**Mini Project Report on**



**React Chat Application**



**Submitted in partial fulfillment of the requirement for the award of the degree of**

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE & ENGINEERING**

**Submitted by:**

**Student Name**  **University Roll No.**

**Arpit Gupta 2016666**

***Under the Mentorship of***

**Ms Tanusha Mittal**



**Department of Computer Science and Engineering**

**Graphic Era (Deemed to be University)**

**Dehradun, Uttarakhand**

**January 2023**



**CANDIDATE’S DECLARATION**

I hereby certify that the work which is being presented in the project report entitled **“React chat Application”** in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineeringof the Graphic Era (Deemed to be University), Dehradun shall be carried out by the under the mentorship of **Ms Tanusha Mittal**, Department of Computer Science and Engineering, Graphic Era (Deemed to be University), Dehradun.

Arpit Gupta 2016666 **signature**

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **Chapter No.** | **Description** | **Page No.** |
| Chapter 1 | Introduction |  |
| Chapter 2 | Literature Survey |  |
| Chapter 3 | Methodology |  |
| Chapter 4 | Result and Discussion |  |
| Chapter 5 | Conclusion and Future Work |  |
|  |  |  |

**Chapter 1**

**Introduction**

In the following sections, a brief introduction and the problem statement for the work has been included.

* 1. **Introduction**

Email, newsgroup and messaging applications provide means for communication among people but these are one-way mechanisms and they do not provide an easy way to carry on a real-time conversation or discussion with multiple people involved. This project is to create a desktop chat application to enable the users to chat with each other. To develop an instant messaging solution to enable users to seamlessly communicate with each other

**Motivation**

The motivation behind choosing this project was my keen interest to learn React Js and explore more in the field of web development. React Js is one of the finest javascript frameworks used to design single-page web applications. We all use chat apps to communicate with the people around us due to the capability of the chat application to provide real-time communication among multiple people. As a result, users enjoy and hardly think of using anything else to connect informally.

**Chapter 2**

**Literature Survey**

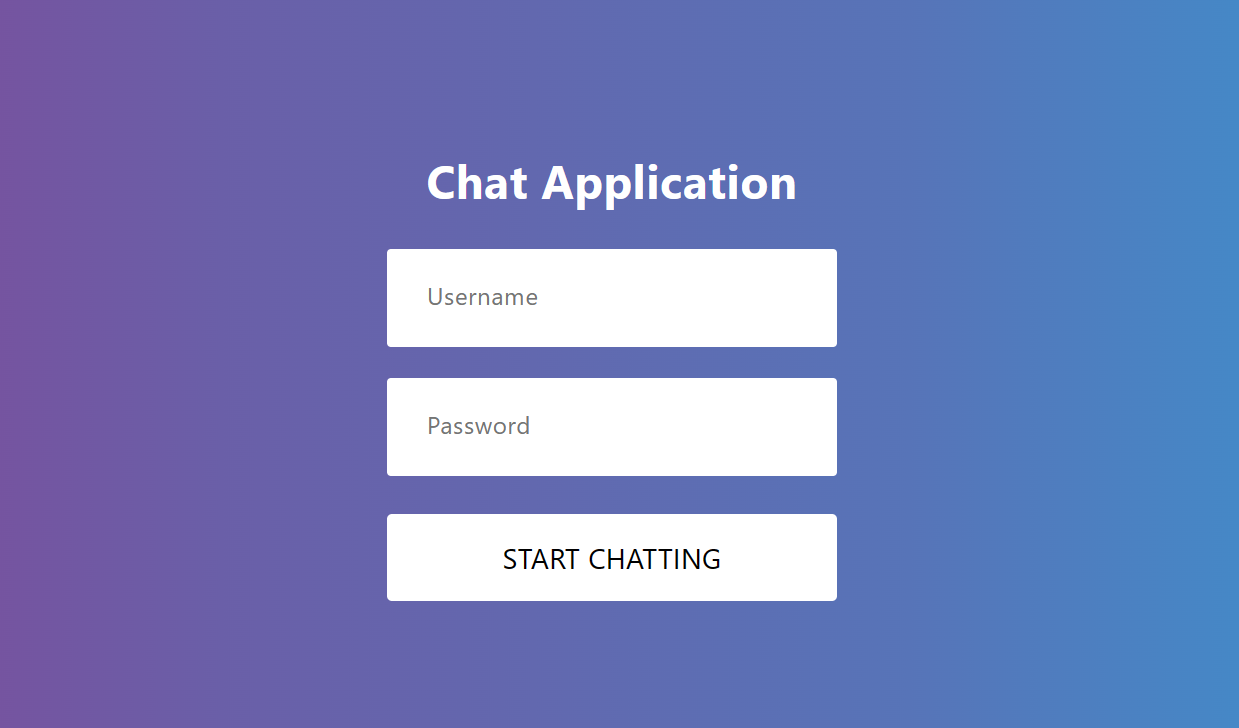
Channels are the javascript project which was created to extend the ability of React.js to the next level. We were working in standard React.js which did not support asynchronous and channels and connection via Firebase.io to create real-time applications. Channels extend the ability of React.js beyond HTTP and make it work with Firebase.io, chat protocols, IoT protocols, and more. It is built on ASGI support which stands for Asynchronous Server Gateway Interface. ASGI is the successor of WSGI which provides an interface between async and React.js. Channels provide the functionality of ASGI, by extending WSGI to it, and it provides ASGI support with WSGI. Channels also bundle the event-driven architecture with the channel layers, a system that allows you to easily communicate between processes and separate your project into different processes.

**Chapter 3**

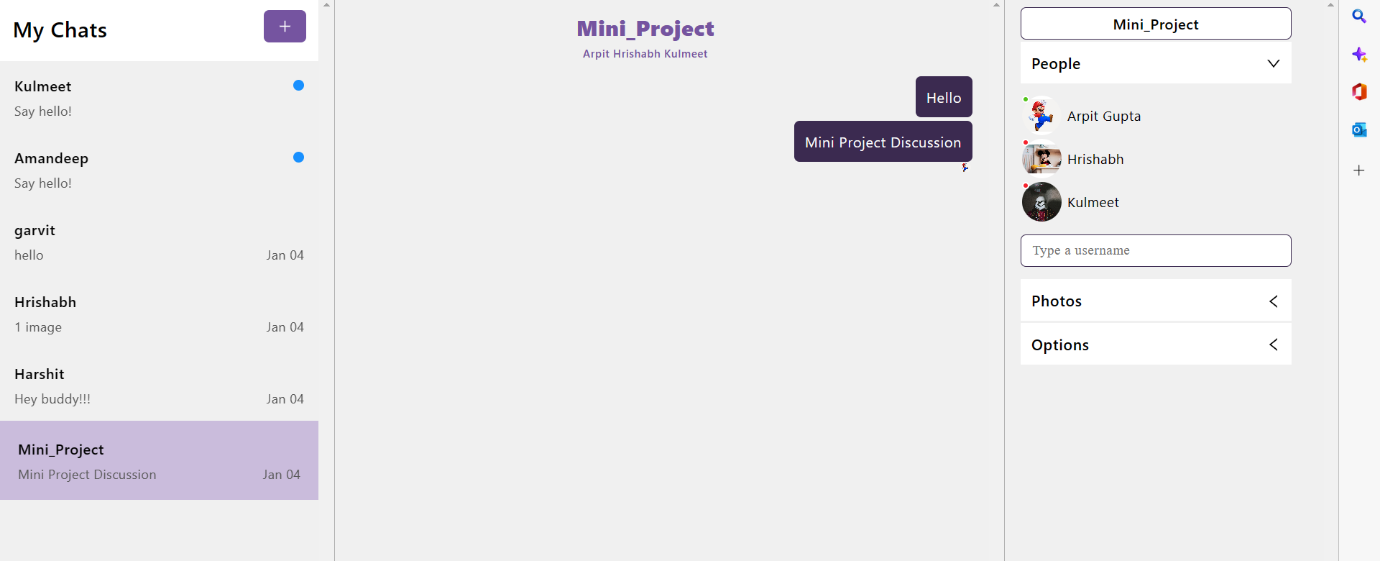
**Methodology**

Technology And Tools Used

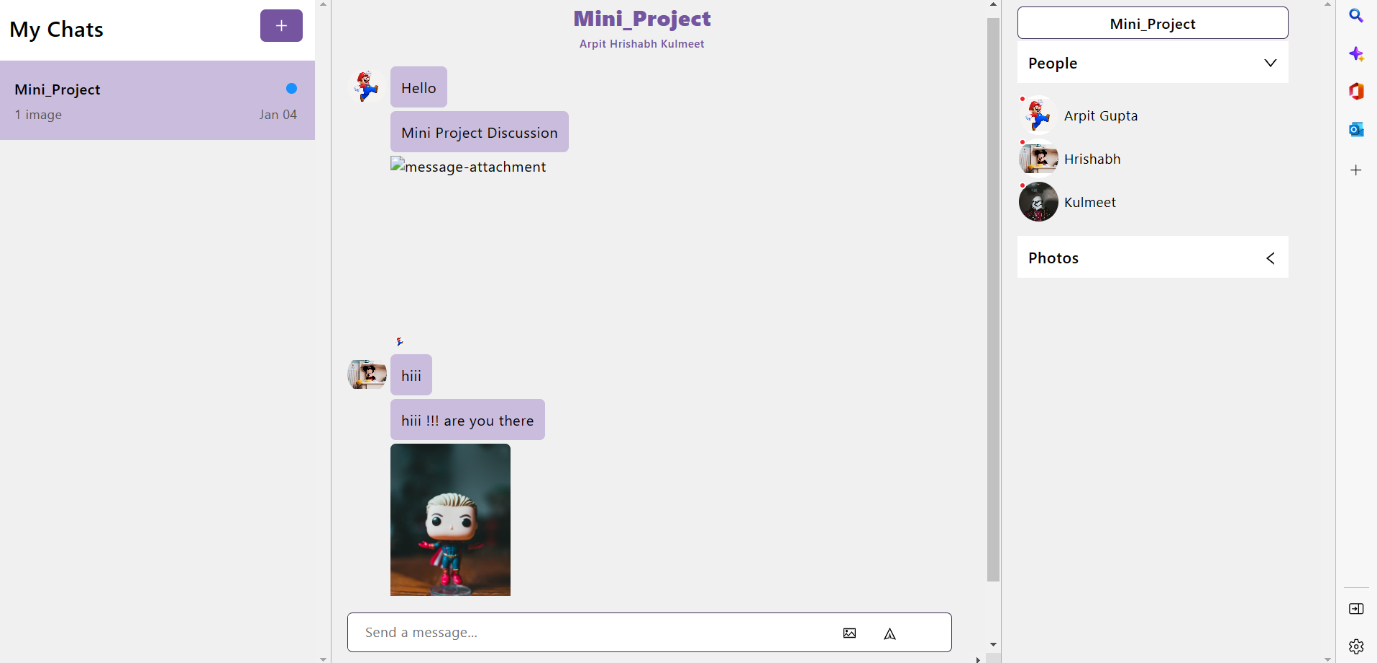
STEP 1 Users land on the login page of Desktop Chat Application LetsChat Work Flow of the Project



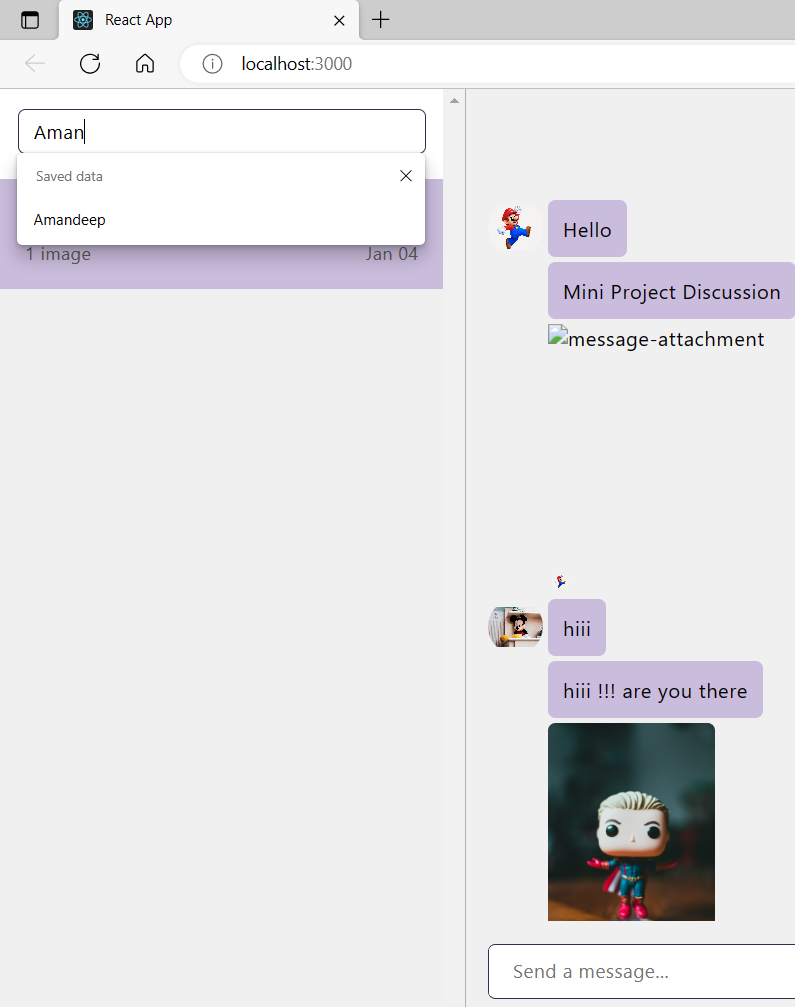
STEP 2 Users land on the main screen of the web application.



STEP 3 Users in a group sending messages,images.



STEP 4 Users can create a new room by clicking on the Add New Chat button on the top to start chatting with their friends.



**Technology and Tools Used:**

### **Prerequisites:**

* Basic knowledge of JavaScript.
* Basic knowledge of [React.js](https://reactjs.org/tutorial/tutorial.html) library.
* [Node.js](https://nodejs.org/en/download/) installed on your local development environment.
* You should be familiar with [Node Package Manager](https://docs.npmjs.com/cli/v6/commands/npm).

**React Js Library-**

[Create React App](https://github.com/facebookincubator/create-react-app) is a comfortable environment for **learning React**, and is the best way to start building **a new**[single-page](https://reactjs.org/docs/glossary.html#single-page-application)**application** in React. It sets up your development environment so that you can use the latest JavaScript features, provides a nice developer experience, and optimizes your app for production. You’ll need to have [Node >= 14.0.0 and npm >= 5.6](https://nodejs.org/en/) on your machine. To create a project, run:

**Npx create-react-app Chat-Application**

**Cd Chat-Application**

**Npm start**

### **What chat engine API is?**

A chat engine is an API that makes it very simple to develop chat services.

The chat engine makes it simple to develop chat services which would rather take time building from scratch.

Using services like chat engines has a greater advantage as compared to systems build from scratch since they provide an easy to use environment.

For instance, it provides an access to REST API for chat hosting.

The chat engine is so powerful that it meets all your needs when it comes to chat UI components development and storage.

### **Creating the application**

Not that we have basic knowledge in chat engines, let’s proceed and set up our application as follows:

Run the following command on your terminal to create a chat application project.

npx create-react-app chat-application-example

Upon installation, cd into the project root and install the chat engine as shown below:

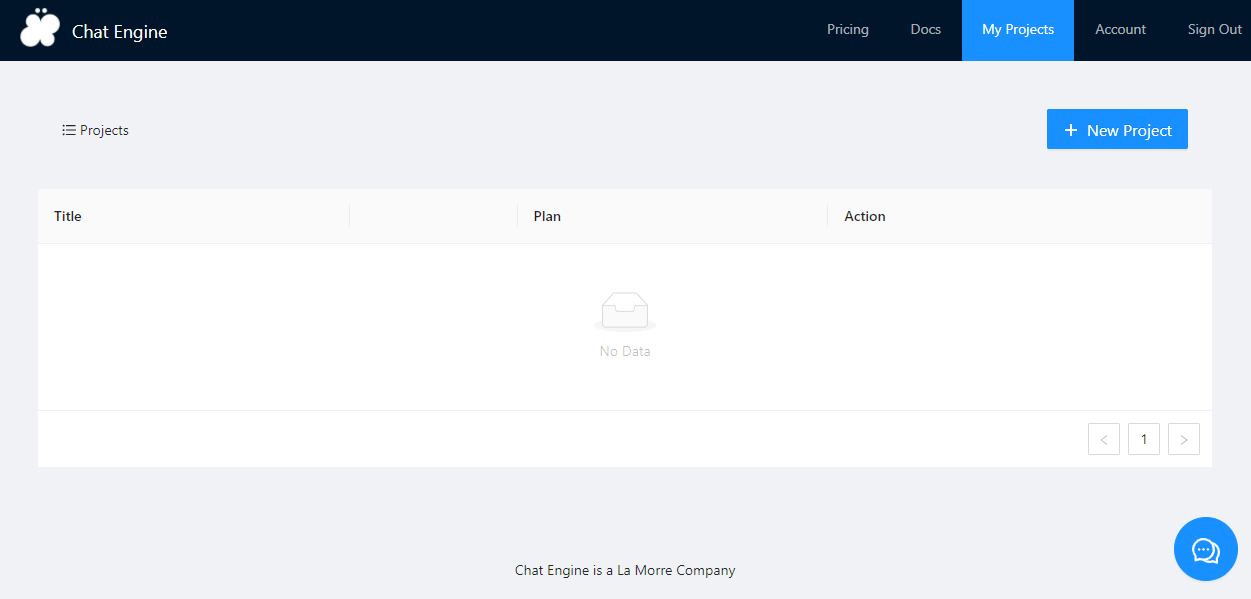
cd chat-application-example

npm i react-chat-engine

#### **Setting up a chat engine Project**

In this section, we are setting up the chat engine application using the steps described below:

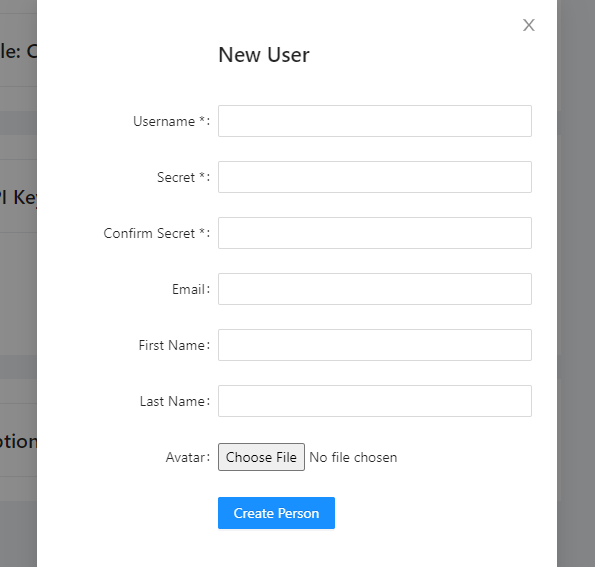
* You are required to have the API keys from your [chatengine.io](https://chatengine.io/) account.
* Register/login to your account.
* You will be redirected to a page like below:



* Click the New Project button you see on the page, and give your project a title.

Save the project ID and the secret keys somewhere safe. We will use them along the way.

* Still, on this page, click the “new user” section at the top right corner and give a name to your new user in a modal that pops up.



#### **Adding the ChatEngine component**

In the React app created, delete every unnecessary file such as **logo.SVG, reportWebVital.js, setupTest.js, and App.test.js**.

Remember to delete the reportWebVital import and its called function in the index.js file.

Setting up the chat engine in our application will be done as shown below:

import React from 'react'

import {ChatEngine} from 'react-chat-engine';

//Using arrow function component

const App = () => {

return (

<ChatEngine

userName=''//Put your userName instead

projectID = ''// Your project id goes here

userSecret=''// Replace with your secret key

/>

)

}

export default App

Properties passed to the component are:

1. **user Name** - That you created in your project at [chat engine](https://www.section.io/engineering-education/creating-a-simple-chat-app-in-react-using-the-chat-engine-api/chatengine.io).
2. **project ID** - This is created for every project you add on the dashboard.
3. **secret key** - We had secretly stored before and the username we came up with for the new user.

This component sets up the chat engine or in layman’s terms, backbone for any chat application.

**Chapter 4**

**Result and Discussion**

Chat refers to the process of communicating, interacting

and/or exchanging messages over the Internet. It involves two

or more individuals that communicate through a chat-enabled

service or software. Chat may be delivered through text, audio

or video communication via the Internet.

A chat application has basic two components, viz server and

client. A server is a computer program or a device that

provides functionality for other programs or devices. Clients

who want to chat with each other connect to the server.

The chat application we are going to make will be more like a

chat room, rather than a peer to peer chat. So this means that

multiple users can connect to the chat server and send their

messages. Every message is broadcasted to every connected

chat user.

**Chapter 5**

**Conclusion and Future Work**

This website provides a real-time chatting experience to users. But still, there's a lot of scope for

improvement here.

1. We can add a feature that will allow users to share images and videos in real-time.

2. We can add a feature that will allow users to share different types of files like docx, xlxs, pdfs, mp3,

etc. in real-time.

2. We can add notifications to this web app so that users get notified of new messages received.

3. We can allow users to manually change the display picture of their profile.

4. We can add end-to-end encryption to make this app more secure and the privacy of users is also

Protected.