HAPPY MESSAGING

I’m Beside You Inc. Intern Task

Harsh Verma,
IIT Kanpur

Table of Content

Happy Messaging

Welcome to the world of "Happy Messaging," a project aimed at revolutionizing your communication experience. In a fast-paced digital era, effective and joyful communication is more essential than ever. To achieve this, we've meticulously selected a powerful technology stack to build a messaging application that's not only efficient but also incredibly engaging.

Frontend: React - Our choice of React as the frontend framework ensures a highly interactive and user-friendly interface. React's component-based architecture allows us to create dynamic and responsive user experiences, making your messaging sessions more enjoyable and visually appealing.

Some alternatives to ReactJS are Angular, Vue.js, and Svelte. React excels with its component-based architecture, virtual DOM, and a vast ecosystem of libraries and tools. It offers a fine balance between performance and developer-friendly features, making it a popular choice for building dynamic and responsive web applications.

Backend: Node and Express - The backbone of our application relies on Node.js and Express. Node.js, known for its speed and scalability, perfectly suits our mission to deliver real-time messaging. Express, a popular Node.js framework, simplifies the creation of APIs and backends, ensuring the smooth operation of our messaging app.

Alternatives to Node.js for server-side JavaScript include Deno and WebAssembly. While Node.js is established and widely adopted, Deno offers enhanced security and TypeScript support. Express, a popular Node.js framework, has alternatives like Koa and Fastify, known for better performance and simplicity, making them attractive choices for specific use cases.

Database: MongoDB Atlas - Data management is a critical aspect of our project, and for this, we turn to MongoDB Atlas. This versatile data and application platform allows us to query data using SQL or NoSQL, depending on specific requirements. Additionally, MongoDB Atlas offers a built-in API, reducing the need for extensive backend code, which means more time can be dedicated to perfecting the user experience.

Alternatives to MongoDB Atlas include Amazon DocumentDB, Google Cloud Firestore, and Azure Cosmos DB. MongoDB Atlas stands out for its user-friendly interface, global scalability, and robust ecosystem. It provides flexible data modeling and rich query capabilities, making it an attractive option for developers looking for a versatile, cloud-based NoSQL database solution.

Real-time communication: Socket.io - To provide instant, bidirectional communication, we've implemented Socket.io. It enables real-time messaging, ensuring your messages are delivered promptly and your conversations are kept in sync, even across different devices.

Socket.io alternatives include SockJS and WebSocket. Socket.io is superior due to its broader browser compatibility, automatic reconnection, and built-in fallback mechanisms. It simplifies real-time application development, making it an excellent choice for developers who need cross-platform, reliable, and resilient real-time communication.

With this technology stack, we're not just building a messaging app; we're creating an immersive, interactive, and delightful platform for communication. "Happy Messaging" will offer you a unique and joyful way to connect with others, all while benefiting from the power and efficiency of these cutting-edge technologies. Get ready to experience messaging like never before – it's not just efficient; it's truly impactful.

What is Free Api?

The FreeAPI project is an innovative and community-driven initiative aimed at providing developers with free and accessible APIs for their projects.

The project focuses on delivering a wide range of APIs that cater to various domains and functionalities, enabling developers to seamlessly integrate these APIs into their applications.

Key highlights of the FreeAPI project include:

Accessibility: The FreeAPI project is committed to eliminating barriers by providing free access to its collection of APIs. Developers can leverage these APIs without any cost limitations, allowing them to experiment, learn, and build innovative applications.

Diverse API Collection: The project offers a diverse and comprehensive collection of APIs that span across different industries, domains, and functionalities. Whether you require social media integrations, payment gateways, machine learning algorithms, or IoT device connectivity, the FreeAPI project has you covered.

Simplified Integration: The FreeAPI project understands the challenges developers face when integrating APIs into their applications. To address this, the project provides clear documentation, code samples, and SDKs, simplifying the integration process and reducing development time and effort.

Community-Driven Development: The project fosters a vibrant and collaborative community of developers. Contributors are encouraged to share their knowledge, engage in discussions, and collaborate on API-related projects. This collective effort ensures the continuous improvement and reliability of the APIs offered by the FreeAPI project.

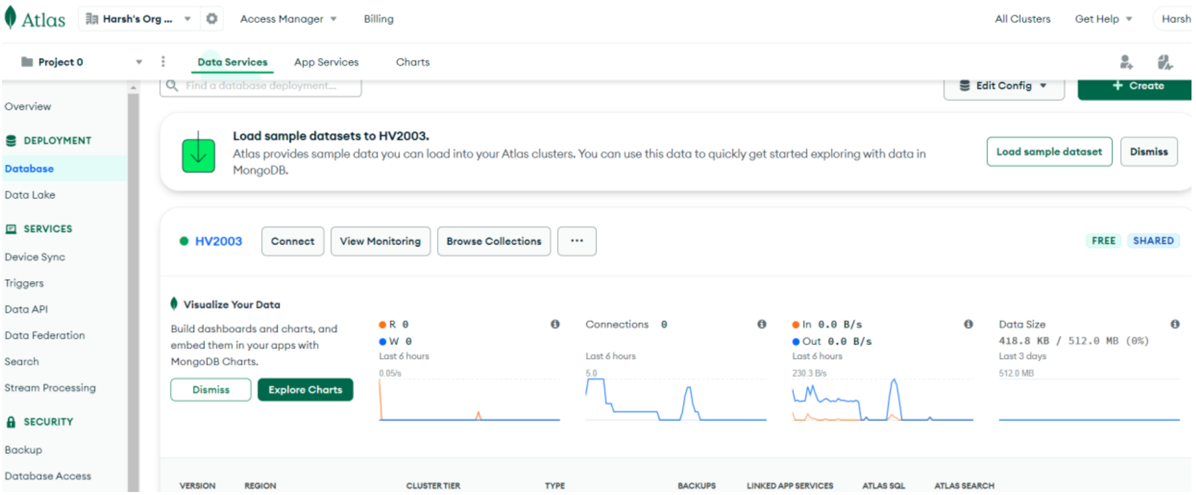
Learning and Skill Development: The FreeAPI project aims to empower developers by providing a platform for learning and skill development. Through access to various APIs and educational resources, developers can enhance their understanding of API integration, expand their knowledge, and showcase their expertise through building complete projects.

Guidelines for Establishing the Prototype

1) Clone the repository to a folder of your choice.

2) Generate a **'.env'** file in the primary directory and copy everything exactly as it appears in the **'.env.sample'** file.

3) Make a slight modification in the **'.env'** file, specifically on line 3 where you define the **MongoDB URL**. If MongoDB is already installed on your device, it should work smoothly, but I still recommend following these steps:
   - Sign up for **MongoDB Atlas**, a free cloud database service that will assist in storing chat data. In the security section, set your username and IP address. Avoid using special characters in your password to simplify the process.
   - After inputting the correct credentials and IP addresses, navigate to the left sidebar and access the database. It will appear as shown below. Click on the **'Connect'** button, followed by the **'Compass'** button.



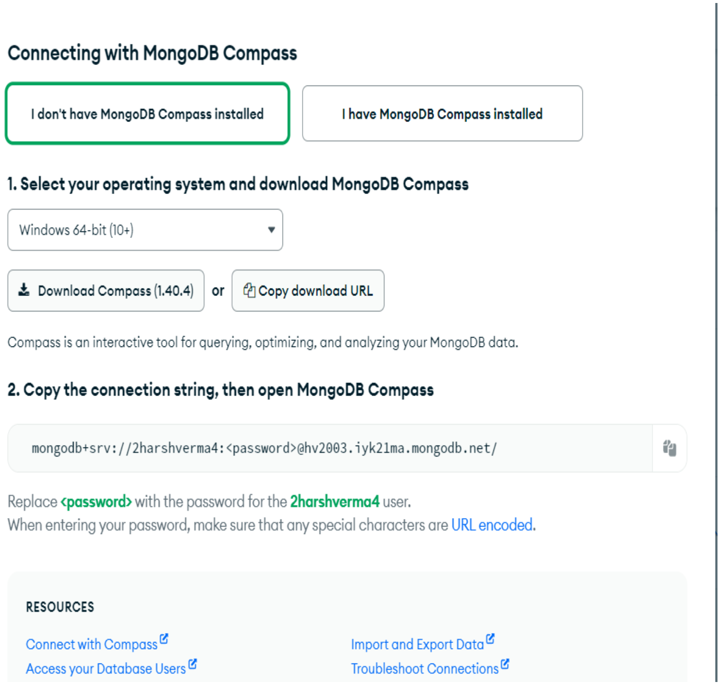
Load sample datasets to HV2003.

Atlas provides sample data you can load into your Atlas clusters. You can use this data to quickly get started exploring with data in MongoDB.

Build dashboards and charts, and embed them in your apps with MongoDB Charts.

- A page will pop up with a URL that includes the specified settings. It will resemble something like this: "**mongodb+srv://<username>:<password>@hv2003.iyk21ma.mongodb.net/**"
- Replace '<username>' and '<password>' with your respective username and password (without the '<>' symbols).

Note: Quiet a complex process but just follow the steps and you will get it.



Connecting with MongoDB Compass

I don't have MongoDB Compass installed
I have MongoDB Compass installed

1. Select your operating system and download MongoDB Compass

Windows 64-bit (10+)

Download Compass (1.40.4)
or
Copy download URL

Compass is an interactive tool for querying, optimizing, and analyzing your MongoDB data.

2. Copy the connection string, then open MongoDB Compass

```
mongodb+srv://2harshverma4:<password>@hv2003.iyk21ma.mongodb.net/
```

Replace <password> with the password for the 2harshverma4 user.
When entering your password, make sure that any special characters are URL encoded.

RESOURCES

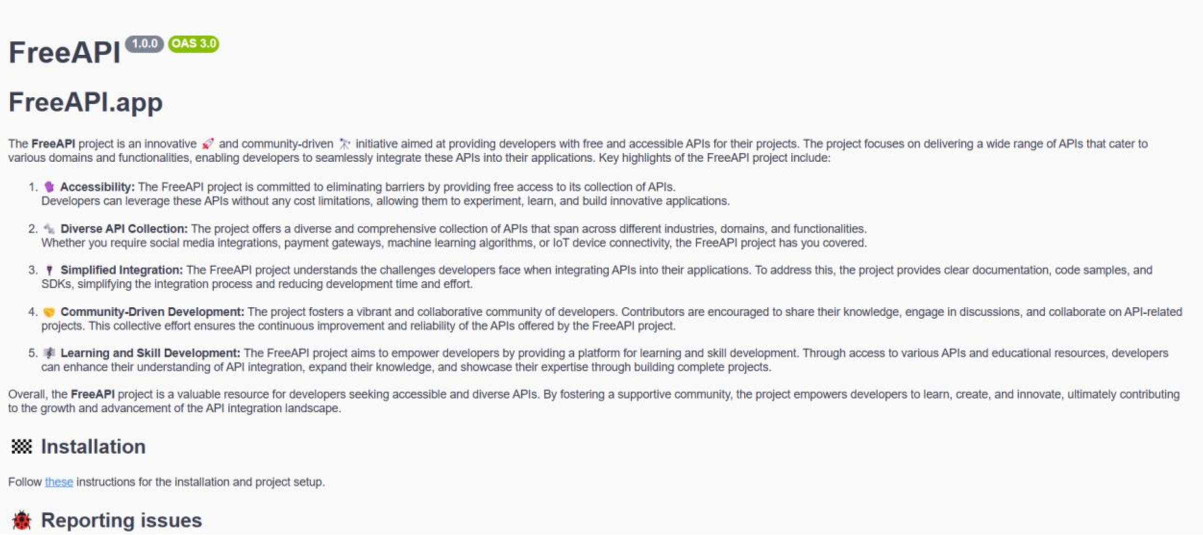
- After you've entered the MongoDB URL, your **'.env'** file is complete.

BACKEND:

4) Open a new terminal and confirm the presence of the **'package.json'** file by using the **'ls'** command.

5) Execute **'npm i'** to install all the necessary dependencies.

6) After all the dependencies have been successfully installed, start by clearing the terminal with the **'clear'** command. Then, initiate the backend using **'npm start'**. This will launch our backend, utilizing a FreeApi at http://localhost:8080, and it will open in your web browser, displaying the following:



FreeAPI 1.0.0 OAS 3.0

FreeAPI.app

The **FreeAPI** project is an innovative 🚀 and community-driven 🔭 initiative aimed at providing developers with free and accessible APIs for their projects. The project focuses on delivering a wide range of APIs that cater to various domains and functionalities, enabling developers to seamlessly integrate these APIs into their applications. Key highlights of the FreeAPI project include:

1. **Accessibility:** The FreeAPI project is committed to eliminating barriers by providing free access to its collection of APIs. Developers can leverage these APIs without any cost limitations, allowing them to experiment, learn, and build innovative applications.
2. **Diverse API Collection:** The project offers a diverse and comprehensive collection of APIs that span across different industries, domains, and functionalities. Whether you require social media integrations, payment gateways, machine learning algorithms, or IoT device connectivity, the FreeAPI project has you covered.
3. **Simplified Integration:** The FreeAPI project understands the challenges developers face when integrating APIs into their applications. To address this, the project provides clear documentation, code samples, and SDKs, simplifying the integration process and reducing development time and effort.
4. 🤝 **Community-Driven Development:** The project fosters a vibrant and collaborative community of developers. Contributors are encouraged to share their knowledge, engage in discussions, and collaborate on API-related projects. This collective effort ensures the continuous improvement and reliability of the APIs offered by the FreeAPI project.
5. **Learning and Skill Development:** The FreeAPI project aims to empower developers by providing a platform for learning and skill development. Through access to various APIs and educational resources, developers can enhance their understanding of API integration, expand their knowledge, and showcase their expertise through building complete projects.

Overall, the **FreeAPI** project is a valuable resource for developers seeking accessible and diverse APIs. By fostering a supportive community, the project empowers developers to learn, create, and innovate, ultimately contributing to the growth and advancement of the API integration landscape.

🏁 Installation

Follow these instructions for the installation and project setup.

🐞 Reporting issues

7) If you've reached this point, congratulations, we have completed the backend setup. To streamline our work, go to the **FreeAPI docs** documentation, click on 'Try it out,' and then 'Execute' to generate predefined users automatically.

8) The backend is now ready for use.

9) Keep the backend terminal running, and open a new terminal window, giving it a name of your choice to distinguish between your backend and frontend.

FRONTEND

10) In the frontend terminal, navigate to **'examples/chat-app/web/'** using the **‘cd examples/chat-app/web/’** command to access the required files.

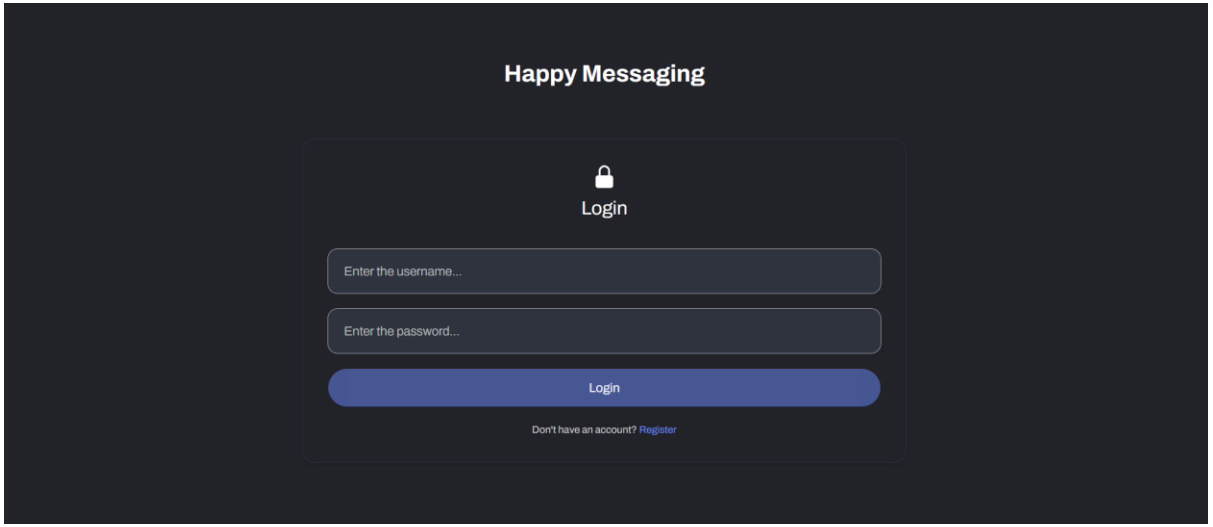
11) Perform a quick **'ls**' command to confirm the presence of the **'package.json'** file. Afterward, use **'npm i'** to install all the necessary dependencies.

12) Within the 'examples' folder, create a **'.env'** file. Ensure that it contains identical content to the **'examples/.env.sample'** file. It's important to double-check that you've correctly copied the content from 'examples/.env.sample' and pasted it into 'examples/.env' to prevent any confusion.

Important: The '.env' file in the main folder pertains to server-side configurations, while the '.env' file inside the 'examples' folder is specific to application settings and purposes.

13) Proceed to the terminal and execute **'npm run dev'**. This action will launch our frontend application at http://localhost:3000/. It will appear as shown in the image below:

Outlook:



Happy Messaging

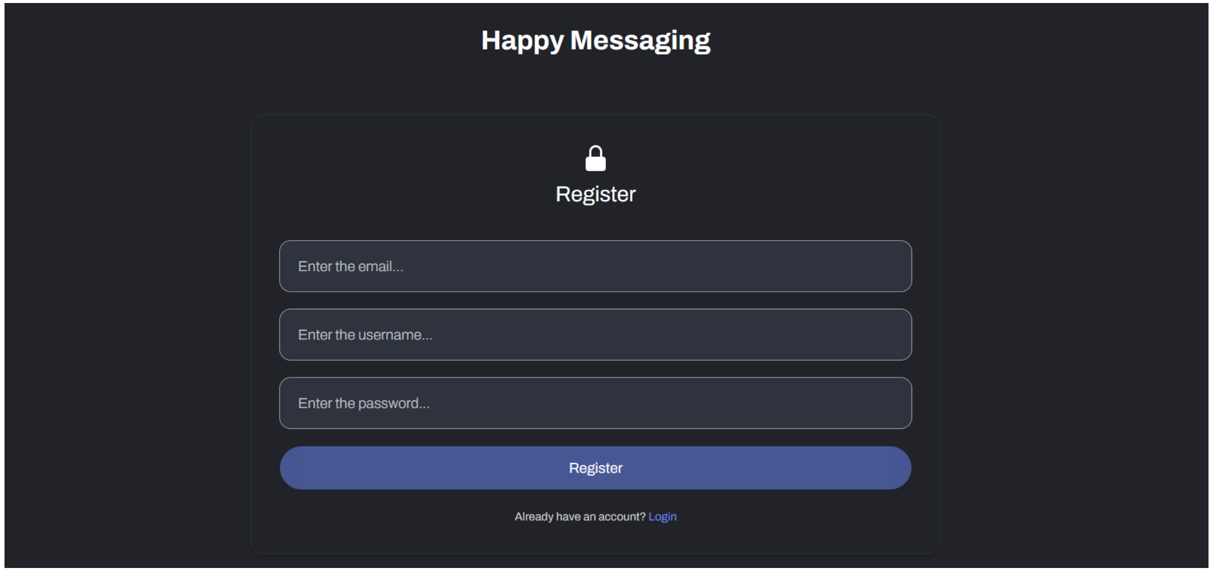
Login

Enter the username...

Enter the password...

Don't have an account? Register

14) Initially, let's start by registering a user. Click on the **'Register'** button, and then proceed to register a fictitious user of your preference. The window will appear as illustrated below:



Happy Messaging

Register

Enter the email...

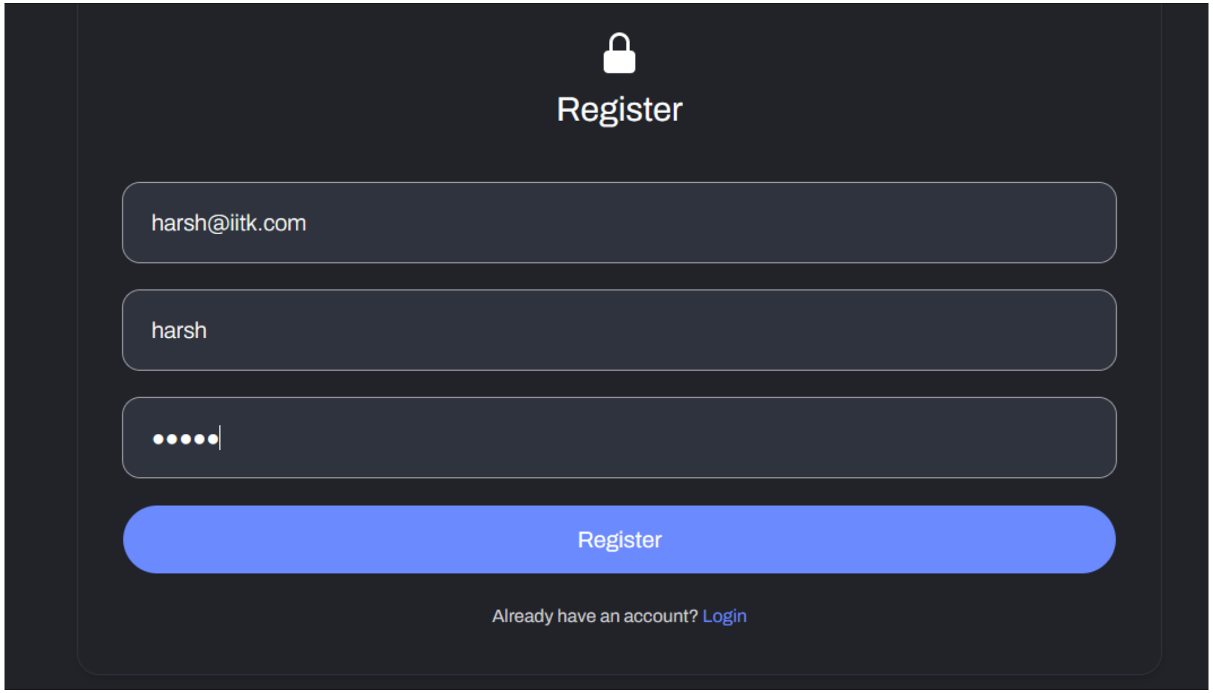
Enter the username...

Enter the password...

Register

Already have an account? Login

15) I used the following ids:

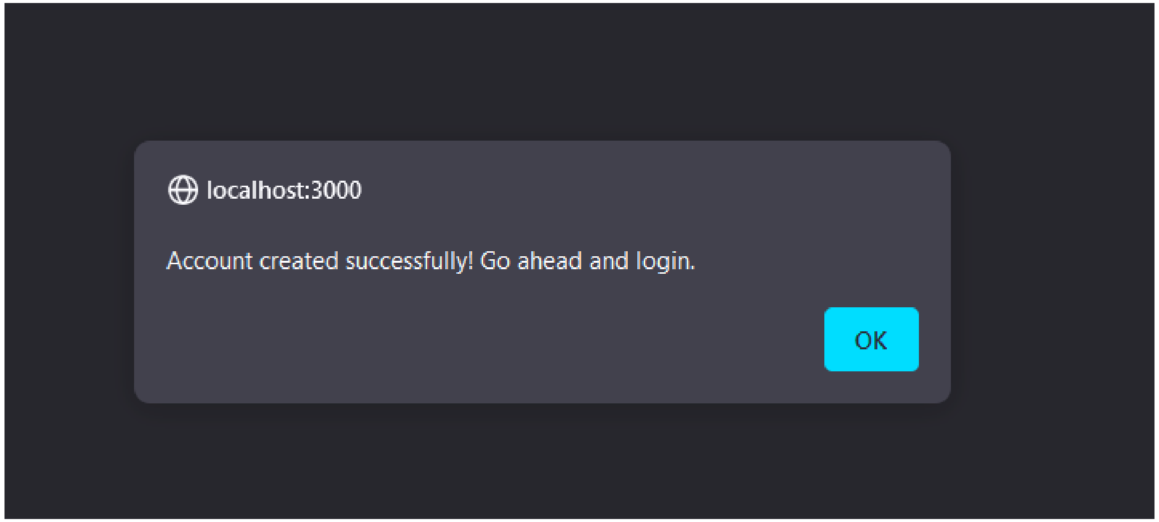


Register

harsh@iitk.com

harsh

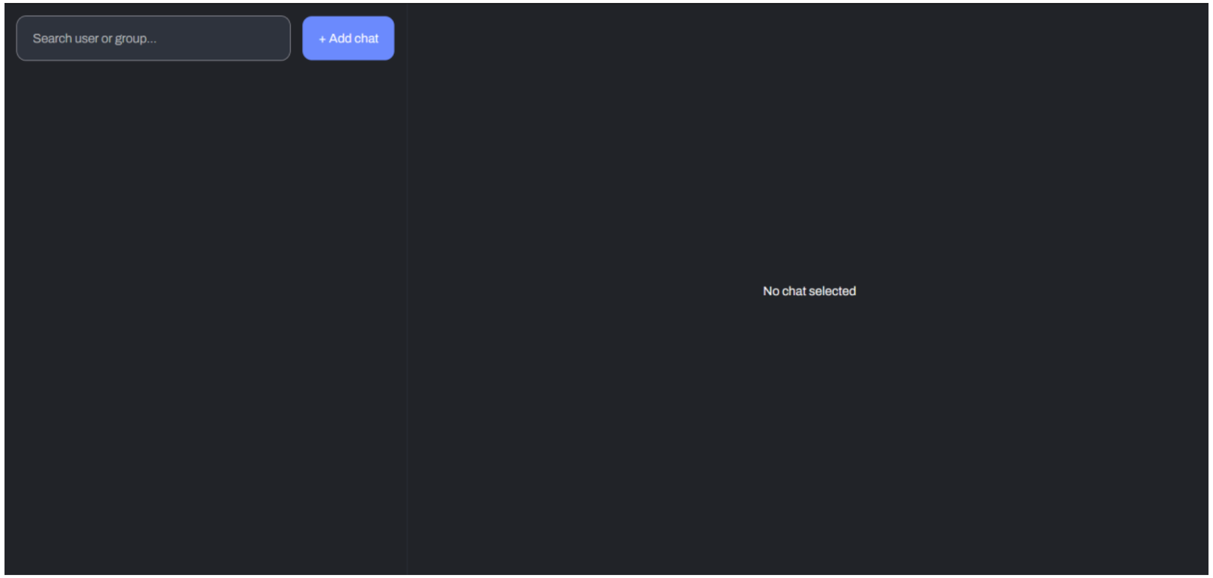
•••••



localhost:3000

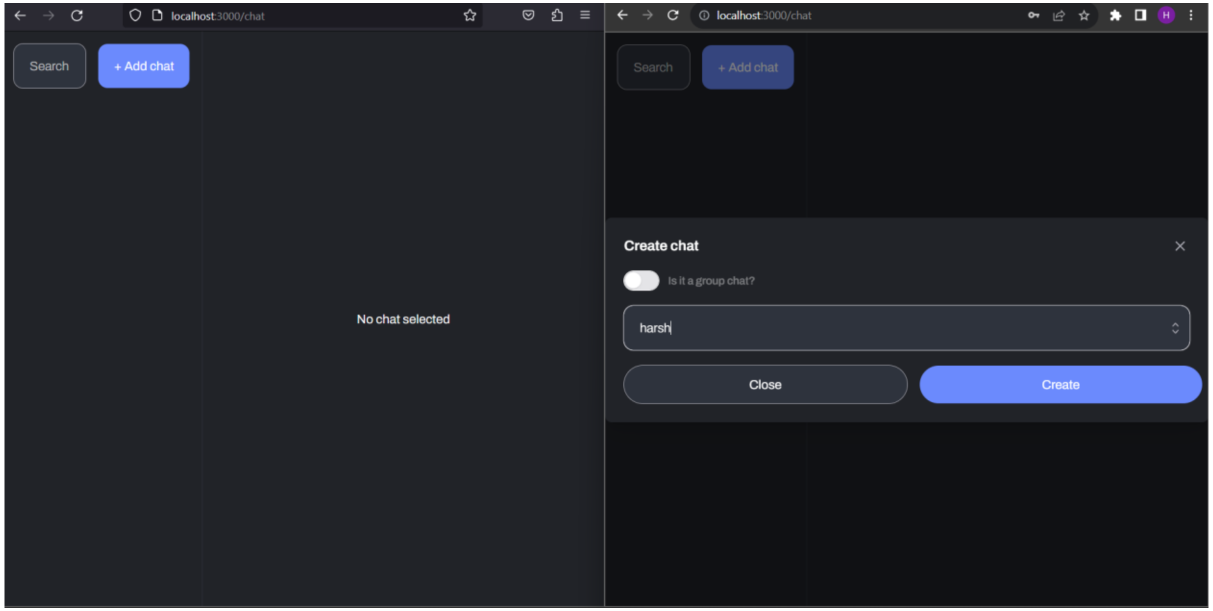
Account created successfully! Go ahead and login.

16) Click **'OK'**, and subsequently, log in using the same credentials. In the beginning, the interface will appear as follows:

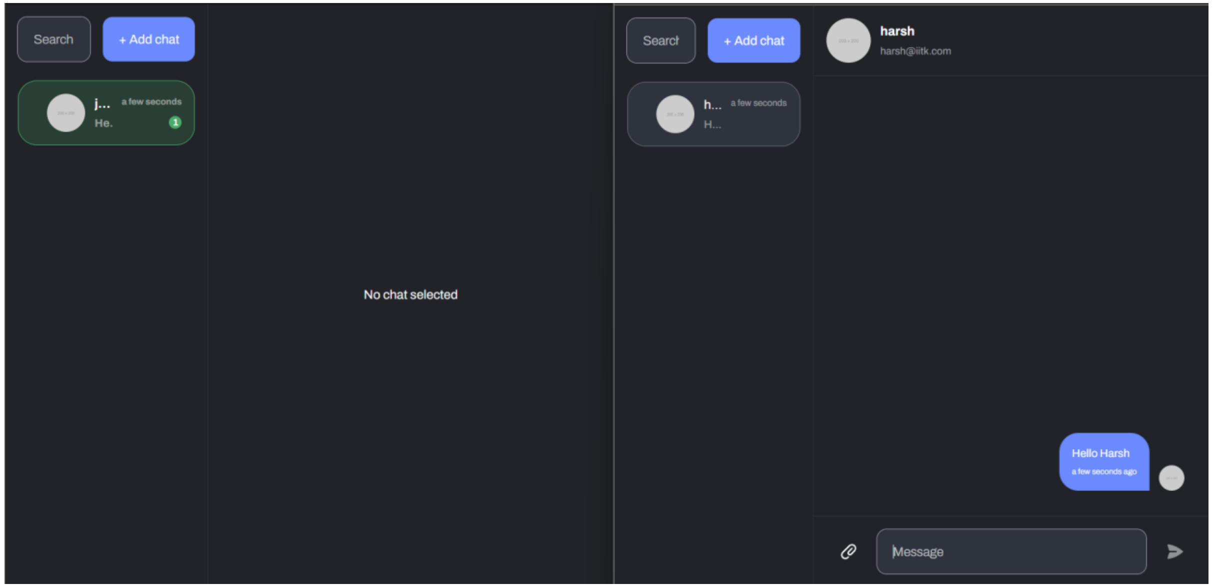


17) To enhance interactivity, open another web browser window (recommended: use incognito mode)

18) Click on **'+Add Chat'** to initiate a new chat. To observe real-time messaging using **socket.io**, it's advisable to create two users in different browsers or in incognito mode for interactive, real-time interaction, as shown.

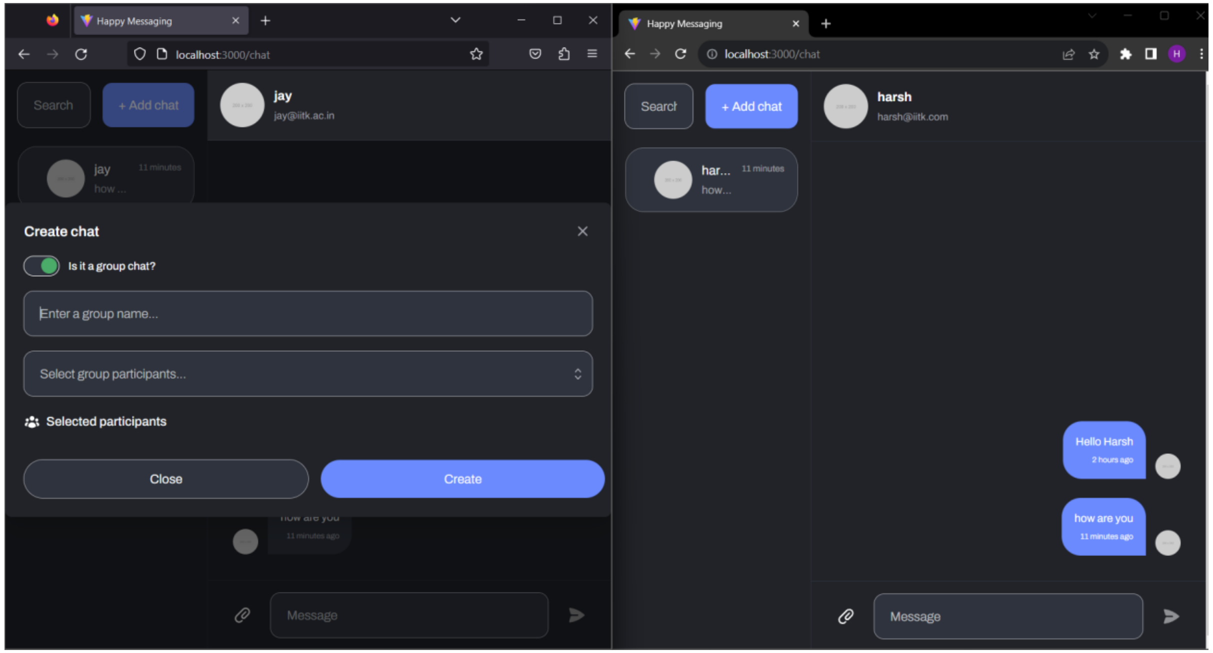


19) The 'Happy Messaging' chatbot facilitates real-time messaging, as demonstrated below. By opening two distinct user sessions, you can witness real-time messages in action.



Hello Harsh
a few seconds ago

20) Enabling the group chat feature is simple. Just click on '+Add chat' and slide the group chat bar.



Create chat

Is it a group chat? ☑

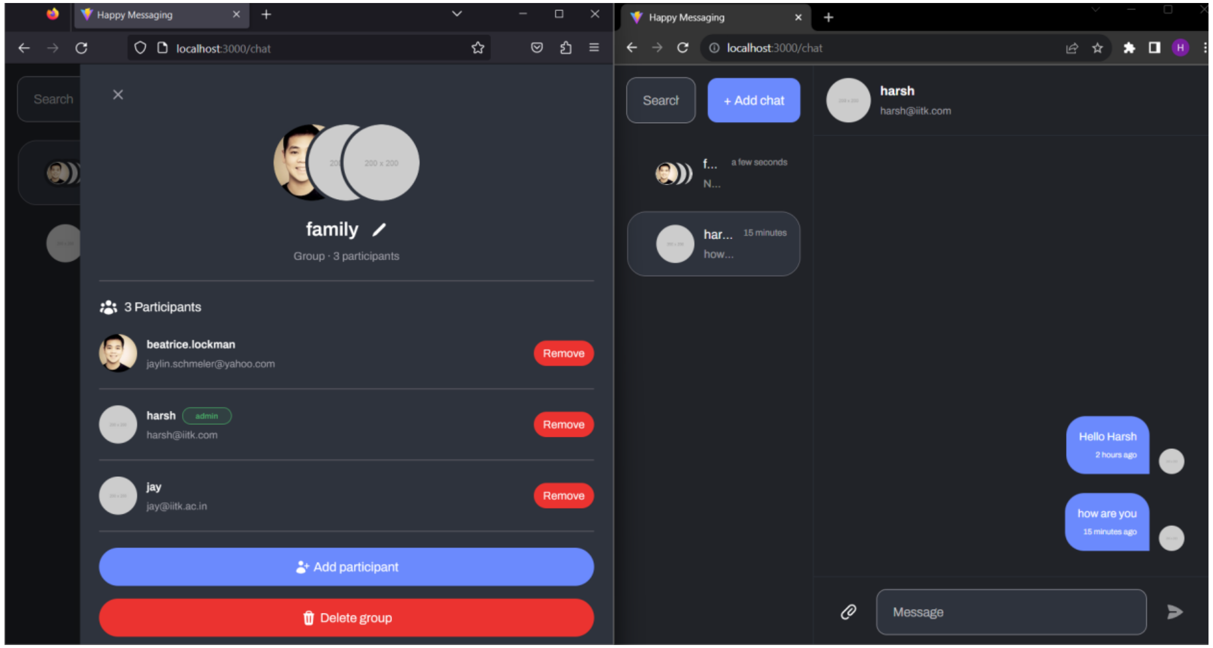
jay
jay@iitk.ac.in

harsh
harsh@iitk.com

Hello Harsh
2 hours ago

how are you
11 minutes ago

21) The group chat feature will appear as follows:



family

Group · 3 participants

3 Participants

beatrice.lockman jaylin.schmeler@yahoo.com

harsh admin harsh@iitk.com

jay jay@iitk.ac.in

harsh harsh@iitk.com

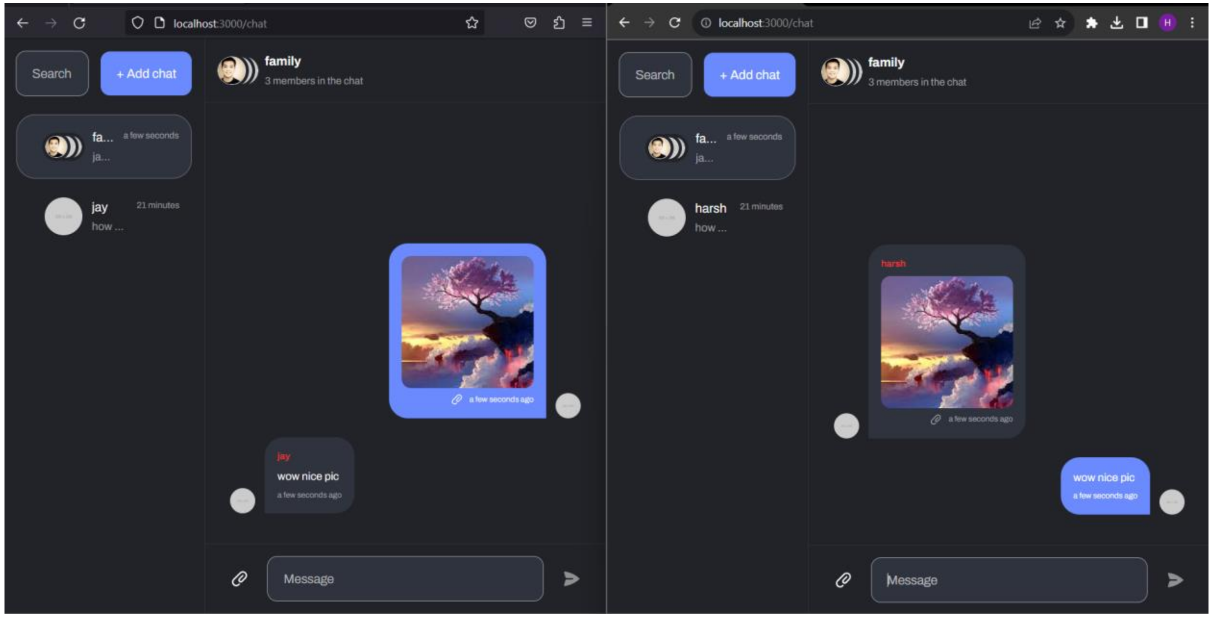
f... a few seconds N...

har... 15 minutes how...

Hello Harsh 2 hours ago

how are you 15 minutes ago

22) The 'Happy Messaging' prototype also allows us to send photos.



a few seconds ago

jay

wow nice pic

a few seconds ago

harsh

a few seconds ago

wow nice pic

a few seconds ago