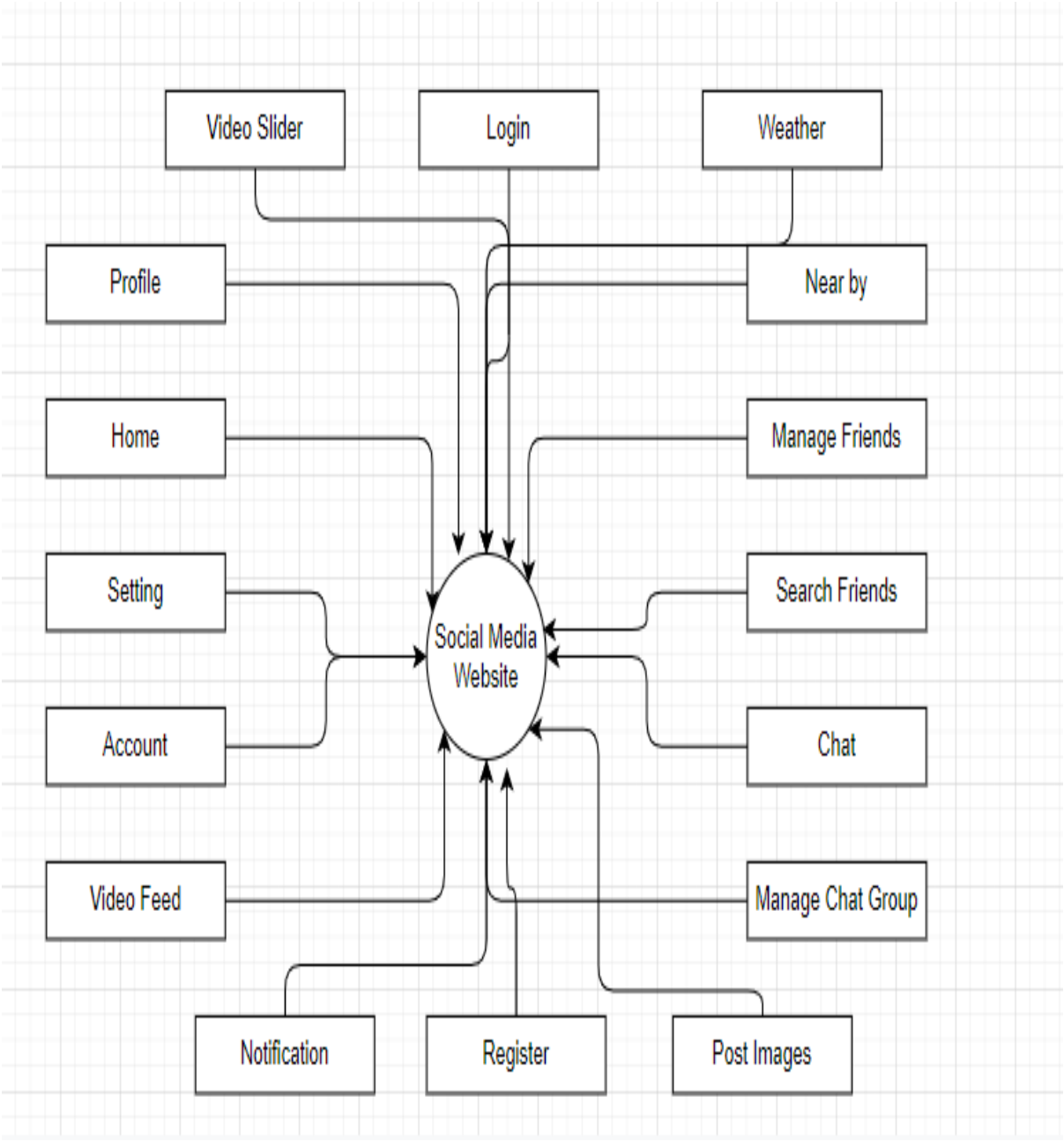
- profile page.
- edit profile page user data.
- password is stored in database in encrypted format with salt.
- create and edit posts.
- like ,comment, share and edit posts.
- posts include text(caption) and image(s).
- comment on posts.
- reply comments.
- like commets.
- clear notification option.
- profile page shows user details and posts with following and followers menu.
- Group chat
- Single chat
- Online friends
- Video feed
- Share location
- Edit image
- Upload profile
- Lazy loading
- Send image to other user while chat
- Chat with socket io

Use Case Diagram



Overview of Project Functionality

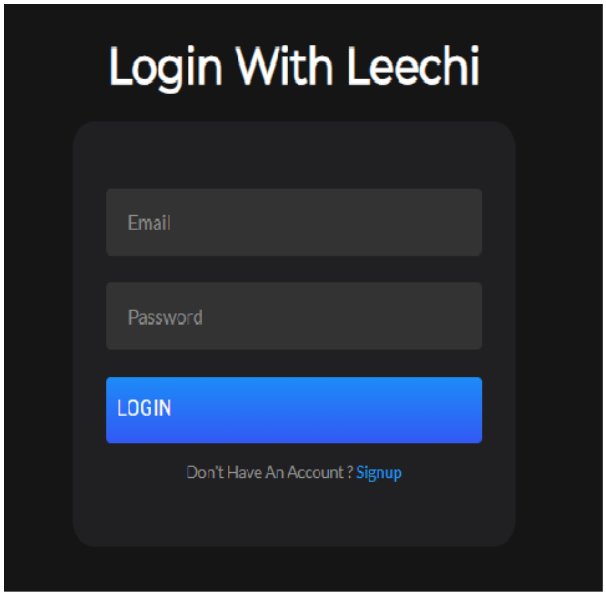
Data Flow Diagram

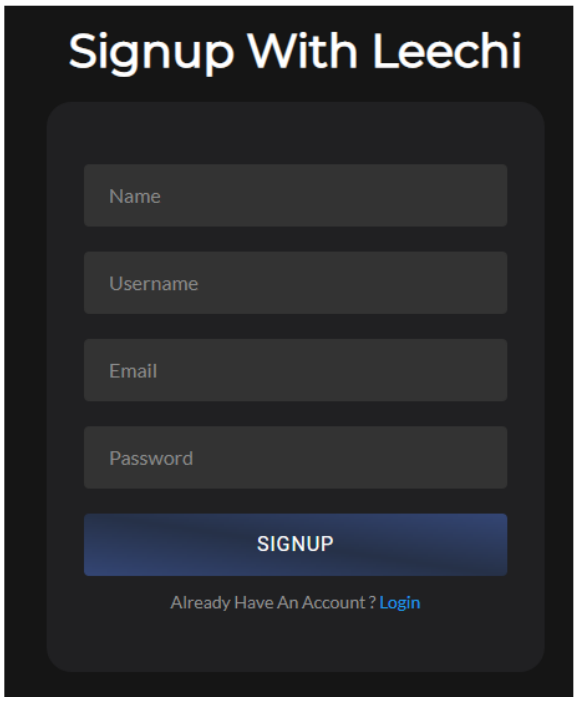


Application Flow With Diagram

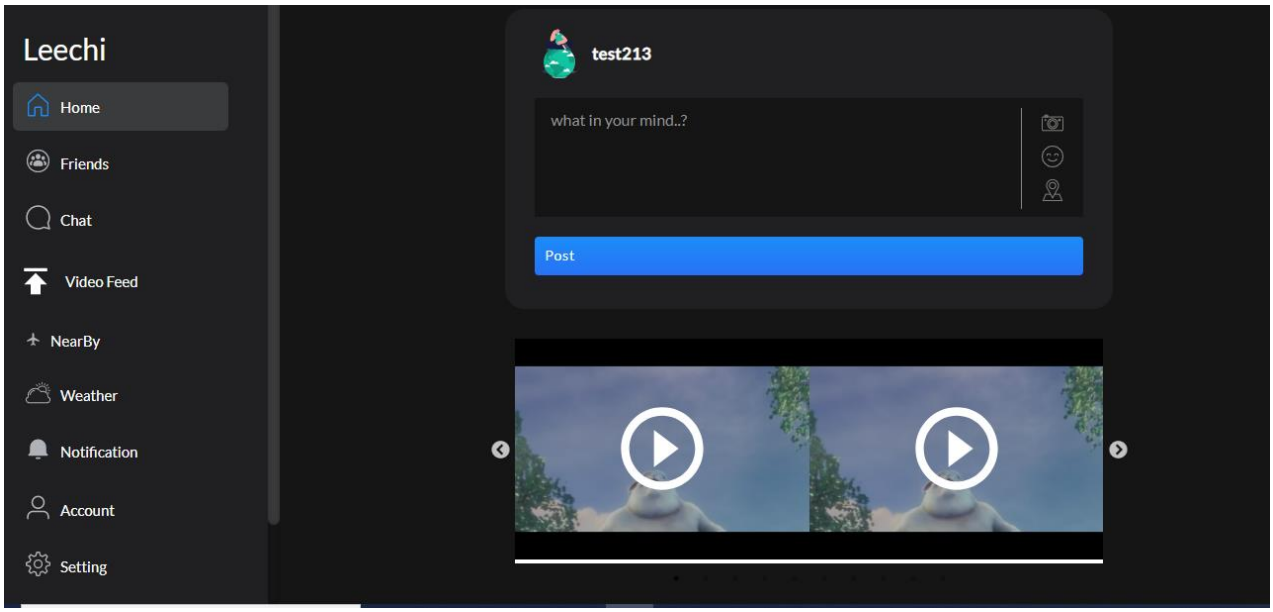
SNAPSHOTS

Login Sceen

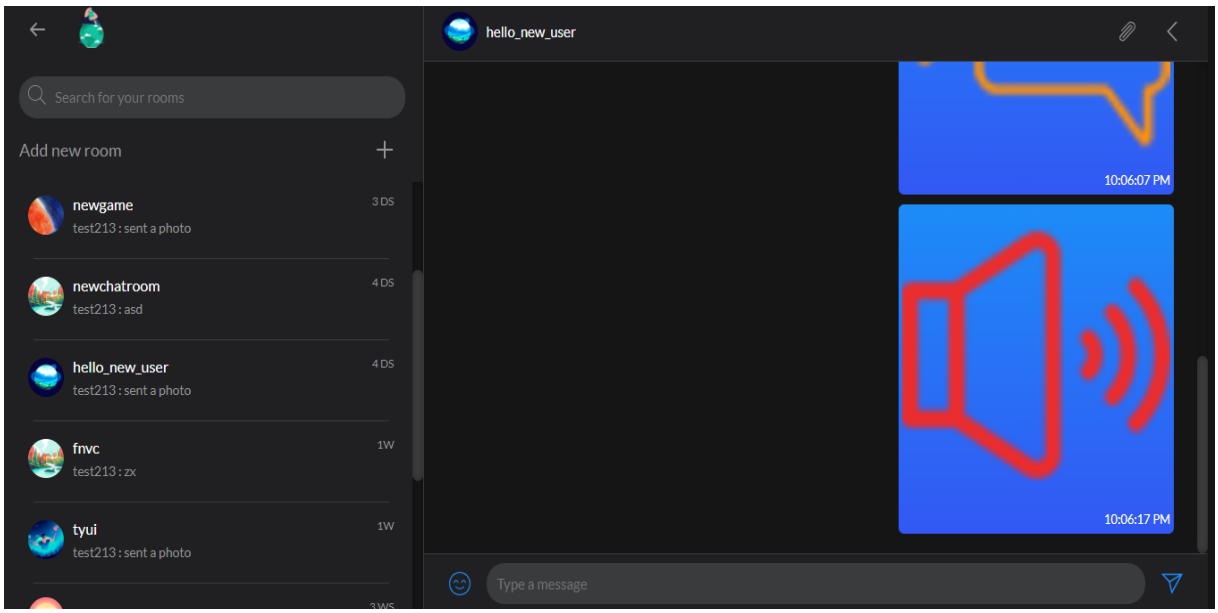




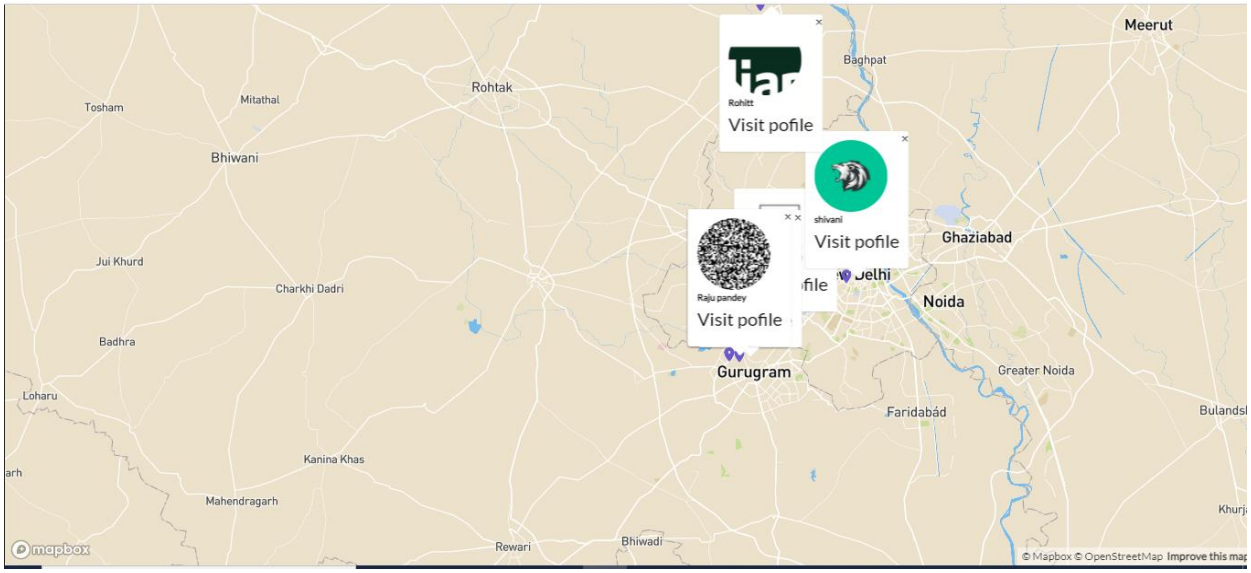
Home Screen



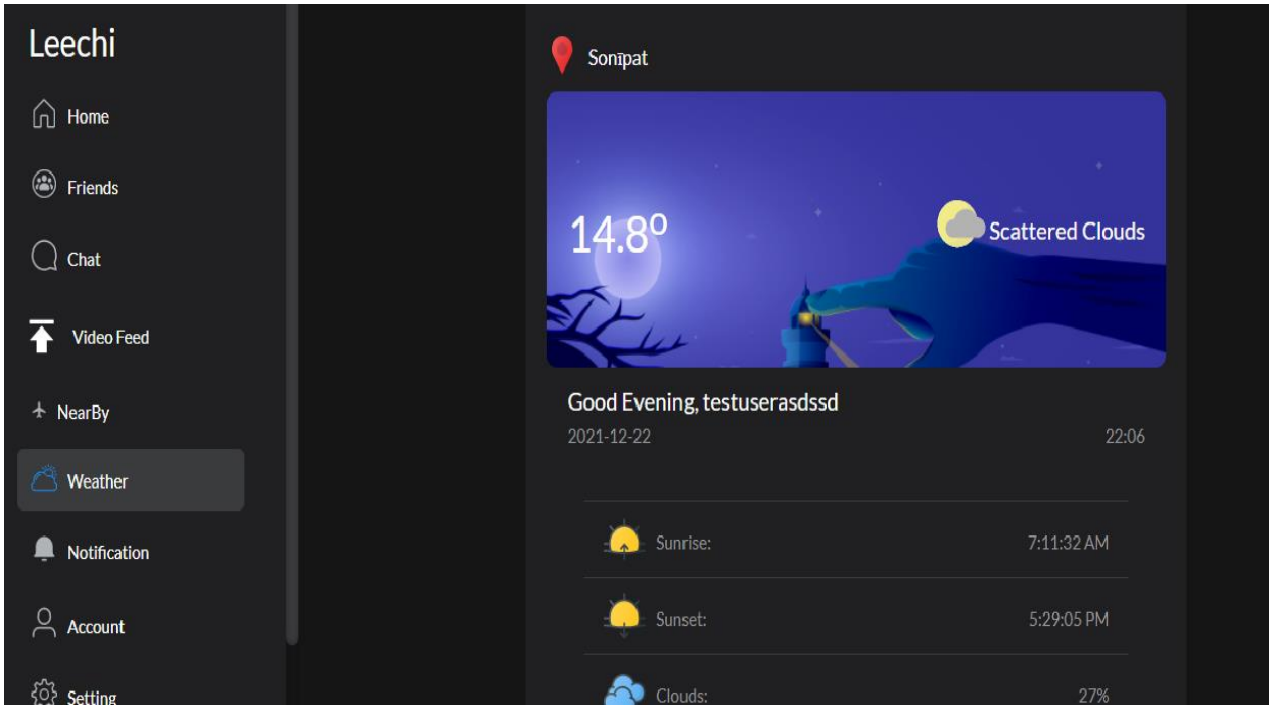
Chat Screen



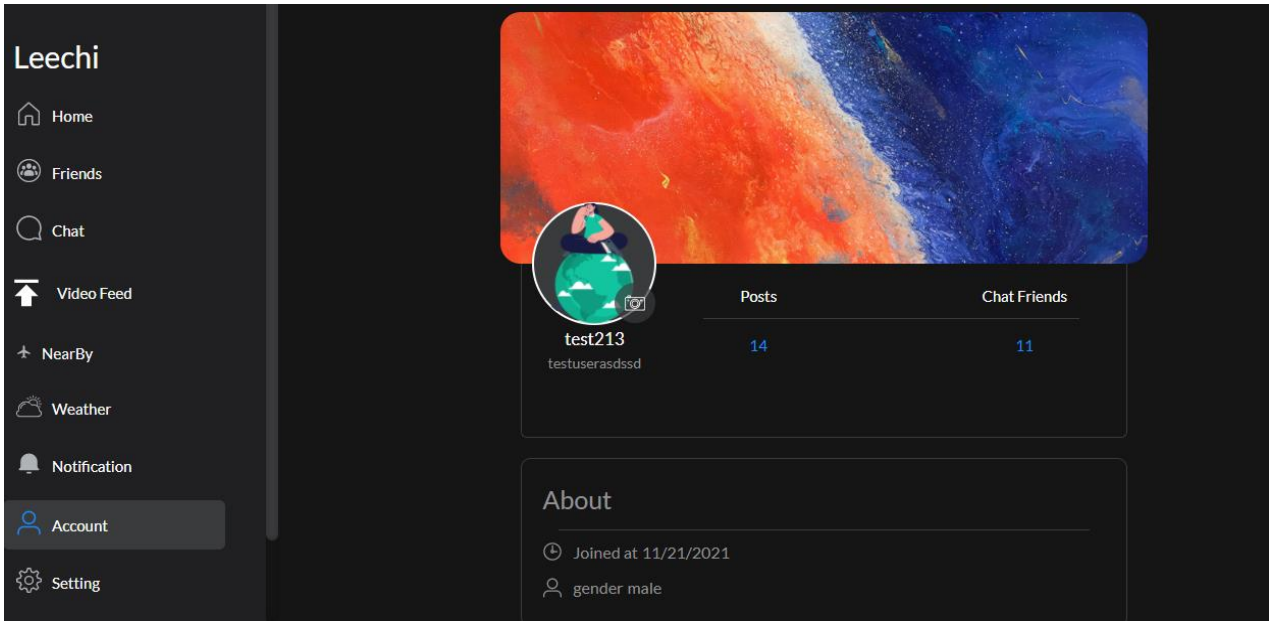
Map



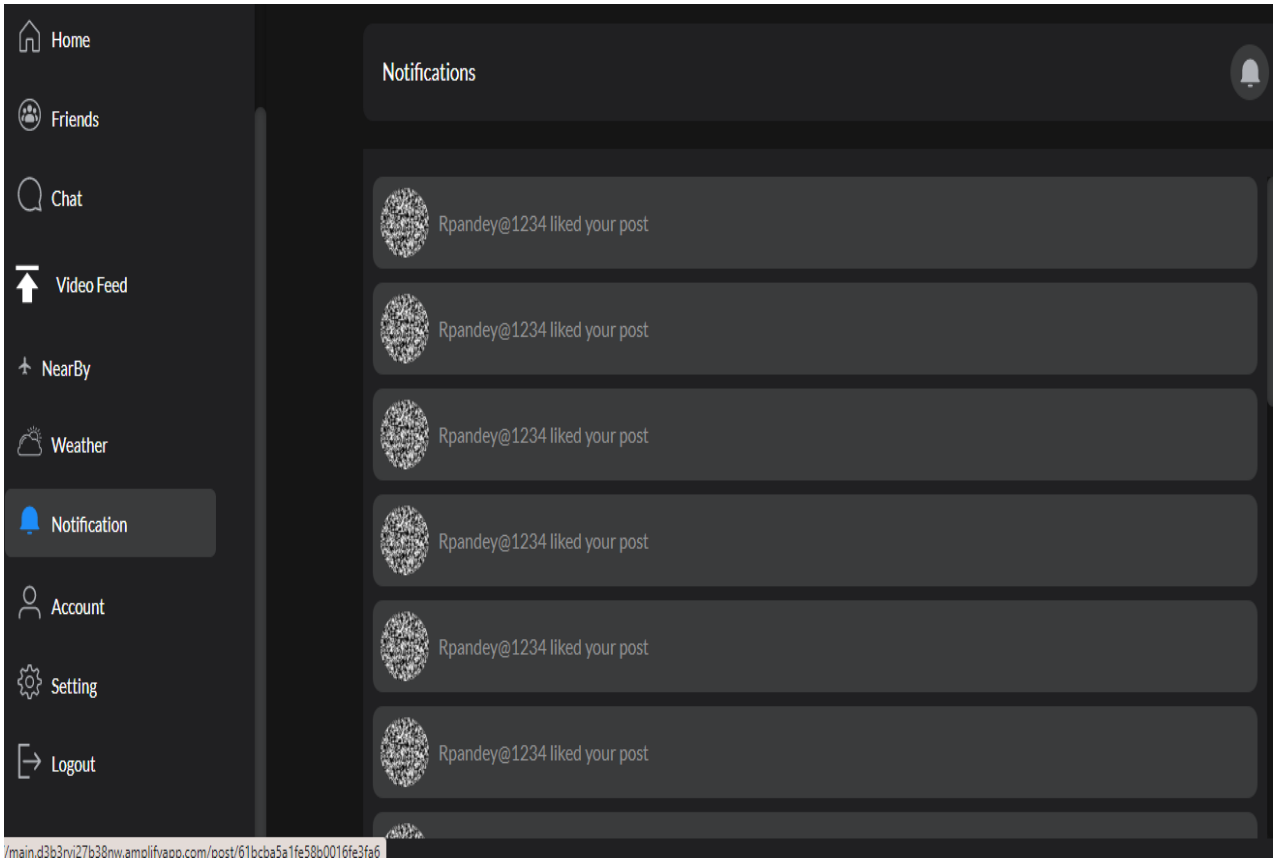
Weather



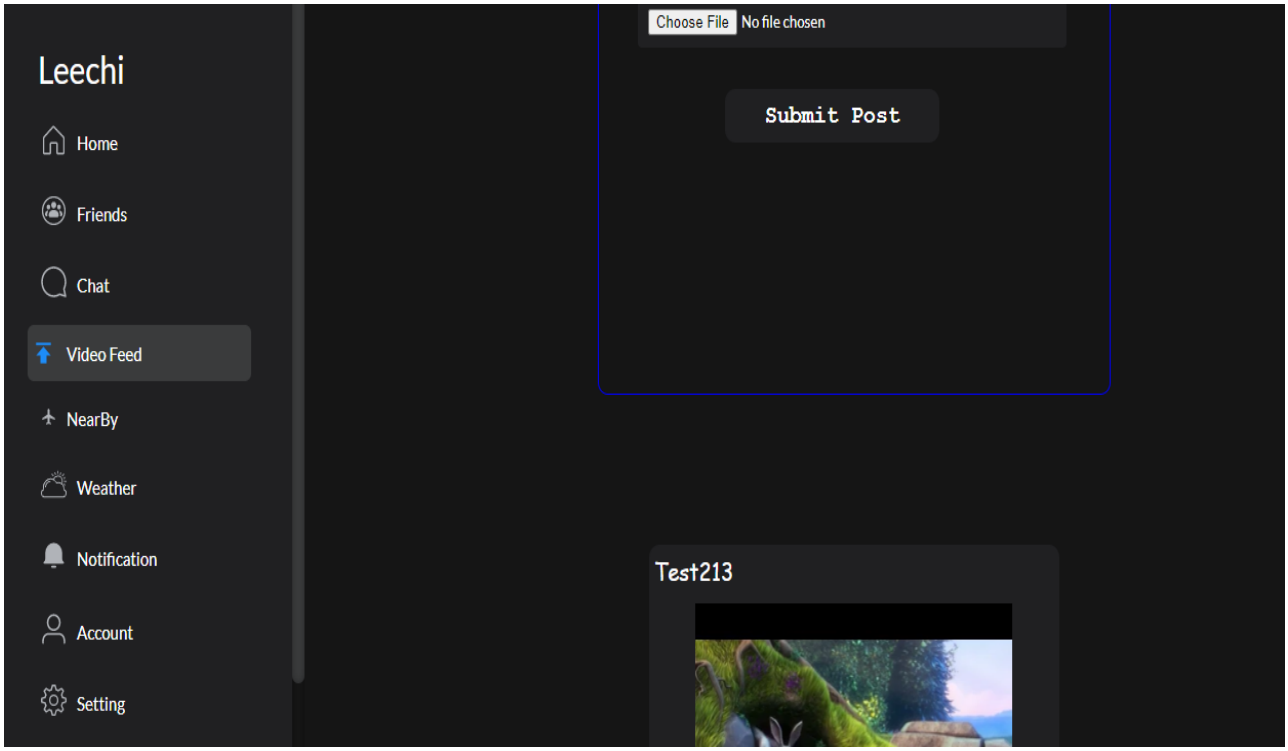
Profile Page



Notification Page



Video Feed Page



References

1. http://stackoverflow.com/
2. http://youtube.com/
3. Socket.io Documentation
4. React js Documentation
5. Mongo Documentation
6. Node js Documentation