Friend's Spy

A Project Report Submitted in partial fulfilment of the requirements for the award of the degree of

Bachelor of Technology

in

Computer Science and Engineering

By

Aman Kushwaha**(181500075)**

Aman Saxena**(181500077)**

Akarshit Srivastava**(181500056)**

Akshat Sinha**(181500063)**

Harshit Saxena**(181500258)**

Under the Guidance of

Mr. Mandeep Singh

Department of Computer Engineering and Applications

Institute of Engineering and Technology

GLA University

Mathura- 281406, India

April, 2021

Department of computer Engineering and Applications
GLA University, Mathura
17 km. Stone NH#2, Mathura-Delhi Road, P.O. – Chaumuha,
Mathura – 281406

Declaration

We hereby declare that the work which is being presented in the CCV Project "**Friend's Spy**", in fulfillment of the requirements for CCV project in Computer Science and Engineering and submitted to the Department of Computer Engineering and Applications of GLA University, Mathura, is an authentic record of our own work carried under the supervision of **Mr. Mandeep Singh(Assistant Professor).**

The contents of this project report, in full or in parts, have not been submitted to any other Institute or University for the award of any degree.

Sign 
Name of Candidate: Aman Kushwaha
University Roll No.: 181500075

Sign _____
Name of Candidate:Aman Saxena
University Roll No.:181500077

Sign _____
Name of Candidate:Akarshit Srivastava
University Roll No.: 181500056

Sign _____
Name of Candidate:Akshat Sinha
University Roll No.: 181500063

Sign _____
Name of Candidate:Harshit Saxena
University Roll No.: 181500258

Certificate

This is to certify that the above statements made by the candidate are correct to the best of my/our knowledge and belief.

_____

Supervisor
Mr. Mandeep Singh
Assistant Professor

Date :15 Apr,2021

ACKNOWLEDGEMENT

It gives us a great sense of pleasure to present the report of the B. Tech Mini Project undertaken during B. Tech. Third Year. This project in itself is an acknowledgement to the inspiration, drive and technical assistance contributed to it by many individuals. This project would never have seen the light of the day without the help and guidance that we have received.

Our heartiest thanks to Dr. (Prof). Anand Singh Jalal, Head of Dept., 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.

We owe special debt of gratitude to Mr. Mandeep Singh, Assistant Professor, for his constant support and guidance throughout the course of our work. His sincerity, thoroughness and perseverance have been a constant source of inspiration for us. He has showered us with all his extensively experienced ideas and insightful comments at virtually all stages of the project & has also taught us about the latest industry-oriented technologies.

We also do not like to miss the opportunity to acknowledge the contribution of all instructors who are available on YouTube and Stackoverflow. I would like to thank all my friends who helped me in making this project.

Last but not the least, I would like to express our deep sense of gratitude and earnest thanks giving to our dear parents for their moral support and heartfelt cooperation during the project.

ABSTRACT

Friend's Spy is a highly scalable real world Dynamic Web Application which is using MERN Stack for a new startup that scale infinitely. where multiple users can login and post/manage contents that they submit. Main target of Academic Earth is to Target millions of potentials users to use and engage with our app.Users could be anyone.

This document aims at defining the overall requirements for "SOCIAL NETWORKING". Efforts have been made to define the requirements exhaustively and accurately. The final product will be having only features or functionalities mentioned in this document and assumptions for any additional functionality should not be made by any of the parties involved in developing / testing / implementation using the product.

In case any additional features are mandatory, formal changes / requests would be produced.

For Extremely simple architecture that is easy to maintain and scale we have created Separate backend API and frontend built with react/nextjs for easy development, code maintenance and deployment

Our aim is to enable just 1 or 2 developer to maintain the entire project and continuously grow in future.

CONTENTS

1. Introduction
1.1 Overview and Motivation

The website “Friend’s SPy” will be used to connect the people in very easy ,simple and efficient way and one can share their feeling ,information ,ideas…..and many more…, the services offered to an individual’s choice(s) and availability for making friends among various areas and destinations. A log concerning the registration and requests for friends and various other features by users are also maintained. The website will also provide benefits to verified user(s).

The website, according to the following proposed solution, will ease the connecting people s thereby converging the world into a small system.

Overview of the project:

Friend’s Spy, as the name suggests is an social networking allowing the users to interact with each other and exchange their views. This project also enables the users to see the details of their friends upload their own photographs, add their friends, leave a scrap & send testimonials.

1.2.OBJECTIVE

1. The project’s objective is to enable users to communicate with other people.
2. It allows the user to search for friends.
3. This website provides user the ability to upload the photographs.
4. It also enables the user to leave the scraps & send the testimonials.
5. Efficient usage of resources.
6. It provides security through verification process.
7. Performance is high.
8. Reduces the effort and time in gathering the information about the users.
9. Provides a complete record of all the available vehicles for pooling.
10. The constraints and checks lead to a valid database.

1.3 Summary Of Application

- Extremely simple architecture that is easy to maintain and scale
- Separate backend API and frontend built with react/nextjs for easy development, code maintenance and deployment
- Keep the frontend as simple as possible
  - **Scenario 1** > create a page in react to do certain task. For example it fetch the data from your API/Server and put that data on the web page for user to see. Job done
  - **Scenario 2 >** create admin page to create a new post. For example to post the data to your API/Server and show the returned response of either success or error. Job done.
- Keep the backend API as simple as possible
  - **Scenario 1 >** You have a GET route. For example '/api/posts'. You will receive a request from react/frontend for getting all posts from database.
  - You pass this request to a controller method. Controller method/function will make a query to database, get all posts and return back to react/frontend. Job done.
  - **Scenario 2 >** You have a POST route. For example '/api/create/post'. You will receive data from react/frontend to create a new post. You pass that data to a controller method.
  - Controller method/function will save that data in database and return the success response, which will be sent back to react. Job done.
- This is the logic of our app or basically any web app out there. The cycle of requests/responses. Request from frontend, Response from server.

1.4 Organization Of The Project

The software has different modules which help it to achieve its objectives, those are:

A. Client Side: Friend's Spy uses React and NextJS (React Framework) in the client Side.

B. Server Side: For API/server Friend's SPy uses Node Express MongoDB.

C. Mongoo DB : We have used Mongo Atlas as Managed Database Service in the cloud.

D. Admin Dashboard: To Restrict some users to access the private content admin authorization is done on admin dashboard. In Admin Dashboard Admin can Access each and everything of this Web Application.

Work can be done in Admin dashboard:-

- Create Post
- Update, Delete Post of Admins
- Update ,Delete Post of Clients
- Delete Users.
- Update Profile of any User/Admin.
- Like/Comment.

F .User Dashboard: User Have Limited Access in Application for example user can not delete or create Category

Work can be done in User Dashboard:

- Create Post
- Update Delete Post
- Like /Comment
- Update Profile

G. Work Organization:

Aman Kushwaha:

Documentation	**Node API**	**React-Front**
Synopsis	Controller Auth	Admin Sign Up
DFD	Controller Post	Admin Log In
Usecase	Controller User	Admin Delete Post
Report	API Docs	Admin Update Profile
Presentation	User Schema	Like and Comment
Testing	Mongo DB Auth	Create Post

Akarshit:

Documentation	**Node API**	**React-Front**
Testing	Helper Index	User Sign In
Unauthorized Post	Model Post	User Log In
Check Follow	Model User	User Update Post
Tidy Up	API Docs	User Update Profile
	Mongo Auth	Image delete

Akshat:

Documentation	**Node API**	**React-Front**
Components	Validator Index	Reset Client ID
Implement Unflow	Client Post	Image Upload
reCaptcha	Client Model	Text Upload
	API Docs	Text Delete
	Mongo Auth	Unfollow User

Aman Saxena:

Documentation	**Node API**	**React-Front**
Tiddy Up	Routes Auth	Sing In Redirect
Render Unflow	Route Post	User Unlike
reCaptcha	Route User	User Comment Delete
	API Docs	User Delete Post
	Mongo Auth	Unfollow User

Harshit Saxena:

Documentation	**Node API**	**React-Front**
Tiddy Up	Sign In Auth	Client Find People
Render Unflow	Async await	Client Edit Profile
Schema	Virtual Fields	Follow Profile
	API Docs	Profile Tabs
	Mongo Auth	Fetch Users

2.Software and Requirement Analysis
2.1 Software Requirement

VISUAL STUDIO:

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight. It can produce both native code and managed code.

Visual Studio includes a code editor supporting IntelliSense (the code completion component) as well as code refactoring. The integrated debugger works both as a source-level debugger and a machine-level debugger. Other built-in tools include a code profiler, forms designer for building GUI applications, web designer, class designer, and database schema designer. It accepts plug- ins that enhance the functionality at almost every level—including adding support for source control systems (like Subversion and Git) and adding new tool sets like editors and visual designers for domain-specific languages or toolsets for other aspects of the software development lifecycle (like the Team Foundation Server client: Team Explorer).

Visual Studio supports 36 different programming languages and allows the code editor and debugger to support (to varying degrees) nearly any programming language, provided a language-specific service exists. Built-in languages include C, C++, C++/CLI, Visual Basic .NET, C#, F#,JavaScript, TypeScript, XML, XSLT, HTML, and CSS. Support for other languages such as Python,Ruby, Node.js, and M among others is available via plug-ins. Java (and J#) were supported in the past.

WEB BROWSER:

A **web browser** (commonly referred to as a **browser**) is a software application for accessing information on the World Wide Web. Each individual web page, image, and video is identified by a distinct Uniform Resource Locator(URL), enabling browsers to retrieve these resources from a web server and display them on the user's device.

A web browser is not the same thing as a search engine, though the two are often confused. For a user, a search engine is just a website, such as google.com, that stores searchable data about other websites. But to connect to a website's server and display its web pages, a user needs to have a web browser installed on their device.

The most popular browsers are Chrome, Firefox, Safari, Internet Explorer, and Edge.

2.2 Language and Framework Requirements

HTML

Hypertext Markup Language (HTML) is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript, it forms a triad of cornerstone technologies for the World Wide Web.

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 the appearance of the document.

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> surround and provide information about document text and may include other tags as sub-elements. 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. Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), maintainer of both the HTML and the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997. HTML code ensures the proper formatting of text and images so that your Internet browser may display them as they are intended to look. Without HTML, a browser would not know how to display text as elements or load images or other elements. HTML also provides a basic structure of the page, upon which Cascading Style Sheets are overlaid to change its appearance. One could think of HTML as the bones (structure) of a web page, and CSS as its skin (appearance).

CSS (Cascading Style Sheets)

Cascading Style Sheets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable.

CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs,variations in display for different devices and screen sizes as well as a variety of other effects.

CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML.

BOOTSTRAP

Bootstrap is a free and open front-end framework for designing websites and web applications. It contains HTML - and CSS -based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions. Unlike many earlier web frameworks, it concerns itself with front end development only.

Bootstrap is the second most-starred project on GitHub, with more than 129,000 stars. Bootstrap comes with several JavaScript components in the form of jQuery plugins. They provide additional user interface elements such as dialog boxes, tooltips, and carousels. They also extend the functionality of some existing interface elements, including for example an auto-complete function for input fields. In version 1.3, the following JavaScript plugins are supported: Modal, Dropdown, Scrollspy, Tab, Tooltip, Popover, Alert, Button, Collapse, Carousel and Typeahead.

JAVA SCRIPT (JS)

JavaScript, often abbreviated as **JS**, is a high-level, interpreted programming language that conforms to the ECMAScript specification. It is a programming language that is characterized as dynamic, weakly typed, prototype-based and multi-paradigm.

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, and major web browsers have a dedicated JavaScript engine to execute it.

As a multi-paradigm language, JavaScript supports event-driven, functional, and imperative (including object-oriented and prototype-based) programming styles. It has APIs for working with text, arrays, dates, regular expressions, and the DOM, but the language itself does not include any I/O, such as networking, storage, or graphics facilities. It relies upon the host environment in which it is embedded to provide these features.

Initially only implemented client-side in web browsers, JavaScript engines are now embedded in many other types of host software, including server-side in web servers and databases, and in non-web programs such as word processors and PDF software, and in runtime environments that make JavaScript available for writing mobile and desktop applications, including desktop widgets.

The terms *Vanilla JavaScript* and *Vanilla JS* refer to JavaScript not extended by any frameworks or additional libraries. Scripts written in Vanilla JS are plain JavaScript code.

Although there are similarities between JavaScript and Java, including language name, syntax, and respective standard libraries, the two languages are distinct and differ greatly in design. JavaScript was influenced by programming languages such as Self and Scheme.

Node JS & Express JS:

Node.js is an open source and cross-platform runtime environment for executing JavaScript code outside of a browser. You need to remember that NodeJS is not a framework and it's not a programming language. Most of the people are confused and understand it's a framework or a programming language. We often use Node.js for building back-end services like APIs like Web App or Mobile App. It's used in production by large companies such as Paypal, Uber, Netflix, Walmart and so on.

Express is a small framework that sits on top of Node.js's web server functionality to simplify its APIs and add helpful new features.It makes it easier to organize your application's functionality with middle ware and routing; it adds helpful utilities to Node.js's HTTP objects;it facilitates the rendering of dynamic HTTP objects.

Express is a part of MEAN stack, a full stack JavaScript solution used in building fast, robust, and maintainable production web applications.

MongoDB:

MongoDB, the most popular NoSQL database, is an open-source document-oriented database. The term 'NoSQL' means 'non-relational'. It means that MongoDB isn't based on the table-like relational database structure but provides an altogether different mechanism for storage and retrieval of data. This format of storage is called JSON format.

SQL databases store data in tabular format. This data is stored in a predefined data model which is not very much flexible for today's real-world highly growing applications. Modern applications are more networked, social and interactive than ever. Applications are storing more and more data and are accessing it at higher rates.

Relational Database Management System(RDBMS) is not the correct choice when it comes to handling big data by the virtue of their design since they are not horizontally scalable. If the database runs on a single server, then it will reach a scaling limit. NoSQL databases are more scalable and provide superior performance. MongoDB is such a NoSQL database that scales by adding more and more servers and increases productivity with its flexible document model

REACT JS

React (also known as **React.js** or **ReactJS**) is an open-source, front end, JavaScript library for building user interfaces or UI components. Advantages of React js.

2.3 Software And Hardware requirement:

Following are the hardware and the software requirements for our project:

1. Hardware:

§ Laptop/Desktop

§ 1.8 GHz or faster processor. Quad-core or better recommended

§ 4 GB of RAM and core i3 processor

§ Hard disk space: Minimum of 500MB

2. Software:

· Windows 8.1 and above

· Visual Studio Code

· Web Browser

· Bootstrap

· Github Desktop

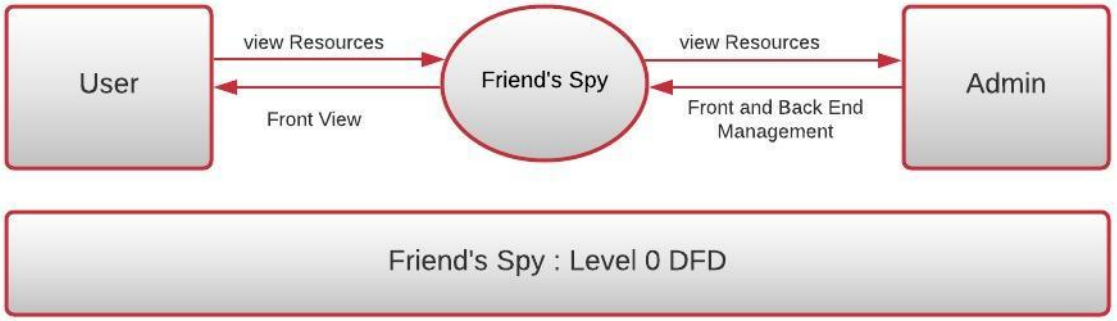
3. Language and Framework Requirements:

- React JS
- CSS
- Bootstrap
- Next JS
- Express JS
- MongoDB

3. Software Design

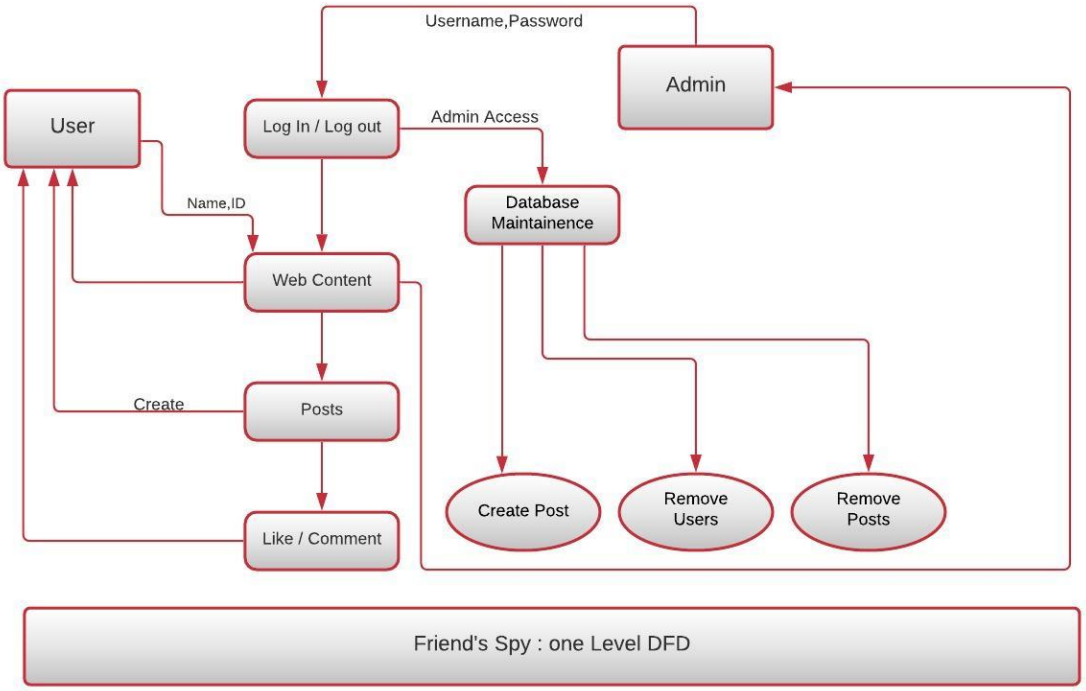
3.1 DFDs

Level Zero:



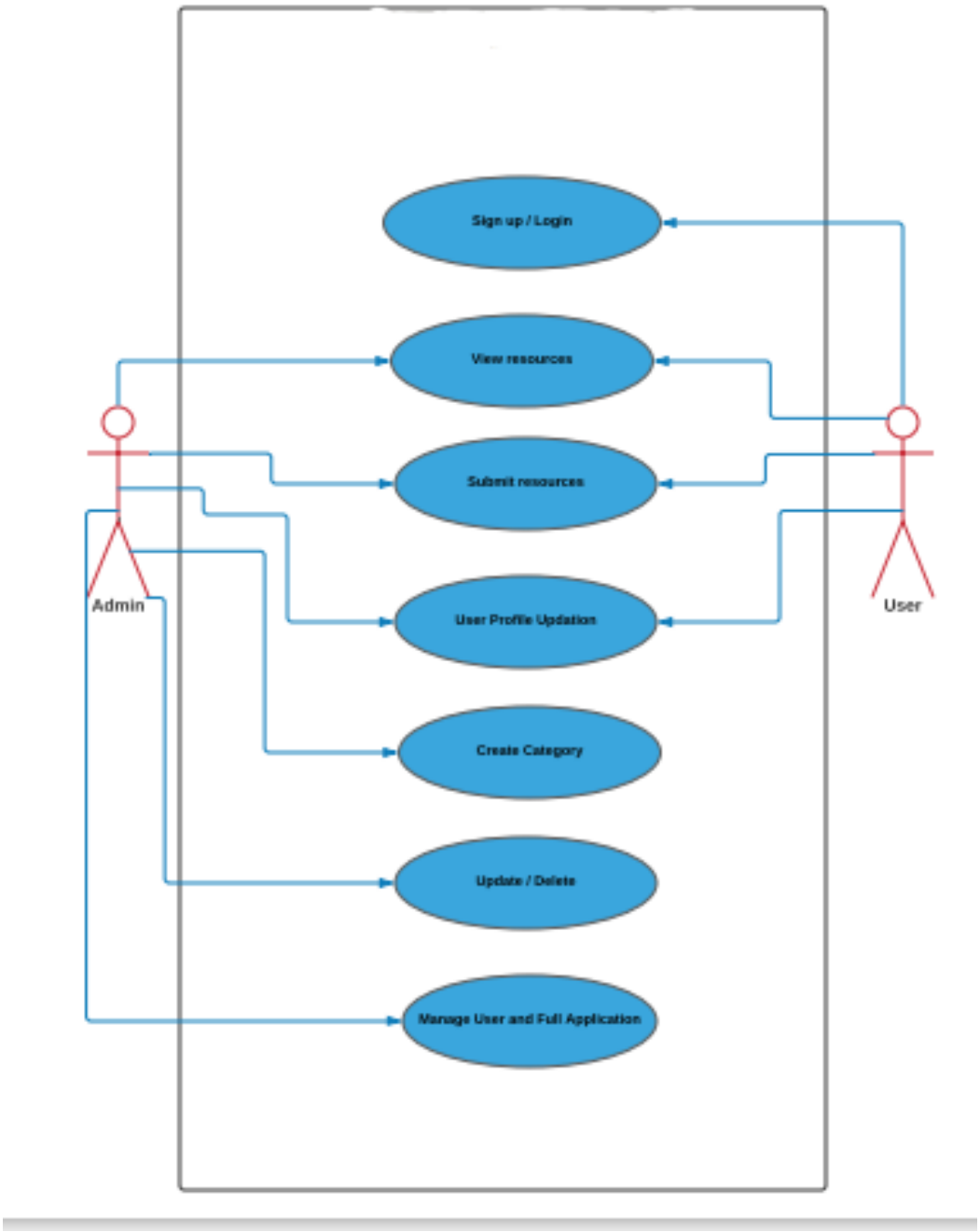
3.1 DFD Level 0

Level One:



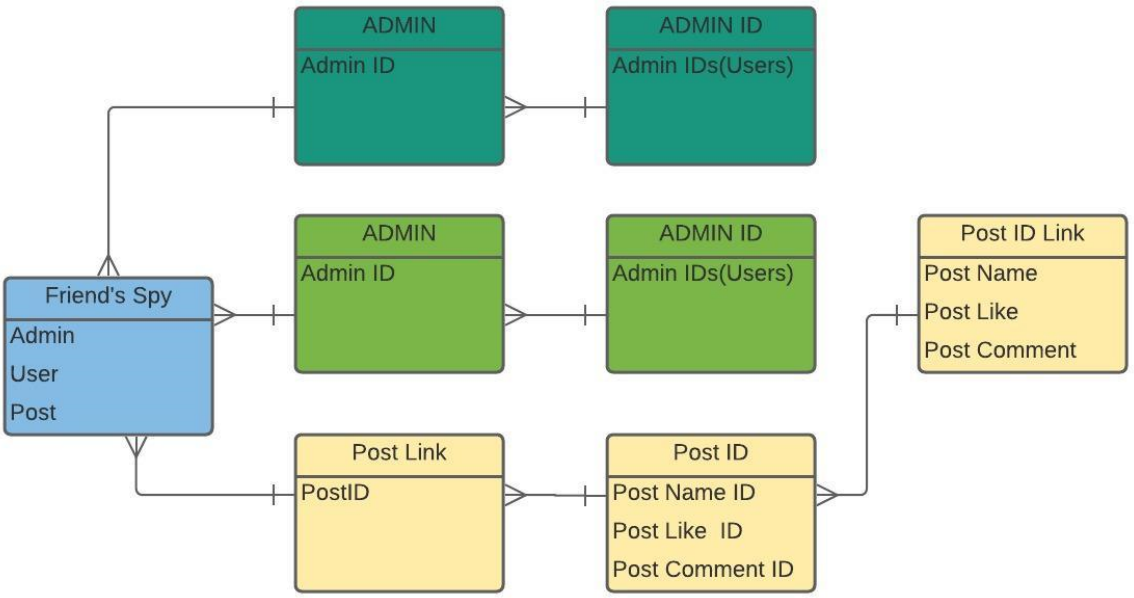
3.2 DFD Level 1

3.2 Usecase Diagram



3.3 Use Case Diagram

3.3 Json Schema

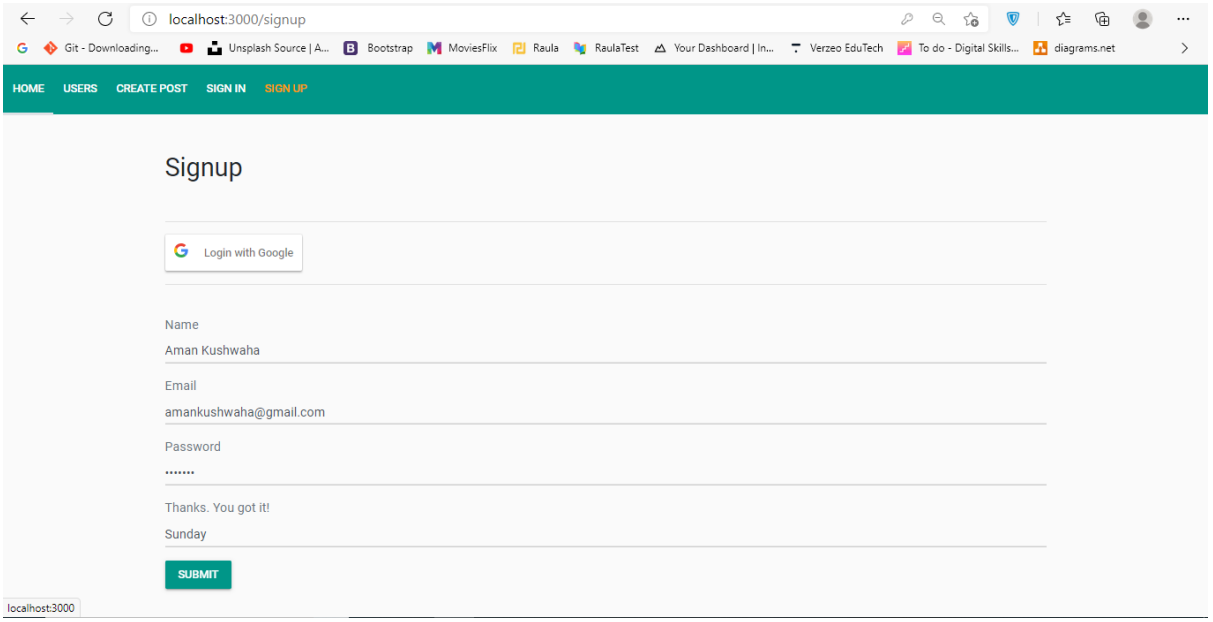


3.4 Json Schema

4.Implementation and User Interface

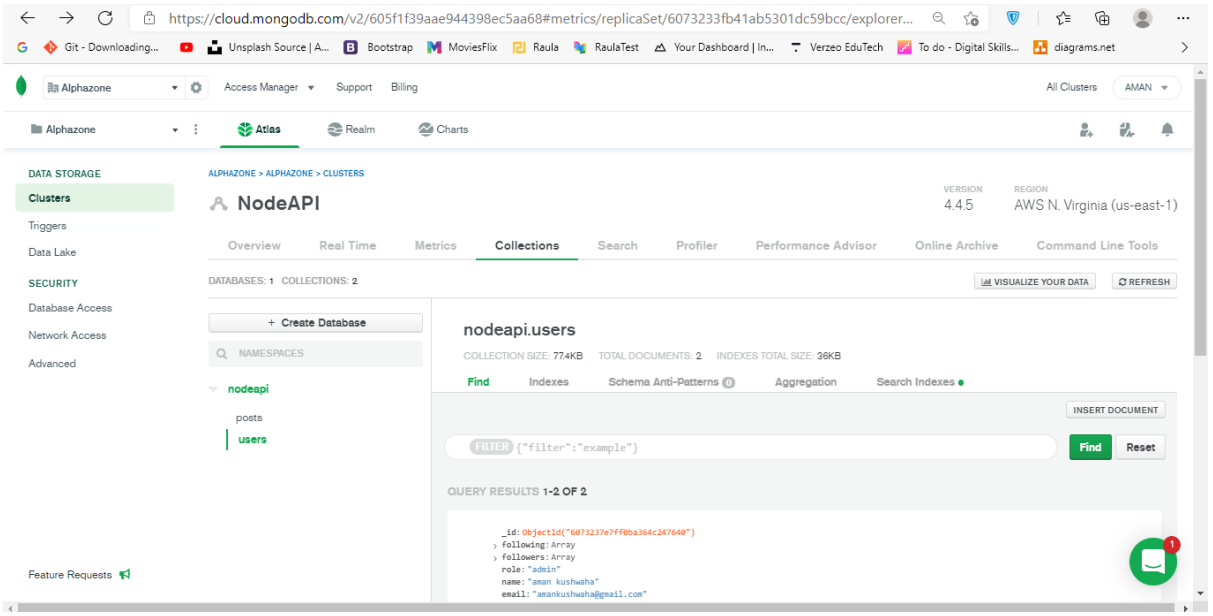
A). Registration: Users will signup/signin to our app to post/share/like the links/urls

Step 1: Data Entry:



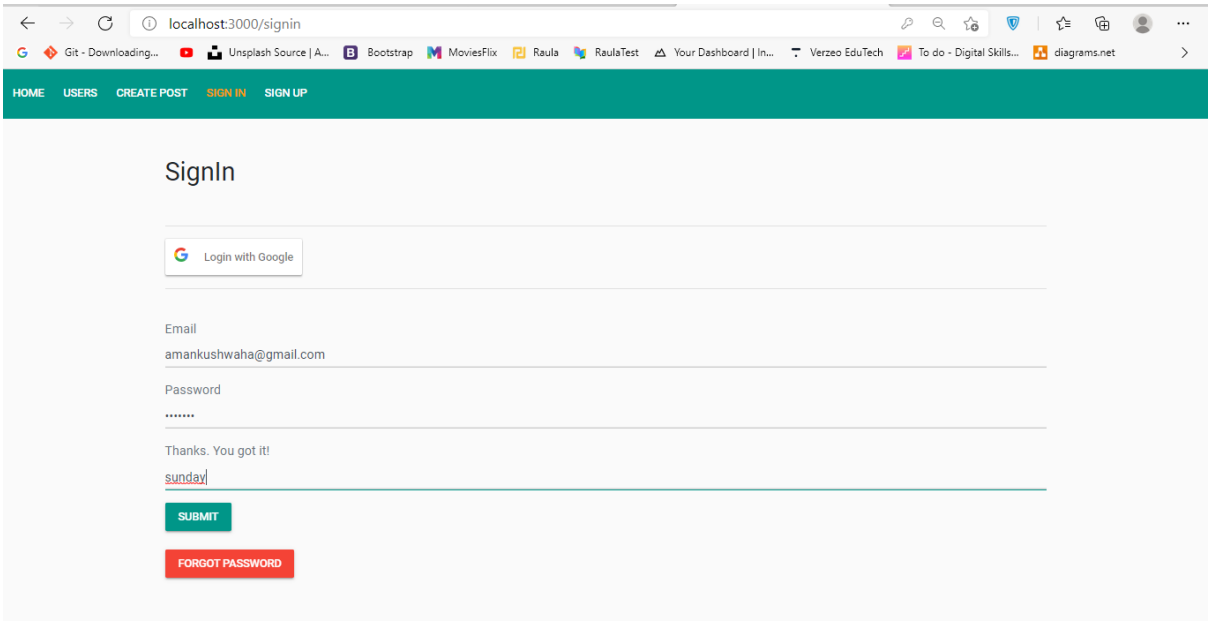
Step 2: Verify Your Email Address:

Users will do this to get free traffic to get people's attention sharing links for free



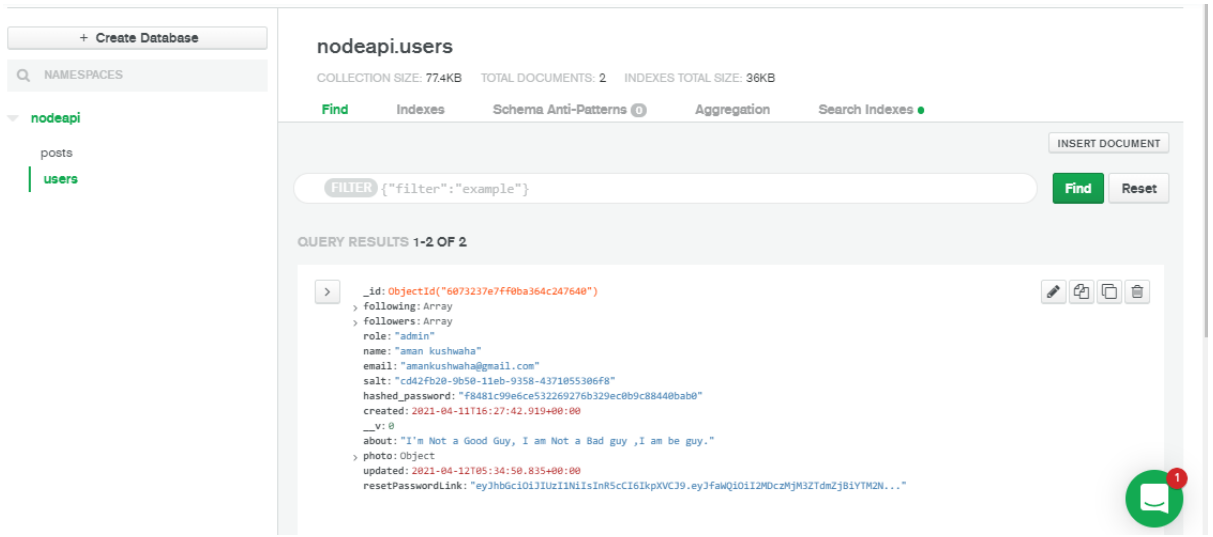
B. Log in:

Step 1: Log In Authorization In Client Side:



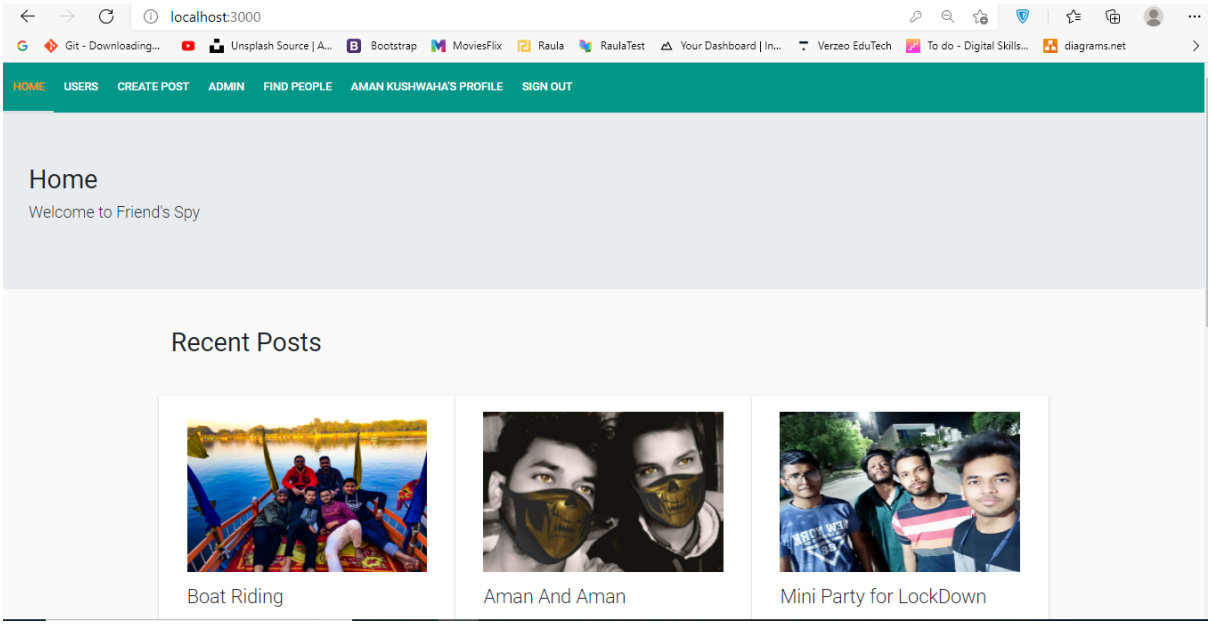
Step 2:

Log in Authorization in Mongo DB:



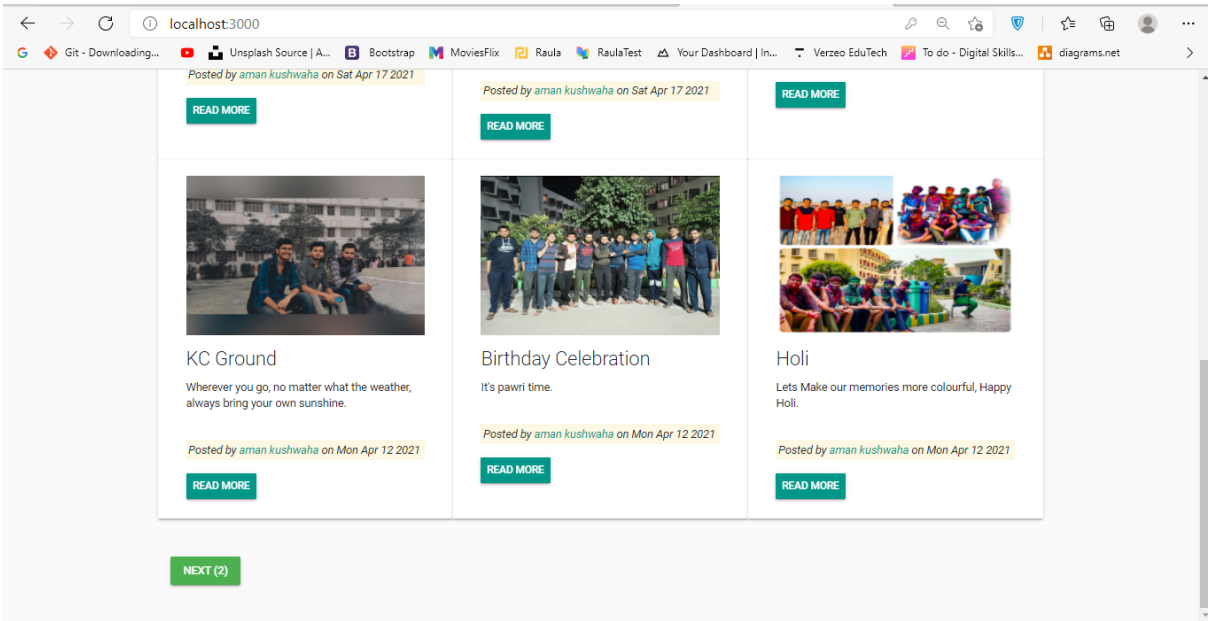
C.)Home Page:

For Extremely simple architecture that is easy to maintain and scale we have created Separate backend API and frontend built with react/nextjs

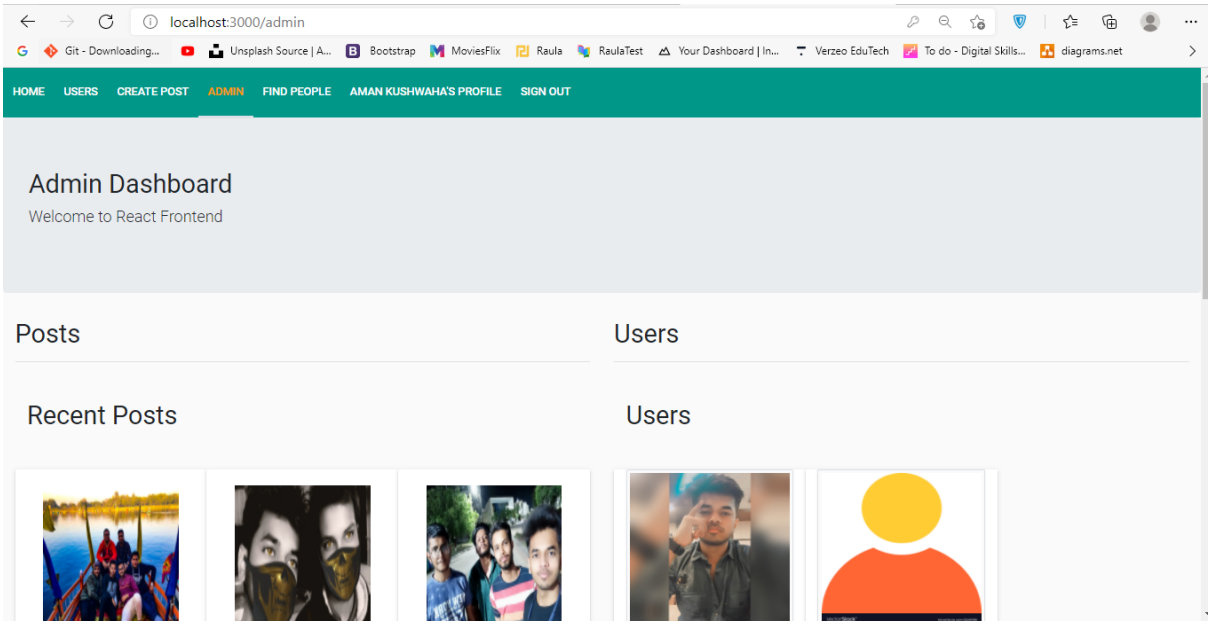


D.) Browse category:

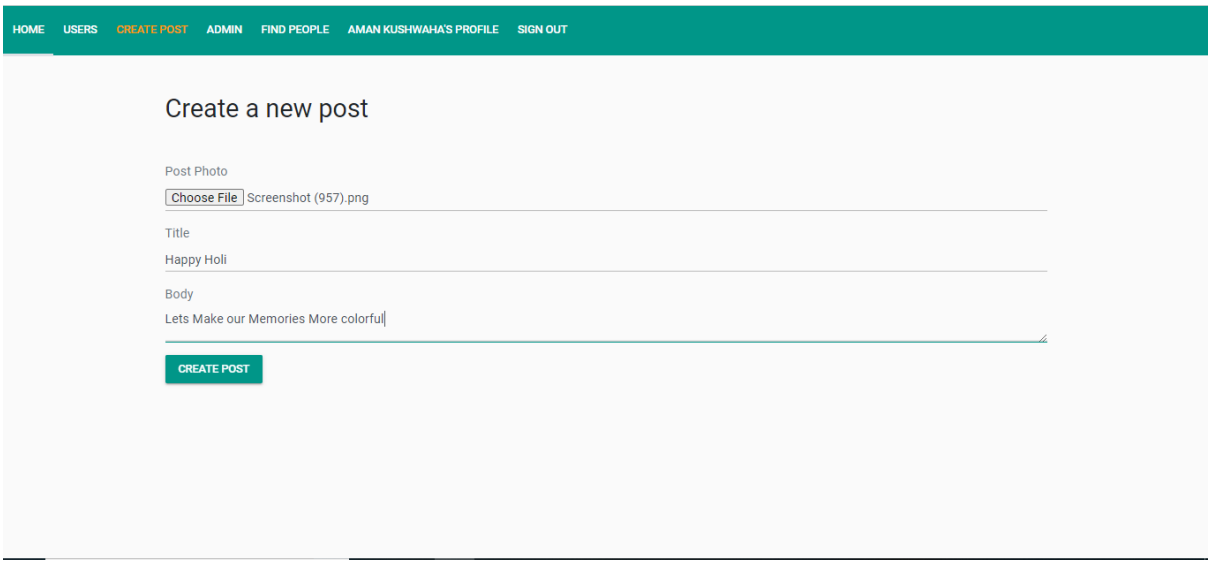
We have given Extremely simple architecture that is easy to maintain and scale.



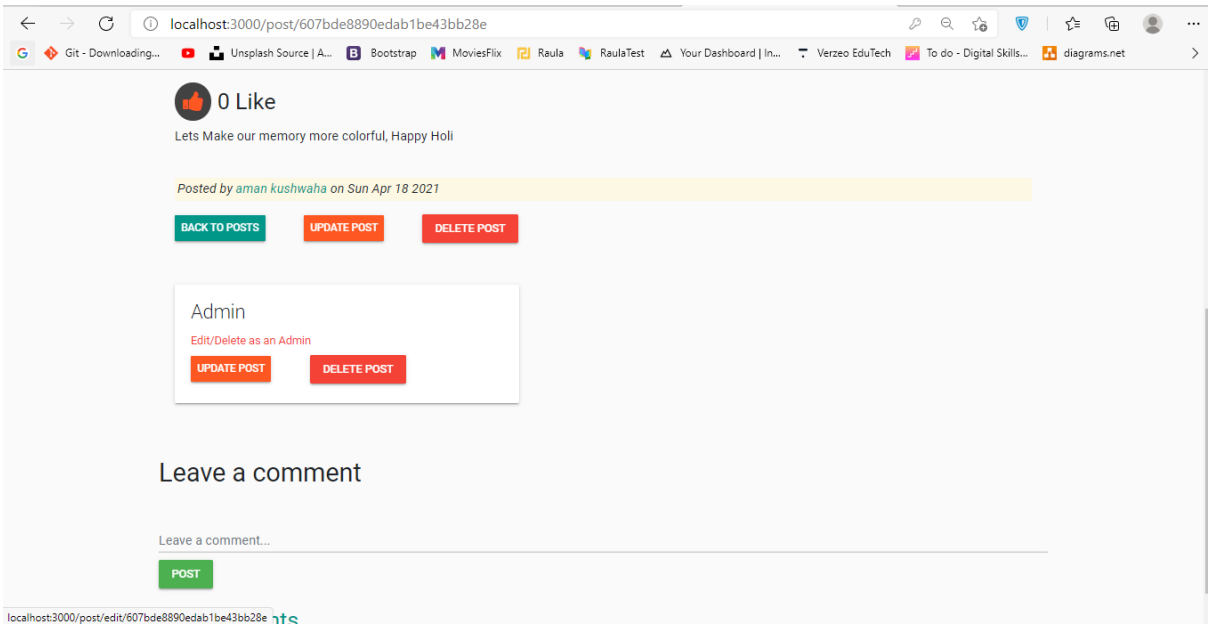
E.) Admin Accessibility:

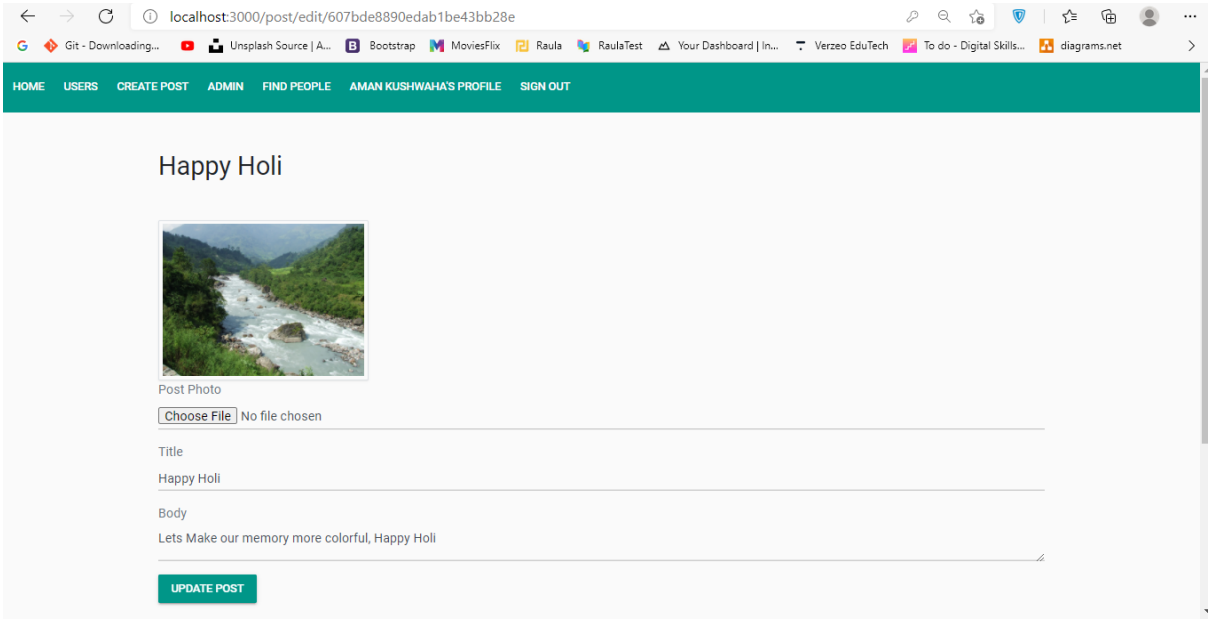


F). Create Post:

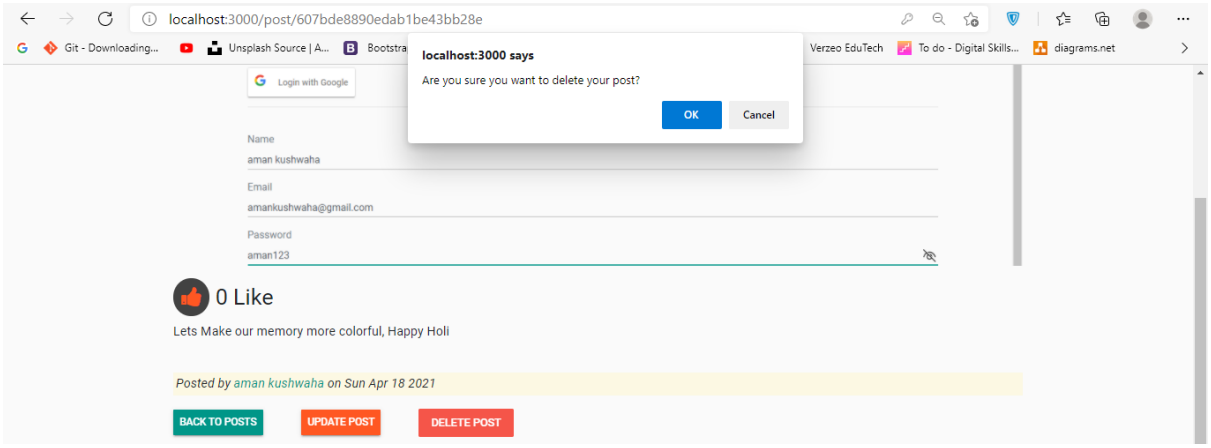


G).Update Post:



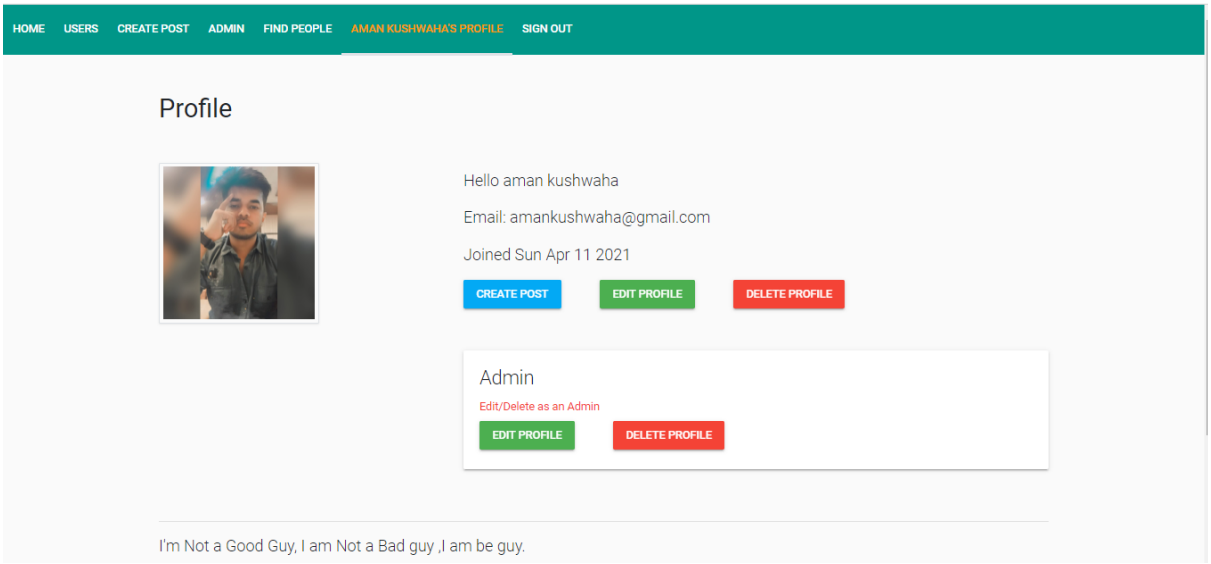


H. Delete Post:



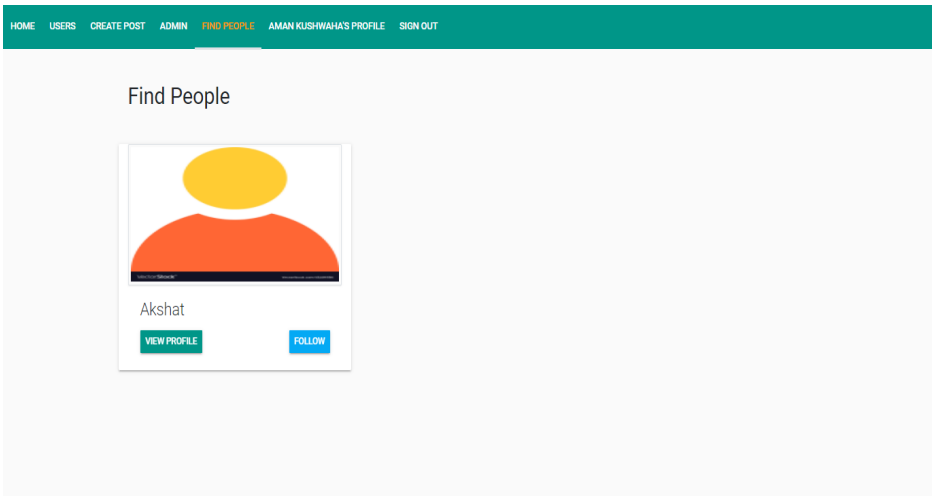
4.12 Delete Post

I.) Update profile(Both Admin and user Side):

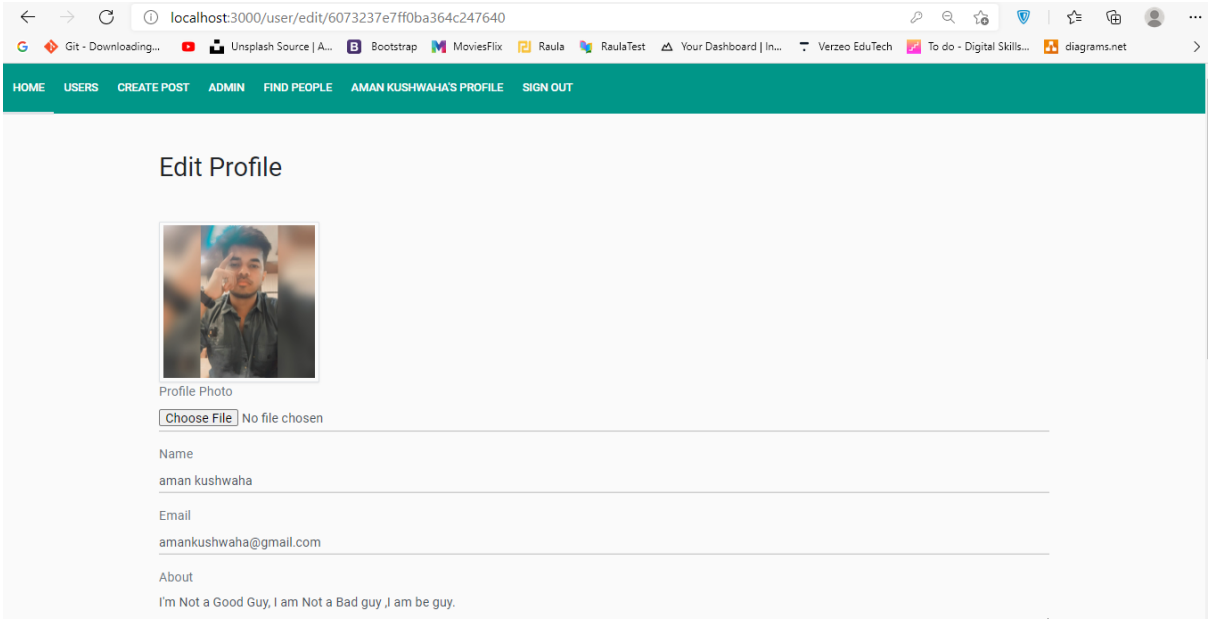


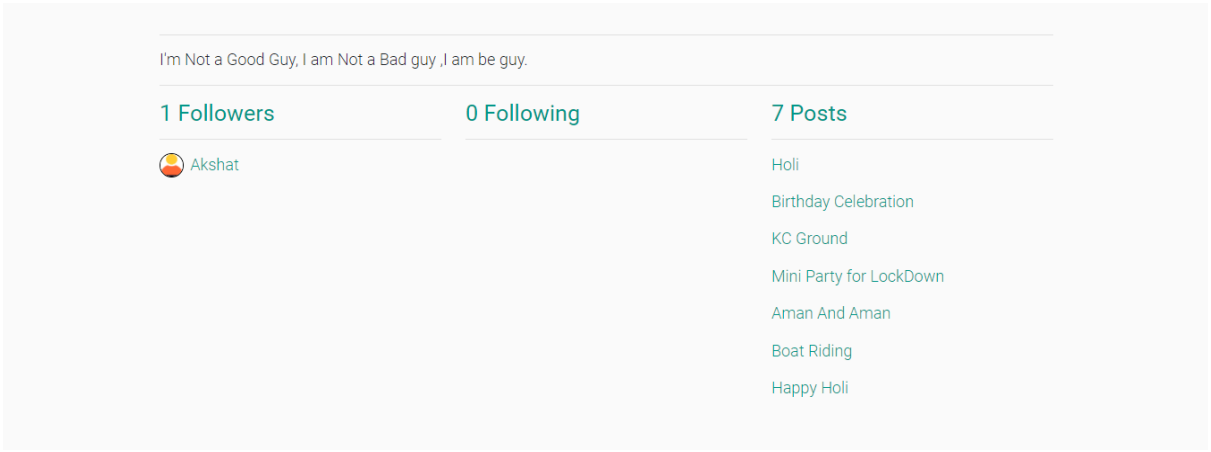
4.19 Update Profile panel

J.) FIND PEOPLE



K.) User Profile Update:





5. Software Testing

The software developed in the Friend's SPy project is integrated with the hardware testing environment and the system was tested. Integration tests were carried out in two phases, Client test scenarios written in the first phase were executed and Mongoo DB test scenarios failed as a result 3 out of 100. The success rate in the first phase was determined as 99.79%.

Test Environment: The errors that appeared in the first phase were corrected and the second phase was passed. In the second phase, a re-test was performed to check whether the errors in the first phase were removed. No problems were encountered in this confirmation test conducted in the second phase, and the success rate of the tests was determined as 100%.

Module	Total Tests Scenarios	Successful Test Scenarios	Failed Test Scenarios
Client Side	100	97	3

6. Conclusion

It will be a wonderful learning experience for us while working on this project. We decided to work on this project because we want to promote quality learning experience with our application.

We have made a good interactive User Interface so that the user will not face any difficulty while using our application.

During the development of this project we have learned different skills like –

How to work under pressure.

How to work in a team and manage our work.

Knowledge of new technologies

Our project is completed but we will continuously try to make it more reliable, secure and add more features.

7. Summary

This website will allow access only to authorized users with specific roles (administrator, user) depending upon user's role, he/she will be able to access only specific modules of the system.

A summary of the major functions that the website will perform are as follows:

-Login facility for enabling only authorized access to the system.

-Administrator will be able to add/modify/delete/update and alter data (i.e. product details) at the back-end as per the requirements.

-Administrator will be responsible for managing user account.

8. References

- Barrell, Dylan. Agile Accessibility Explained: A practical guide to sustainable accessible software development, Amazon Digital Services, 2019.
- Blanck, Peter. eQuality: The Struggle for Web Accessibility by Persons with Cognitive Disabilities, Cambridge Disability Law and Policy Series, 2015.
- Burgstahle, Sheryl. Universal Design in Higher Education: From Principles to Practice, Harvard Education Press, 2008.
- Byrne, Jim. 60 hot to touch Accessible Web Design tips – the tips no web developer can live without!, Jim Byrne, 2006, (ISBN: 978-1-4116-6729-7).
- Chisholm, and May. Universal Design for Web Applications: Web Applications That Reach Everyone, O'Reilly Media, 2008.
- Clark, Joe. Building Accessible Websites, New Riders Publishing, 2002.

- Coolidge, Doner, Robertson, and Gray. Accessibility Toolkit - 2nd Edition, BCcampus, 2020.
- Coombs, Norman. Making Online Teaching Accessible, Jossey-Bass, 2010. “Making Online Teaching Accessible" is downloadable from Bookshare.
- Cunningham, Katie. The Accessibility Handbook, O'Reilly Media, 2012.
- Duckett, Jon. Accessible XHTML and CSS Web Sites Problem Design Solution, Wrox, 2005.
- Feingold , Lainey.Structured Negotiation: A Winning Alternative to Lawsuits , American Bar Association, 2016.
- Gay, Greg et al.Digital Accessibility as a Business Practice, Ryerson University Pressbooks, 2017.
- Gay, Greg et al.Introduction to Web Accessibility, Ryerson University Pressbooks, 2019.
- Gay, Greg et al.Professional Web Accessibility Auditing Made Easy, Ryerson University Pressbooks, 2016.
- Gay, Greg et al.Web Accessibility for Developers, Ryerson University Pressbooks, 2019.
- Gilbert, Regine M. Inclusive Design for a Digital World: Designing with Accessibility in Mind (Design Thinking) , Apress, 2020.
- Hamraie, Aimi. Building Access: Universal Design and the Politics of Disability