

# INDUSTRIAL TRAINING REPORT

ON

***“LEARN TO BUILD A REAL TIME WEBSITE LIKE  
STACKOVERFLOW – MERN STACK”***

BY

**NULL CLASS**



**JORHAT ENGINEERING COLLEGE**



**JORHAT, ASSAM - 785007**

**Affiliated To**

**ASSAM SCIENCE AND TECHNOLOGY UNIVERSITY**

# Candidate's Declaration

I hereby declare that the work which is being presented in this report entitled "*LEARN TO BUILD A REAL TIME WEBSITE LIKE STACKOVERFLOW – MERN STACK*" is an authentic record of my own work carried out from 3<sup>rd</sup> July, 2023 to 31<sup>st</sup> July, 2023. The matter presented in this report has not been submitted elsewhere for any credit or requirement.

-----  
Hirak Jyoti Das

# Acknowledgement

I want to express my sincerest gratitude for the invaluable learning and professional experience I gained during my industrial internship at NULL Class. I am extremely thankful for the opportunity to be a part of your esteemed organization.

Throughout my internship tenure, which spanned from 3<sup>rd</sup> July, 2023 to 31<sup>st</sup> July, 2023, I had the privilege of working alongside talented and knowledgeable professionals who not only provided me with guidance but also inspired and motivated me to excel in my role. I am deeply appreciative of their mentorship and support, which played a crucial role in my growth and development.

During my time at NULL Class, I was exposed to a wide array of challenging tasks and real-world scenarios that enhanced my understanding of the industry and its operations. I am grateful to the NULL Class administration for providing me with such a significant chance. I believe I will participate in more such activities in the future. Finally, I'd like to express my gratitude to my parents and friends for their excellent comments and guidance during the completion of this project.

# Abstract

This internship project involved the development of a full-stack web application, replicating the renowned Stack Overflow platform using the MERN (MongoDB, Express.js, React, Node.js) technology stack. The aim of this endeavor was to gain practical experience in modern web development, hone skills in each layer of the stack, and comprehend the complexities of building a feature-rich Q&A platform.

The project's backend, powered by Node.js and Express.js, utilized MongoDB as the database to efficiently manage and store diverse user-generated content. This allowed for seamless integration of RESTful API endpoints, ensuring robust communication between the server and client-side components. The utilization of Express.js enabled the implementation of authentication, authorization, and middleware functionalities to enhance the overall security and performance of the application.

On the frontend, React was employed to design an intuitive and responsive user interface, offering a seamless and dynamic browsing experience for users. The use of React components facilitated modularization, code reusability, and the efficient handling of state management, contributing to a more maintainable codebase.

Key features of the StackOverflow clone included user authentication, question and answer posting, commenting, upvoting/downvoting, user profiles, and a sophisticated search functionality.

Throughout the development process, industry-standard best practices, version control systems, and agile methodologies were employed. Challenges encountered during the internship were addressed using systematic troubleshooting and collaborative problem-solving approaches. The project's success demonstrates the practical application of the MERN stack in creating a scalable, feature-rich web application, providing valuable insights into the intricacies of full-stack development.

This report comprehensively documents the project's conception, architecture, implementation, challenges faced, and lessons learned. The gained experience serves as a valuable foundation for further exploration and specialization in the rapidly evolving field of web development.

# Contents

## List of Abbreviations

### 1) Introduction

### 2) Technologies Used

- Javascript
- NodeJS
- Express.js
- MongoDB
- ReactJS
  - Virtual-DOM
  - Component
  - Props and State
  - Pros and Cons of ReactJS
- MERN Stack in Website Development
  - Concept of Stack technology
  - Concept of MERN Stack

### 3) Working of the application

### 4) Conclusion

## List of Abbreviations:

<b>DOM</b>	Document Object Model
<b>W3C</b>	World Wide Web Consortium
<b>ISO</b>	International Organization for Standardization
<b>NPM</b>	Node Package Manager
<b>API</b>	Application Program Interface
<b>HTML</b>	Hypertext Markup Language
<b>HTTP</b>	Hypertext Transfer Protocol
<b>URL</b>	Uniform Resource Locator
<b>JSON</b>	JavaScript Object Notation
<b>JS</b>	JavaScript
<b>JSX</b>	JavaScript XML
<b>XML</b>	Extensive Markup Language

# Introduction

In the ever-evolving landscape of web development, the MERN (MongoDB, Express.js, React, Node.js) stack has emerged as a powerful and versatile technology stack, empowering developers to create dynamic and scalable web applications. During my internship, I had the opportunity to delve into the intricacies of this technology stack while working on a project of significant practical and educational value – a Stack Overflow clone.

## **Purpose and Relevance:**

The Stack Overflow Clone serves as a testament to the capabilities of the MERN stack by replicating the essence of the renowned Stack Overflow platform. Stack Overflow, a widely utilized platform for programmers and developers, has revolutionized knowledge sharing within the coding community. The clone, being modeled on this established platform, aims to provide users with a familiar and user-friendly environment for seeking and sharing knowledge related to programming and software development.

## **Key Features and User Benefits:**

This project is not merely an exercise in replication; it extends the functionality of the original platform by incorporating enhancements and optimizations. Users of the Stack Overflow Clone can expect an intuitive interface that facilitates seamless navigation, efficient question-and-answer interactions, and a robust voting and reputation system. The platform encourages collaborative learning and problem-solving within the developer community, offering a feature-rich experience that fosters the exchange of ideas and solutions.

## **Performance and Scalability:**

Built on the MERN stack, the Stack Overflow Clone prioritizes performance and scalability. The use of MongoDB ensures a flexible and scalable database structure, while Express.js and Node.js contribute to the creation of a fast and responsive backend. React, on the frontend, enhances the user interface with its component-based architecture and efficient rendering.

In conclusion, the Stack Overflow Clone developed during my internship encapsulates the synergy of the MERN stack and its potential to create dynamic, scalable, and user-centric web applications. Through this project, I aimed to bridge the gap between theoretical knowledge and practical application while offering a valuable tool for developers to engage in collaborative learning and knowledge sharing.

# Technologies Used

## 1) MERN Stack

### 1.1 JavaScript

JavaScript is a scripting, object-oriented, cross-platform programming language. Objects of host environment can be connected to JavaScript and arrange ways to operate them. Standard libraries for objects are contained by JavaScript, for such as Array, Date, Math, and the essence component of programming languages for instance managers, control framework and statements. By adding objects, JavaScript could be protracted for many principles, such as:

**Client-side JavaScript:** JavaScript is developed by implementing objects for controlling the browser and DOM. For instance, an application is granted by client-side extensions to influence components on an HTML page and answer to user behavior like mouse hovers, form input and page changeover.

**Server-side JavaScript:** JavaScript is developed by implementing the supplementary objects required to run JavaScript on the server. For instance, an application is granted by this server-side extension to connect to a database, transfer data frequently from one request to other section of the application or execute application with another function file on the server.

In 1996, JavaScript was officially named ECMAScript. ECMAScript 2 was released in 1998 and ECMAScript 3 was released in 1999. It is continuously evolving into today's JavaScript, now works on all browsers and devices from mobile to desktop. Open standard language can be used by association to establish their own JavaScript applications. The ECMAScript Standard is one of the parts of the ECMA-262 specification. ISO has approved the ECMA-262 standard at ISO-16262. The ECMAScript standard does not include descriptions for the DOM, it is standardized by the W3C. The DOM specifies how your scripts display HTML objects. To get a advance anticipate of the distinctive innovations used when programming with JavaScript, check out the JavaScript technology analysis article.



## 1.2 NodeJS

Node.js is an open source, a system application and furthermore is an environment for servers. Node.js is an independent development platform built on Chrome's JavaScript Runtime that we can build network applications quickly and easily. Google V8 JavaScript engine is used by Node.js to execute code. Moreover, a huge proportion of essential modules are written in JavaScript. Node.js accommodate a built-in library which allows applications to serve as a Webserver left out demanding software like Apache HTTP Server, Nginx or IIS. An event-driven, non-blocking I / O mechanisms (Input / Output) are implemented by Node.js. It optimizes application throughout and is extremely high extensible. Node.js use asynchronous in its functions. Therefore, Node.js processes and executes all tasks in the background (background processing). products that have a lot of traffic are applying Node.js. Nonetheless, Node.js handle the application that need to spread expeditiously, develop innovation, or build Startup projects as rapidly as possible.

### **Applications using NodeJS:**

- WebSocket server
- Notification system
- Applications that need to upload files on the client.
- Other real-time data applications.

### **NodeJS Pros:**

- Node.js is the exclusive application that with only a single thread, it can obtain and handle numerous connections. Building new threads for each query is not needed, therefore the structure expends the least amount of RAM and run rapidly. Secondly, Node.js produces the most of server property without generate latency with the JavaScript's non-blocking I/O.
- **JSON APIs.** JSON Web services can take advantages of that because of the event-driven, non-blocking I/O structures and JavaScript-enabled model.
- **Single page application.** NodeJS is very suitable with an application on a single page. Node.js has the capability to handle different requests concurrent and quick return. Node JS should be used in an application that does not have to reload the page, including users who makes a vast number of requests and need a quick procedure to show professionalism.

- **Shelling tools Unix.** Node.js usually uses Unix to work. They can handle multiple processes and return them for best performance. Programmers often use Node.js to build real Web applications like chat, feeds, etc.
- **Streaming Data.** Typical websites send HTTP requests and also receive responses. Node.js can handle many questions and feedback, so they are suitable if the developer wants to create an application on the page. In addition, Node.js also builds proxies to stream the data, this is to ensure maximum operation for other data streams.
- **Real-time Web Application.** Node.js is sufficient to develop real-time innovations like chat apps, social networking services like Facebook, Twitter because of the opening of mobile application.

#### **NodeJS Cons:**

- **Resource-intensive applications.** Node.js is written in C ++ & JavaScript, so when programmers need to handle applications that use a lot of file conversion, video encoding, decoding, etc., they should not be used Node.js. Programmers need to use it more carefully in this case.
- The final purpose of NodeJS is like other programming languages such as Ruby, PHP, .NET, Python, that is developing web application. Therefore, do not expect NodeJS to outperform other language for now. But with NodeJS the application can be developed successfully as expected.
- **Build resource-intensive applications:** Do not use Node.js when creating a video converter application. Node.js often comes down to bottlenecks when working with large files.
- **An all-CRUD-only application:** Node.js is not faster than PHP when doing heavy I/O tasks. In addition, with the long-term stability of other webserver scripts, its CRUD tasks have been optimized. Node.js will come up with odd APIs and never be used
- **Stability in the application:** Within 11 years of development (2009-2020), the current version of Node.js is already v14.2.0. Every API can be changed – in a way that is not backwards compatible.
- **Lack of knowledge about Node.js:** Node.js is extremely dangerous in this case, you will fall into a world full of difficulties. With most non-blocking/async APIs, not understanding the

problem will cause an error that you do not even know where it came from. Moreover, when the Node.js community is not strong enough, and there will be less support from the community.

**NodeJS should be used when:**

- **Building RESTful API (JSON).** You can use Node.js in building RESTful API (JSON). They handle JSON very easily, even more than JavaScript. API servers when using Node.js usually do not have to perform heavy processing, but the number of concurrent requests is high.
- **Applications that demand alternative connection protocols,** not just http. With TCP protocol backing, any custom protocol can be built easily.
- **Real-time applications.**
- **Stateful websites.** Every request on the invariable procedure is handled by Node.js, therefore building caching is simpler: store it to a comprehensive variable then all requests can approach the cache. The status of one client can be saved and shared with other clients and do not have to go through external memory.

## 1.3 Express.js

Express.js is a framework built on top of Node.js. It provides powerful features for web or mobile development. Express.js supports HTTP and middleware methods, making the API extremely powerful and easy to use. Express implements extra features to developer which help them get a better programming environment, not scaling down the speed of NodeJS. Importantly, the well-known frameworks of NodeJS apply Express.js as a substance function, for instance: Sails.js, MEAN.

## 1.4 MongoDB

MongoDB is an open-source database; it is also the leading NoSQL database currently used by millions of people. It is written in one of the most popular programming languages today. In addition, MongoDB is cross-platform data that operates on the concepts of Collections and Documents, providing high performance with high availability and ease of expansion. NoSQL is a source database format that does not use Transact-SQL to access information, this database was developed on JavaScript Framework on JSON data type. With its introduction, it has

overcome the disadvantages of RDBMS relational data model to improve operating speed, functionality, model scalability, cache ...

Furthermore, MongoDB is a cross-platform database, performing on Collection and Document approach, it produces sharp production, huge availability, and effortless scalability.

### **Commonly used terms in MongoDB:**

- **\_id:** Almost every document required this field. The `_id` field illustrates a exceptional value in the MongoDB document. The `_id` field can also be interpreted as the primary key in the document. If you add a new document, MongoDB will automatically generate a `_id` representing that document and be unique in the MongoDB database.
- **Collection:** A group of many documents in MongoDB. Collection can be interpreted as a corresponding table in the RDBMS (Relational Database Management System) database. Collection resides in a single database. Collections do not have to define columns, rows or data types first.
- **Cursor:** This is a pointer to the outcome set of a query. The client can emphasize over a cursor to get the result.
- **Database:** The location of the collections, similar to the RDMS database that contains the tables. Each Database has a separate file stored on physical memory. Some MongoDB owners may contain various databases
- **Document:** A transcript belonging to a Collection. Documents, in turn, include name and value fields.
- **Field:** A name-value pair in a document. A document may not need all the fields. The fields are like columns in a relational database.
- **JSON:** Short for JavaScript Object Notation. Human readability is in the plain text format representing structured data. JSON currently supports a lot of programming languages.
- **Index:** Exclusive data structures used to save a small allocation of data sets for simple scanning. The index puts the value of a individual field or sets of fields, sorted by the value of these fields. Index effectively supports the analysis of queries. Without an index, MongoDB will have to scan all the documents of the set to choose the documents that pair the query. This scan is ineffective and requires MongoDB to progress a vast amount of data.

MongoDB Atlas is MongoDB's cloud database launched in 2016 on AWS, Microsoft Azure and Google Cloud Platform.

The data in each Cluster in the Atlas is stored by Replication mechanism, with 3 nodes: 1 master (primary) and 2 slaves (secondary).

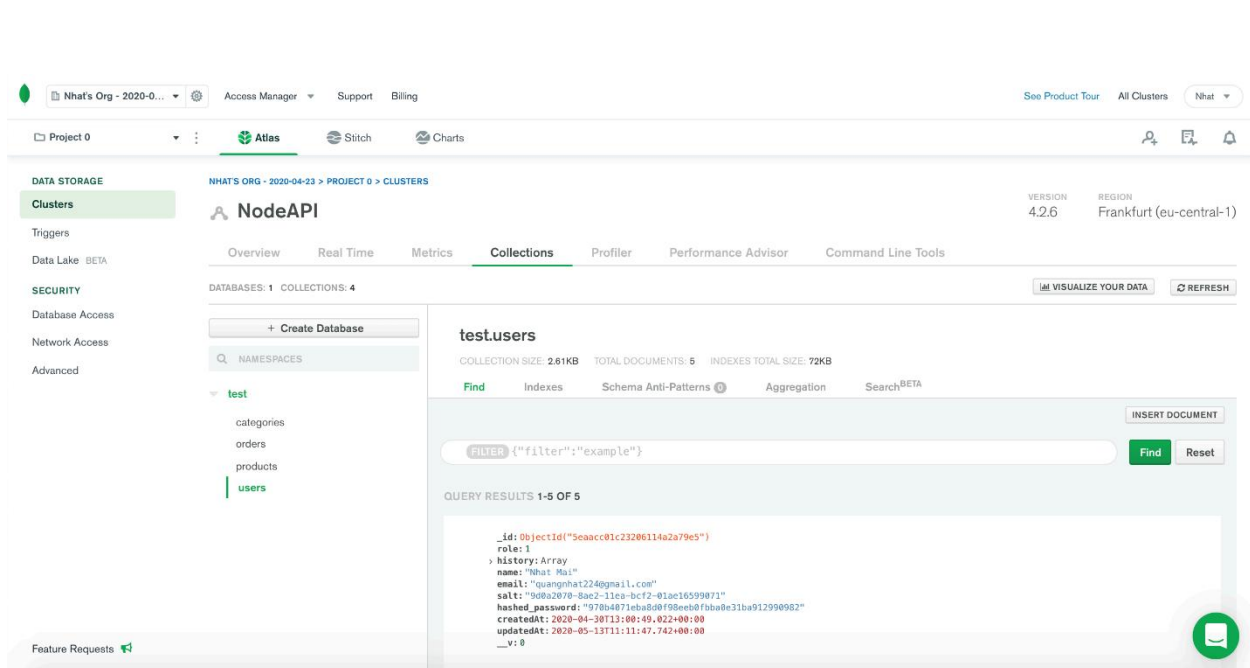


Figure 1. MongoDB Atlas screenshot.

## 1.5 ReactJS

### 1.5.1 Virtual-DOM

Virtual-DOM is a JavaScript object, each object contains all the information needed to create a DOM, when the data changes it will calculate the change between the object and the real tree, which will help optimize re-render DOM tree. It can be assumed that is a virtual model can handle client data.

### 1.5.2 Component

React is built around components, not templates like other frameworks. A component can be created by the create Class function of the React object, the starting point when accessing this library.

ReactJS creates HTML tags unlike we normally write but uses Component to wrap HTML tags into stratified objects to render.

Among React Components, render function is the most important. It is a function that handles the generation of HTML tags as well as a demonstration of the ability to process via Virtual-DOM.

Any changes of data at any time will be processed and updated immediately by Virtual-DOM.

### 1.5.3 Props and State

**Props:** are not controlled by Component, actually stands for Properties.

```
1  import React from 'react';
2  import Menu from './Menu';
3  import '../styles.css';
4
5  const Layout = ({
6    title = "Title",
7    description = "Description",
8    className,
9    children
10 }) => (
11   <div>
12     <Menu />
13     <div className="jumbotron">
14       <h2>{title}</h2>
15       <p className="lead">{description}</p>
16     </div>
17     <div className={className}>{children}</div>
18   </div>
19 );
20
21 export default Layout;
```

Figure 2. Layout component.

The title = "Title" line creates a name attribute with the value "Title". It looks like a function call. It is true that props are passed to the component in the same way that an argument is passed to a function.

A component may also have a default props, so if the component does not pass any props, it will still be set.

**State:** private data is controlled by Component.

Like props, state also holds information about the component. However, the type of information and how to handle it varies. State works differently than Props. The state is a component of the component, while props are passed in from the outside into the component. It should be noted that we should not update the state directly using `this.state` but always use `setState` to update. Update the state of the objects. Use `setState` to re renders one component and all its children. This is great, because we don't have to worry about writing event handlers like any other language.

### **1.5.4 Pros and Cons of ReactJS**

#### **Pros of ReactJS:**

- Update data changes quickly.
- React is not a framework so it offloads the constraints of libraries together.
- Easy access to who understands JS.

#### **Cons of ReactJS:**

- ReactJS only serves the View tier, but the library size is on par with Angular while Angular is a complete framework.
- Incorporating ReactJS within common MVC frameworks demands reconfiguration.
- Hard to reach for beginners on website developing.

## **1.6 MERN Stack in Website Development**

### **1.6.1 Concept of Stack technology**

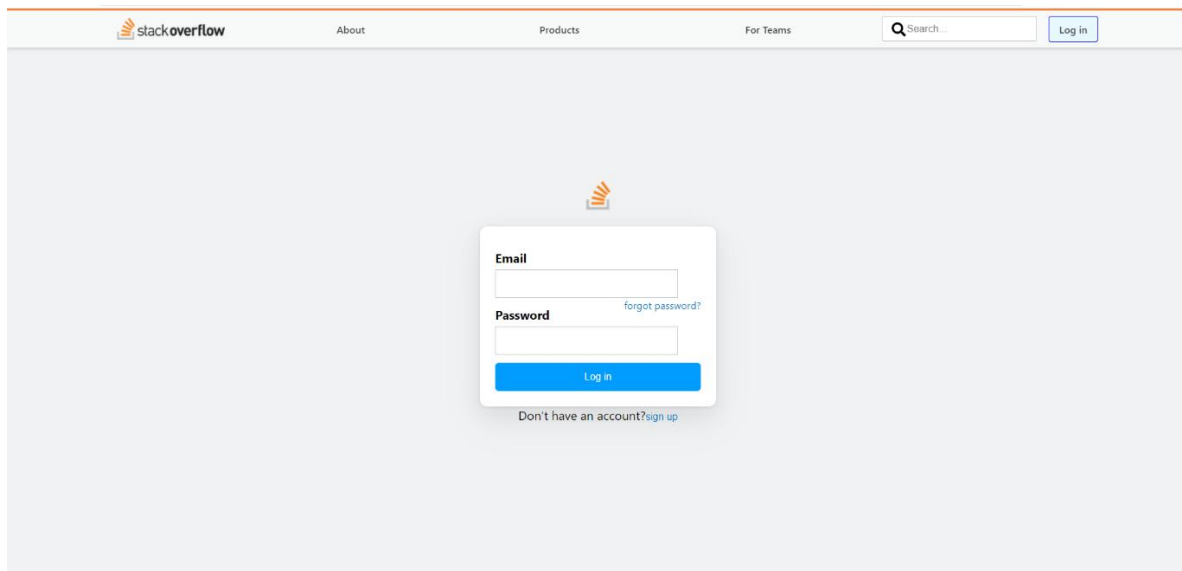
The technical stack is a combination of technologies / frameworks / programming languages, etc. to create a complete software. With current software, there are usually two parts: client side and server side, also known as frontend and backend. Therefore, people also split the backend stack, the frontend stack as well. We often use the first letter to name the technical stack: LAMP (Linux, Apache, MySQL, PHP), MEAN (MongoDB, Express, Angular, NodeJS).

### **1.6.2 Concept of MERN Stack**

The MERN stack is a complete open-source combination, all JavaScript-related technologies are also the hottest today: MongoDB, Express.js, React / React Native, NodeJS. People use the MERN stack to build Universal React App.

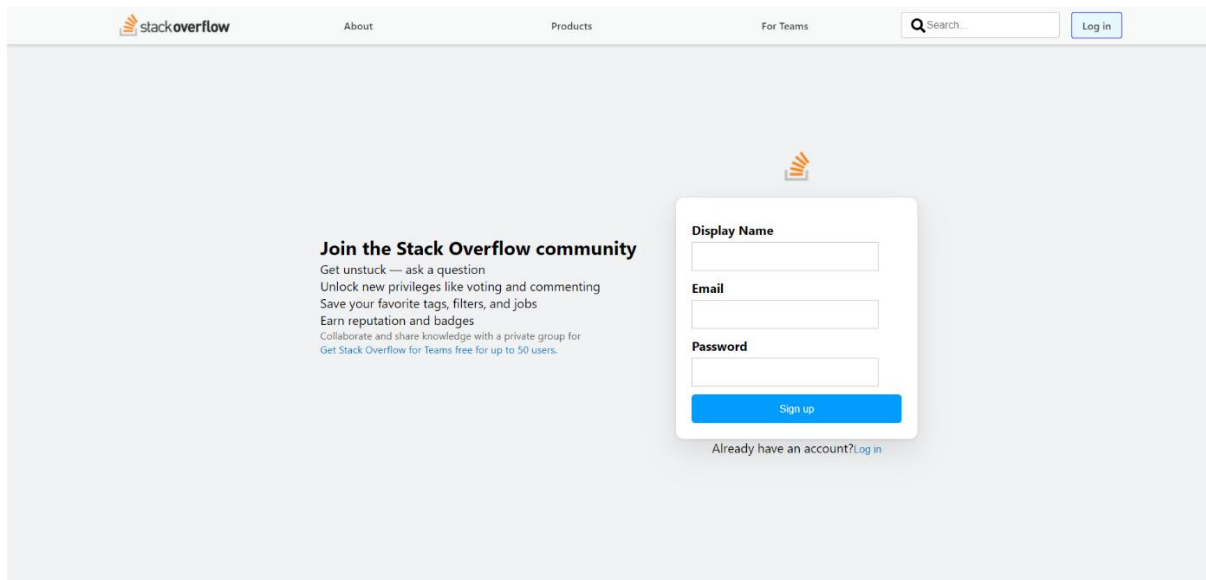
## Working of the application

**1)** Firstly, a user needs to login into the application. The login page contains 2 fields: Email and Password using which a user can login into their account.

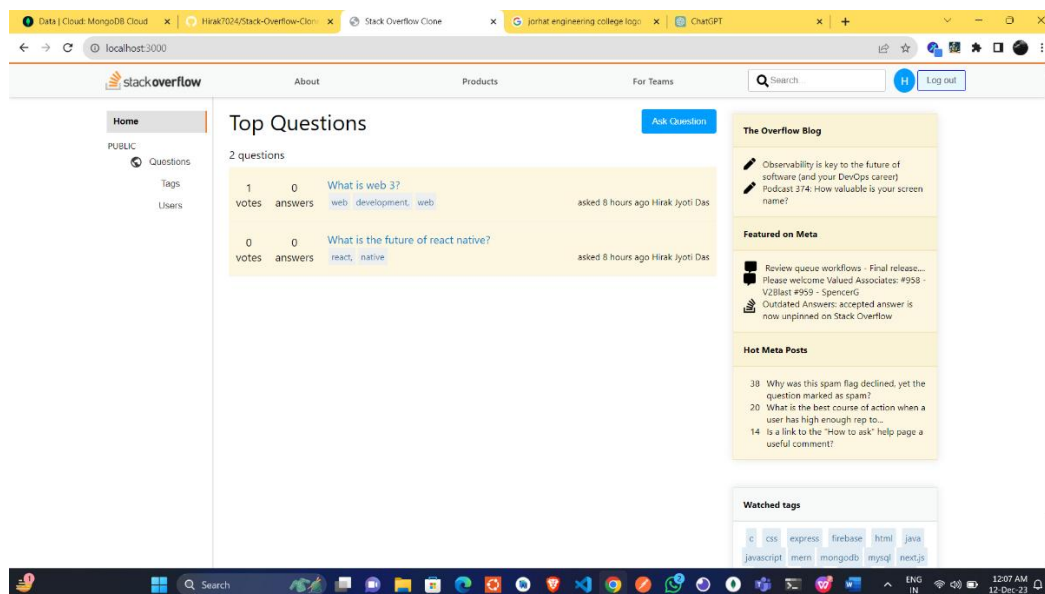
A screenshot of a web application's login page. The page has a light gray background. At the top, there is a navigation bar with the 'stackoverflow' logo on the left, and links for 'About', 'Products', and 'For Teams' in the center. On the right side of the navigation bar, there is a search bar with a magnifying glass icon and the text 'Search...', and a 'Log in' button. In the center of the page, there is a white login form with a small orange icon above it. The form has two input fields: 'Email' and 'Password'. Below the 'Password' field, there is a link that says 'forgot password?'. At the bottom of the form is a blue 'Log in' button. Below the form, there is a link that says 'Don't have an account? sign up'.

**2)** If the user is not registered, then, below the login form is a signup link. Users can click on that link to go to the signup page where they can register themselves. The register page contains 3 fields: Name, Email and Password. Using these three fields, a user can register themselves in the application.



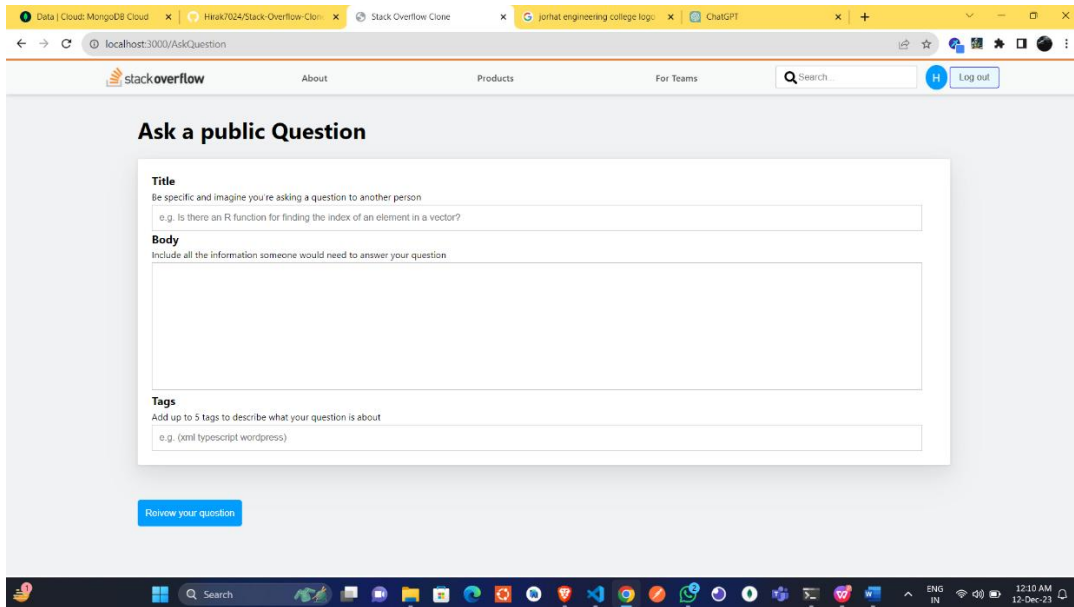


3) After a user successfully logs in, user gets directed to the Homepage. The Homepage of the application looks as follows:



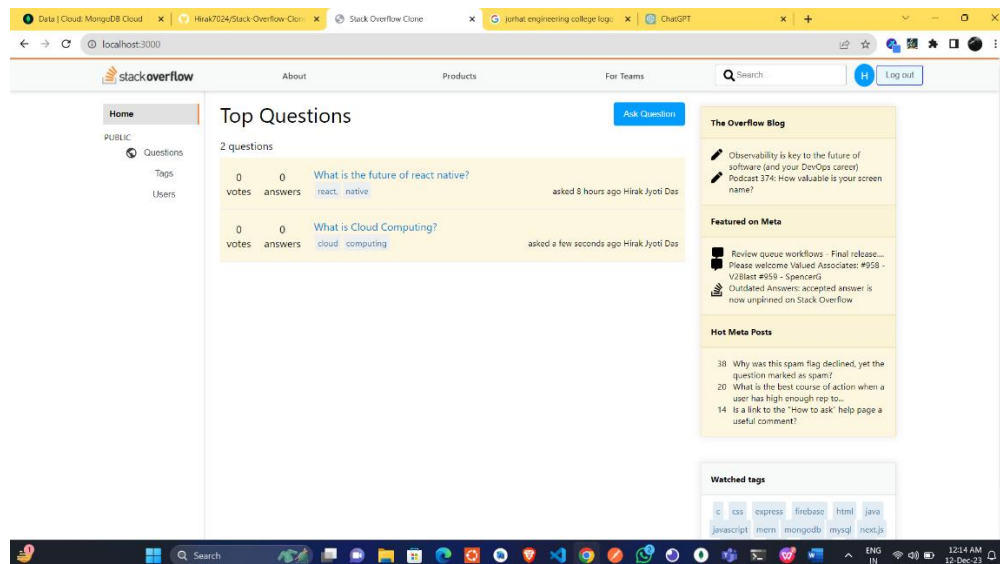
At the top of the Homepage is the Navbar. We can also see there is a left sidebar, question section and right sidebar. The Question section contains all the questions that different users have asked.

4) As we can see in the homepage, there is a button “Ask Question”. On click of that button, a user goes to a new page where they can ask their question:

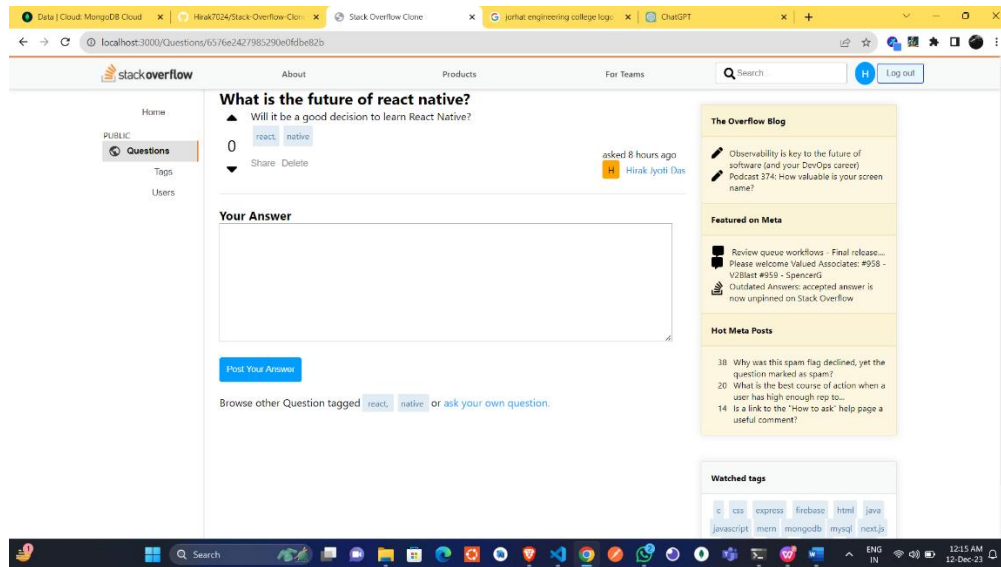


A user can fill all these details about their question and then they can post their question which will be displayed on the Homepage.

5) The new question will be displayed in the Homepage as:

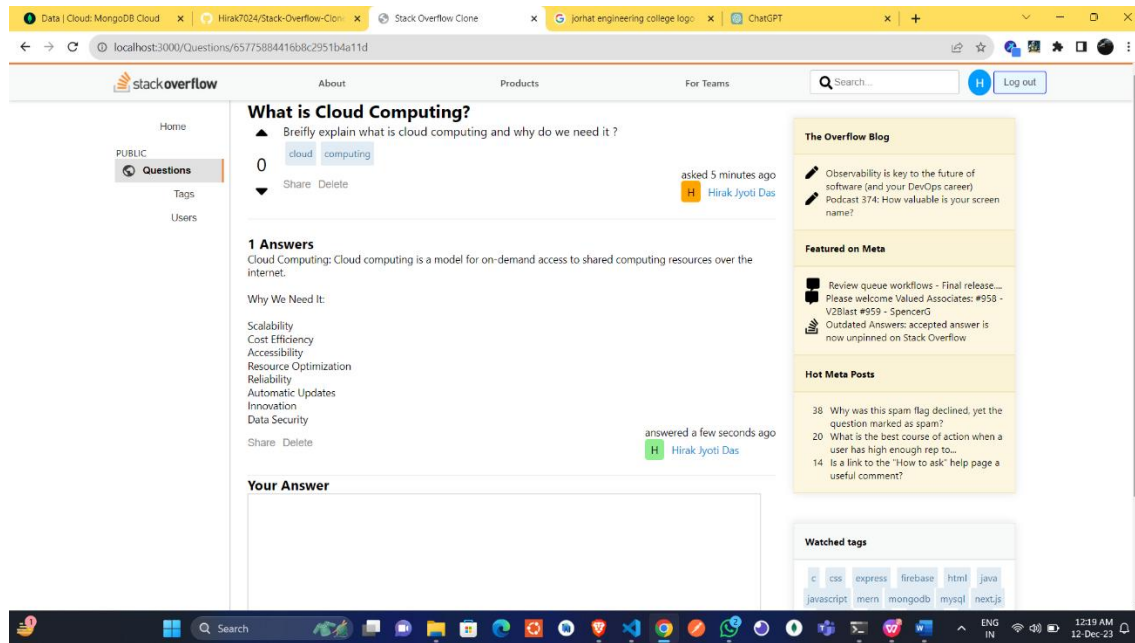


6) A different user can click on the question and then they will be redirected to a page where they can post an answer for the question:

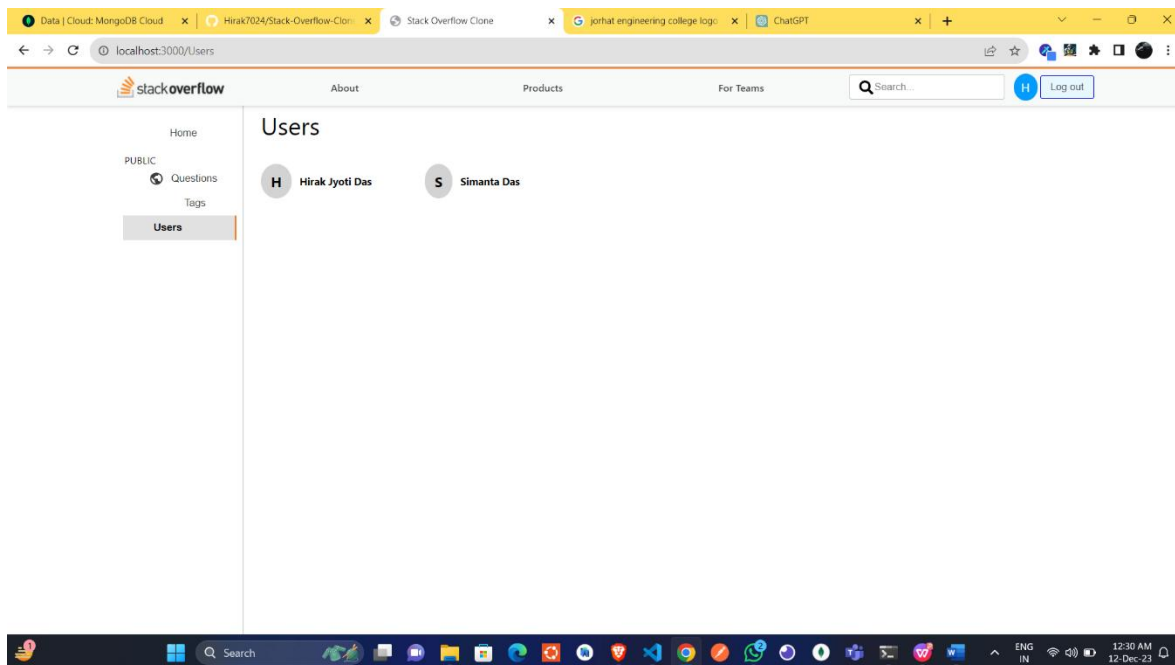


We can see that there is option to delete and share the question. Users can share the question and only that user who has posted that question can delete the question on click of the delete button. Different users can post an answer for the question. After typing the answer in the answer box, user can click on the “Post Your Answer” button to post their answer. Users can also upvote or downvote the question.

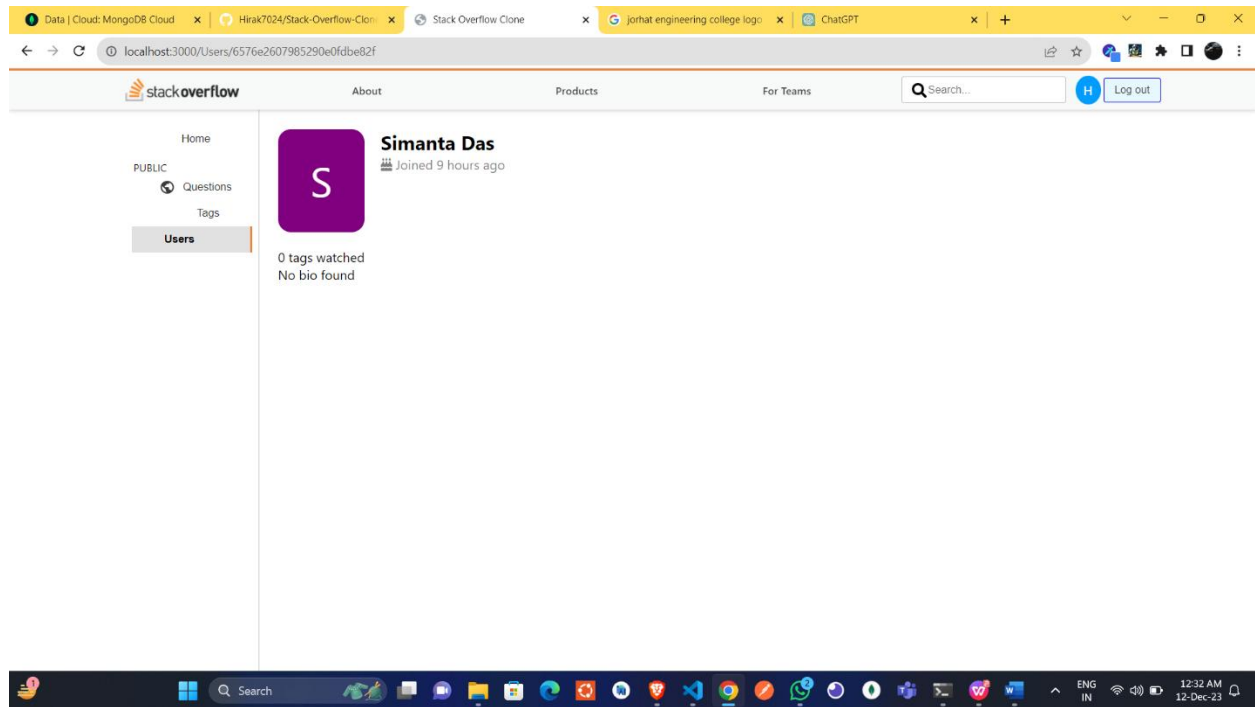
7) Now, as we can see there is an answer for the question. In this way different users can post their answers for the question which will be showed right below the question. A user can also delete or share the answer that they have posted. Users can delete only those answers that they have posted and not the answers posted by others.



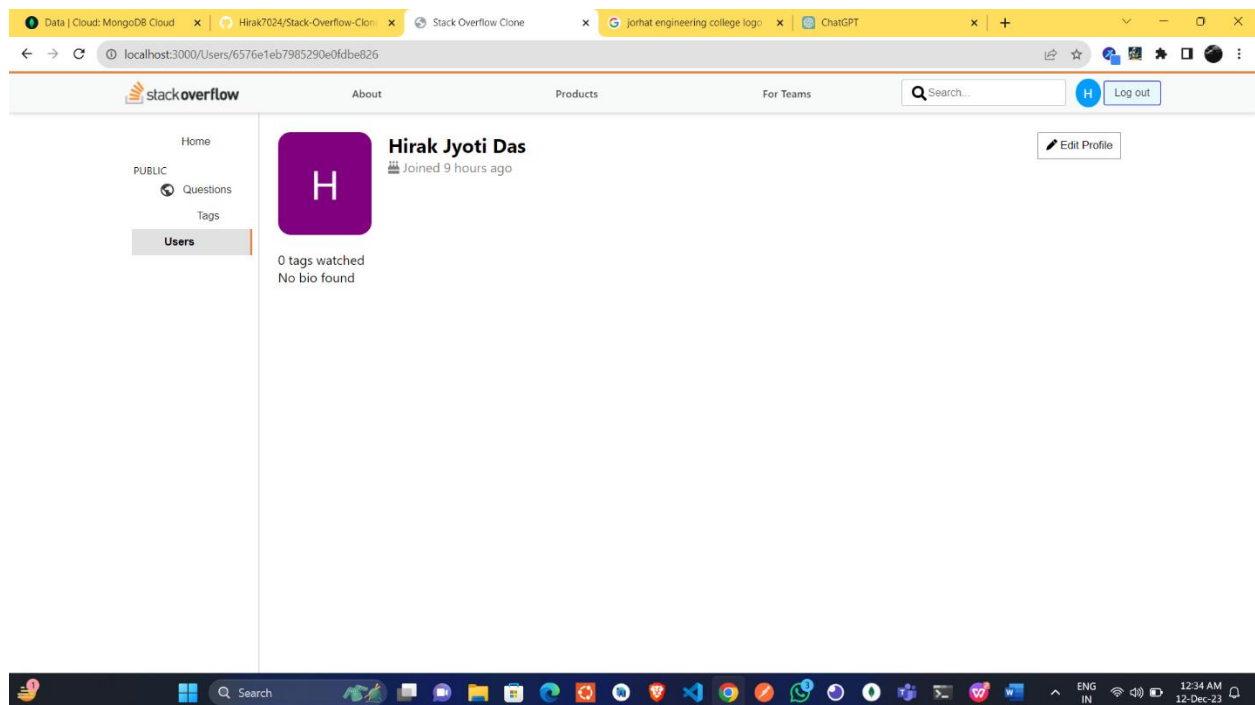
8) In the homepage, when user clicks on the “Users” tab from left sidebar, then the user is redirected to a new page where all the users that have registered in the application are displayed:



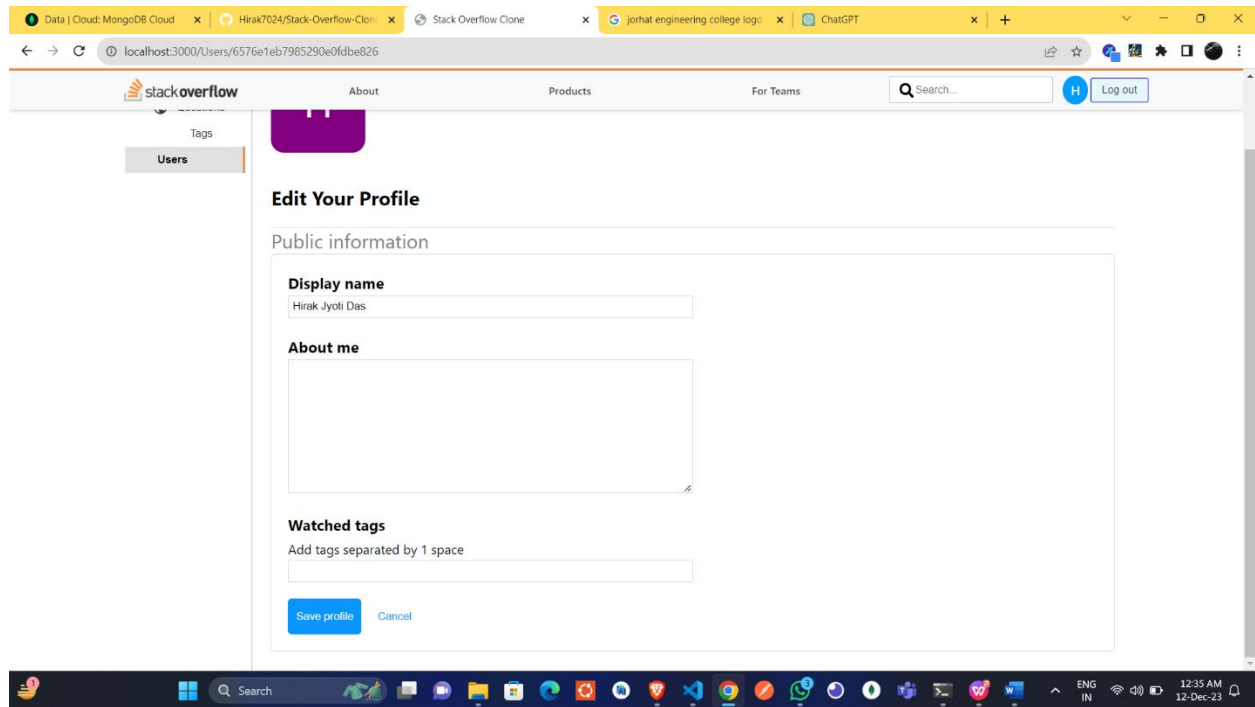
9) When user clicks on any of these users, they can see the details of the user:



10) If the user clicks on his own name, then his profile will be showed along with an “Edit Profile” button:

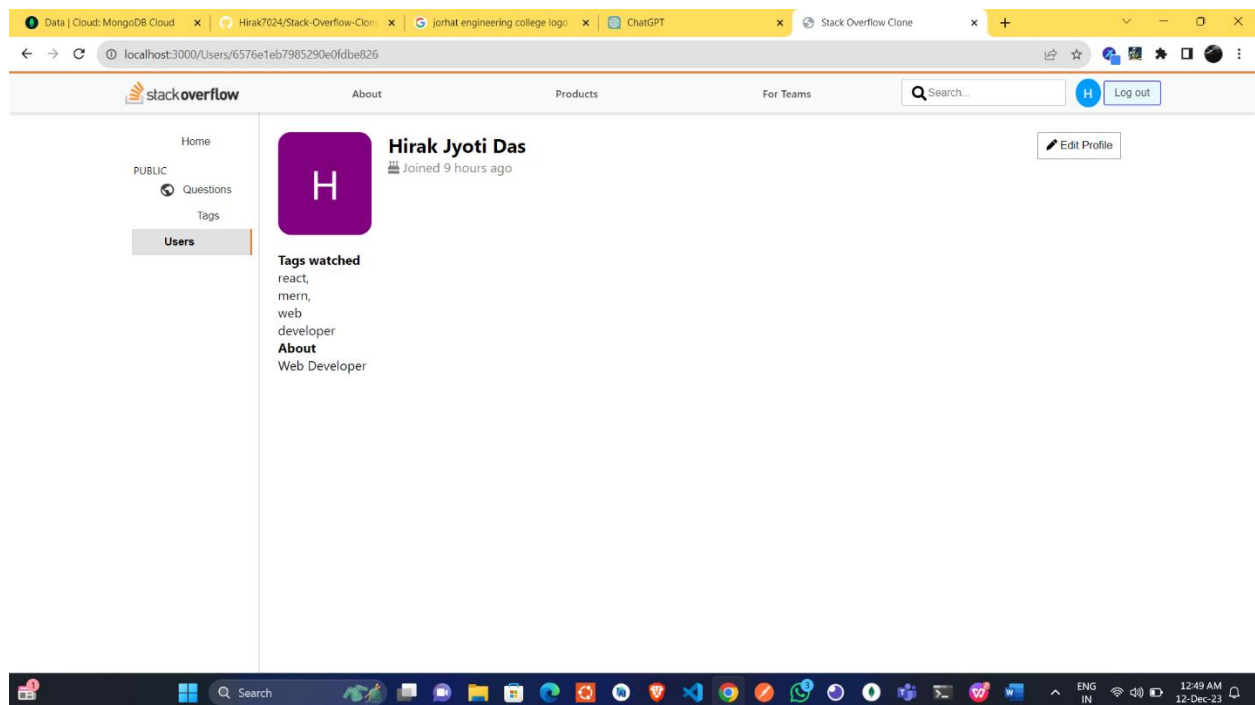


On Click of the “Edit Profile” button, a form appears where the user can put their details:



After entering the user details, user can click on the “Save Profile” button to save their profile details.

The updated user profile will be displayed like this:



# Conclusion

In concluding the development and exploration of the Stack Overflow Clone using the MERN stack, this internship project has been a transformative journey into the world of modern web development. The successful creation of a feature-rich, responsive, and scalable platform that mirrors the functionality of Stack Overflow underscores the prowess of the MERN stack in crafting dynamic and interactive web applications.

Through this project, I have not only honed my technical skills but have also gained invaluable insights into the importance of collaborative platforms within the developer community. The Stack Overflow Clone stands as a testament to the adaptability and efficiency of the MERN stack, showcasing its ability to streamline the development process and deliver a user-friendly experience.

As technology continues to advance, the significance of robust, scalable, and user-centric web applications becomes increasingly pronounced. The lessons learned from this internship project extend beyond the confines of code, emphasizing the importance of continuous learning, adaptability, and community engagement in the field of software development.

In the spirit of open-source collaboration, the codebase of the Stack Overflow Clone is presented not only as a culmination of my efforts but as a contribution to the wider developer community. It is my hope that this project serves as a source of inspiration and practical reference for aspiring developers seeking to explore the capabilities of the MERN stack and contribute to the evolving landscape of web development.

In essence, the Stack Overflow Clone not only marks the completion of a successful internship project but also symbolizes the beginning of a continued journey of learning, growth, and contribution to the dynamic world of software development.