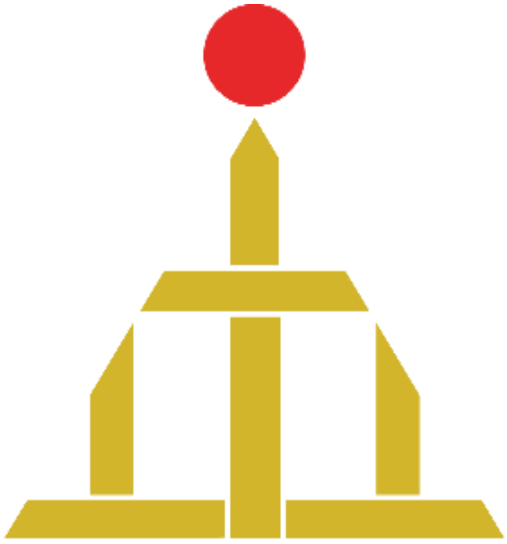
PROJECT REPORT

CHAT APPLICATION

USING

FLASK, PYTHON, FLASK-SOCKET IO & MongoDB



भारतीय सूचना प्रौद्योगिकी संस्थान भागलपुर
Indian Institute of Information Technology
Bhagalpur

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Indian Institute of Information Technology

Bhagalpur, Bihar – 813210

Analysis and Implementation of Chat Application

Project submitted in

Dec 2020

to the department of

Computer Science and Engineering of

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY BHAGALPUR

in partial fulfilment of the requirements for the degree of

Bachelor of Technology in Computer Science and Engineering

By

ABHISHEK KUMAR (170101003) CSE

HIMANSHU RANJAN (170101017) CSE

TABLE OF CONTENT

1. INTRODUCTION

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.

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.

1.1 DOCUMENT SCOPE AND PURPOSE

This document provides a description of the technical design for Chat-Application **U CHAT**. This document provides an architectural overview of the system to depict different aspects of the system. This document also functions as a foundational reference point for developers.

1.2 TARGET AUDIENCE

This document is targeted (but not limited) to technical stakeholders:

- Development Team
- Clients
- Respective Authority

It is assumed that the reader has a technical background in software design and development.

2. REQUIREMENT ANALYSIS

2.1 SYSTEM REQUIREMENT

- Development: Python + Flask
- Unit Test: unit test(localhost)
- Database Management: Mongo DB
- Database: Document-Oriented NoSQL
- Server: Mongo Client
- Discussion: Normal Group Discussion

2.2 TOOLS REQUIREMENTS

2.2.1 Python

It's often used as a "scripting language" for web applications. This means that it can automate specific series of tasks, making it more efficient. Consequently, Python (and languages like it) is often used in software applications, pages within a web browser, the shells of operating systems and some games.

2.2.2 HTML

(Hypertext markup language) is the set of markup symbols or codes inserted in a file intended for display on a World Wide Web browser page.

2.2.3 FLASK

Flask is a web framework. This means **flask** provides you with tools, libraries and technologies that allow you to build a web application. This web application can be some web pages, a blog, a wiki or go as big as a web-based calendar application or a commercial website.

2.2.4 Mongo DB

MongoDB is a document-oriented NoSQL database **used** for high volume data storage. Instead of using tables and rows as in the traditional relational databases, **MongoDB** makes **use** of collections and documents. Documents consist of key-value pairs which are the basic unit of data in **MongoDB**.

3. DESIGN

3.1 DESIGN APPROACH

The design approach used here is based on the following:

I. DATA FLOW DESIGN: The data flow of the Chat Application U CHAT is Internet-based.

II. ARCHITECHTURE DESIGN: The Chat Application is developed with the help of Python, bson, werkzeug, flask, flask-socketio, eventlet, flask-login. The database used is NoSQL, and the web application is deployed through mongo client (pymongo) Server.

III. UI DESIGN: The Internal Complaint Management System uses HTML, CSS at the Front End. It has been made such that the web application is very user friendly and all the functions present will make the life of user easier.

3.2 DESIGN PATTERNS

This application is designed as an object-oriented system for an Internet-based architecture by factoring application classes into the following layers:

I. The Presentation layer: This is the layer where the physical window and widget objects live. Any new user interface widgets developed for this application are put in this layer.

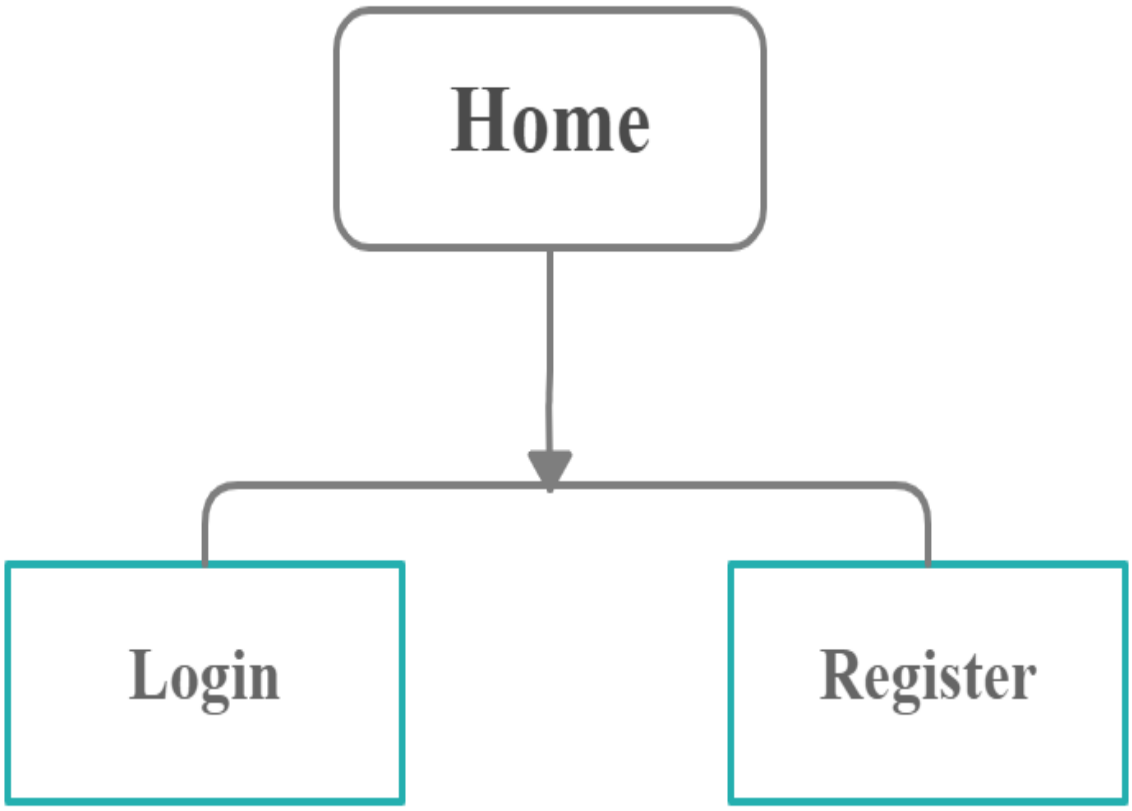
II. THE DOMAIN MODE: Most objects identified in the OO analysis and design will reside. To a great extent, the objects in this layer can be application-independent. Generic objects may be used in this application to reap the benefits of Object-Oriented programming.

III. THE DATA LAYER: The data is managed by NoSQL.

3.3 FLOW CHARTS

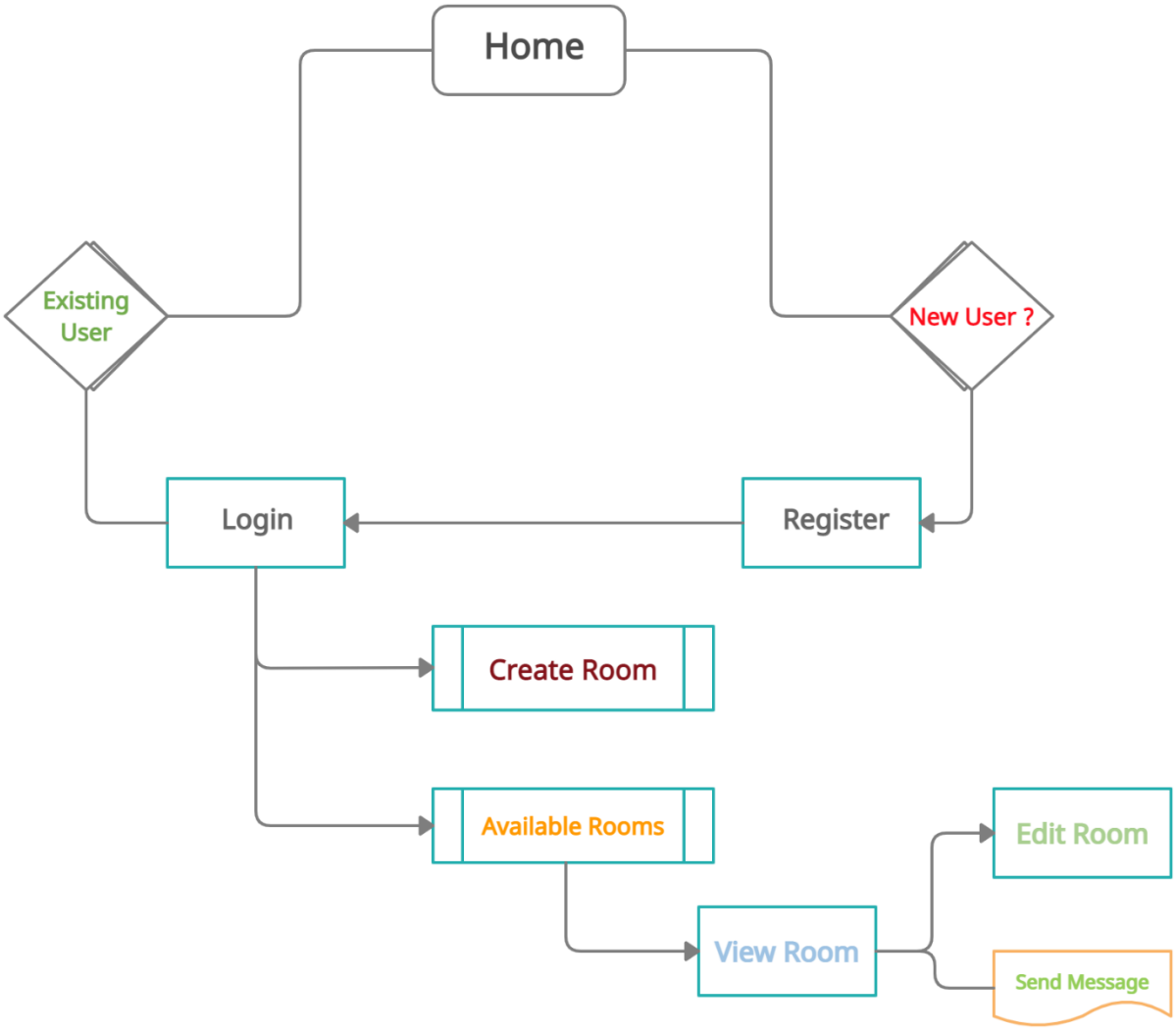
3.2.1 U CHAT Home

Chat Application U CHAT



3.2.2 User Login/Registrations and User Chatting Space

Chat Application U CHAT



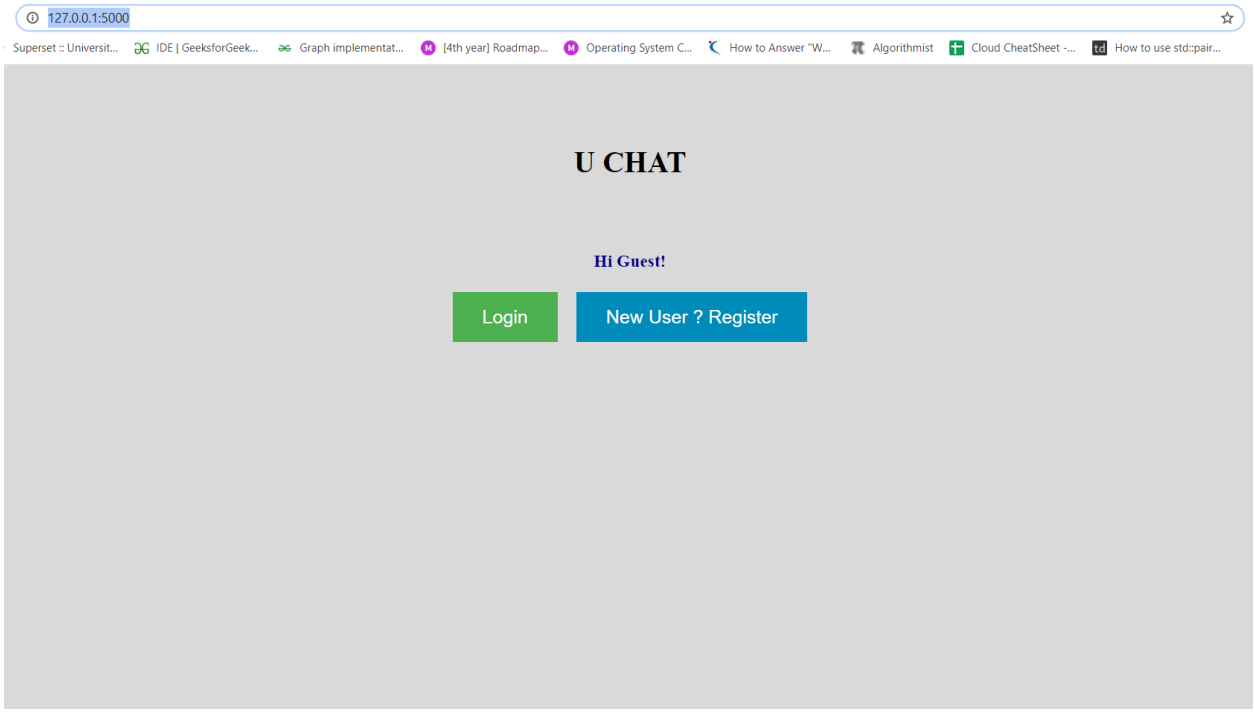
4. IMPLEMENATION

4.1 DATA DICTIONARY

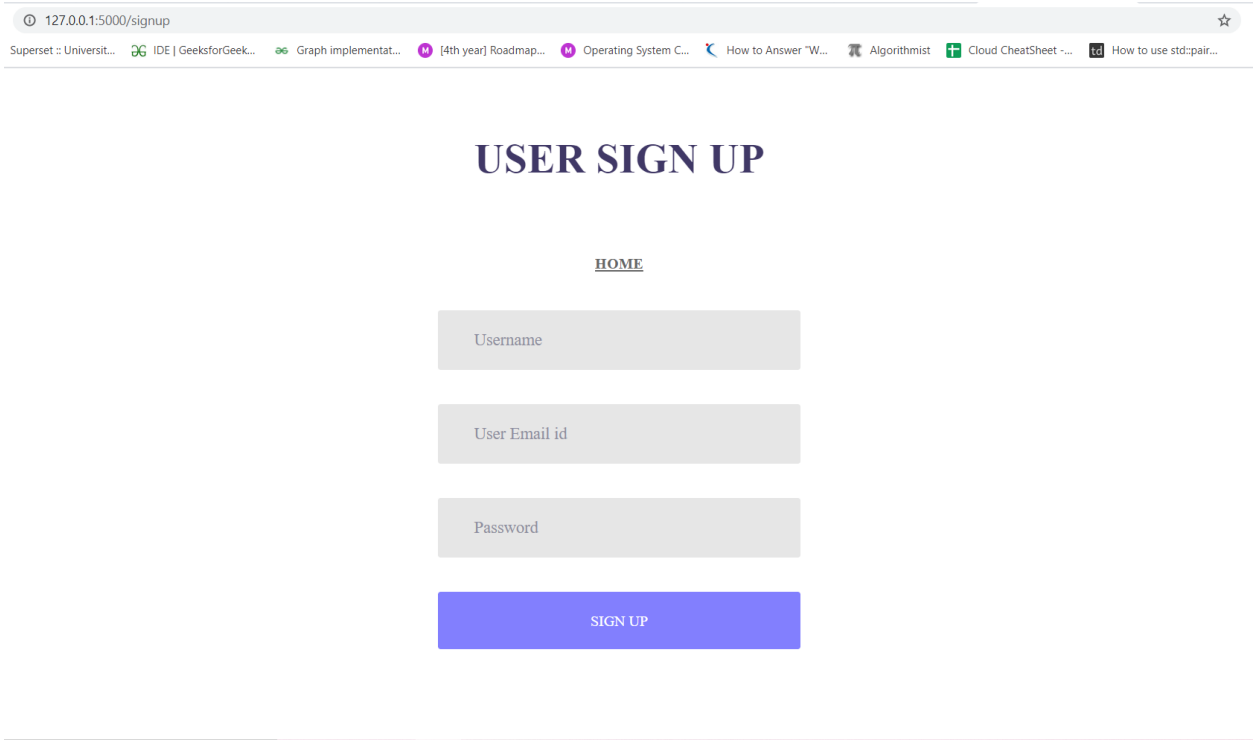
users	**_id**	Varchar (225)
	email	Varchar (255)
	password	Varchar (225)
message	**_id**	Int (11)
	room_id	Varchar (225)
	text	Varchar (225)
	sender	Varchar (225)
	created_at	timestamp
room_members	**_id**	Int (11)
	room_name	Varchar (225)
	added_by	Varchar (225)
	added_at	timestamp
	is_room_admin	text
rooms	**_id**	Int (11)
	name	Varchar (225)
	created_by	Varchar (225)
	created_at	Timestamp

4.2 SNAPSHOTS

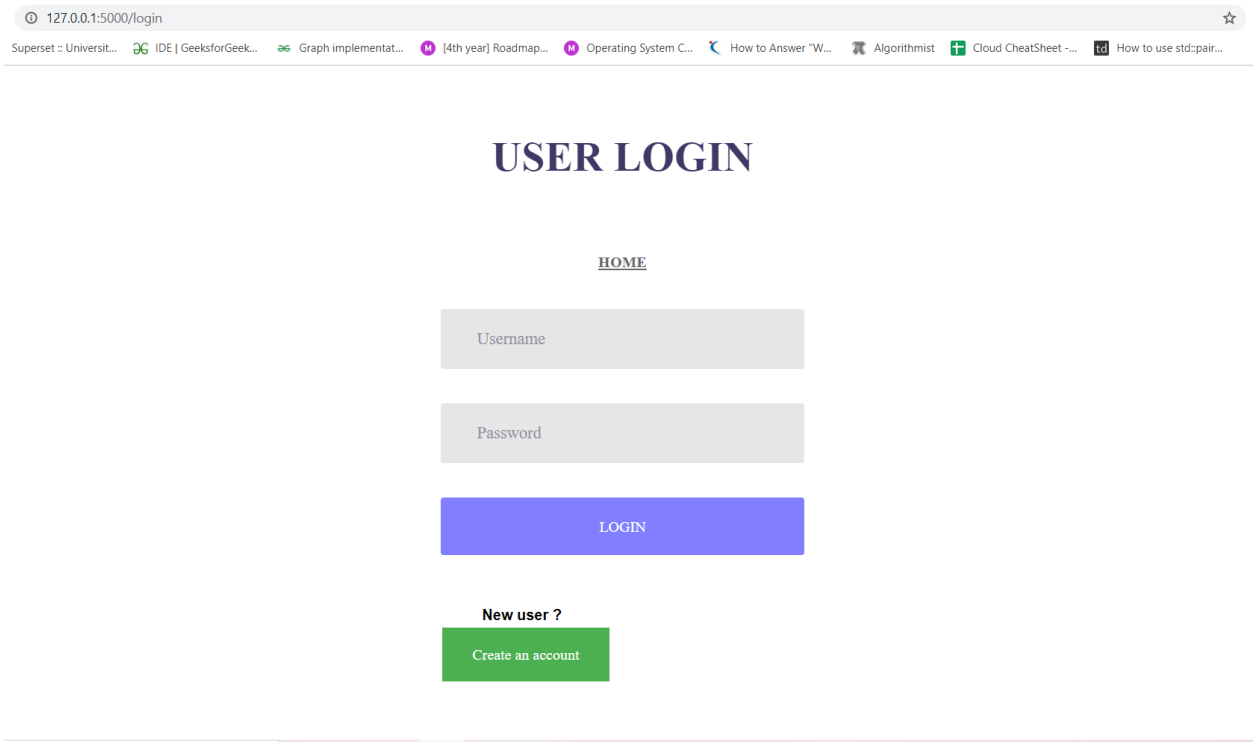
4.2.1 Home



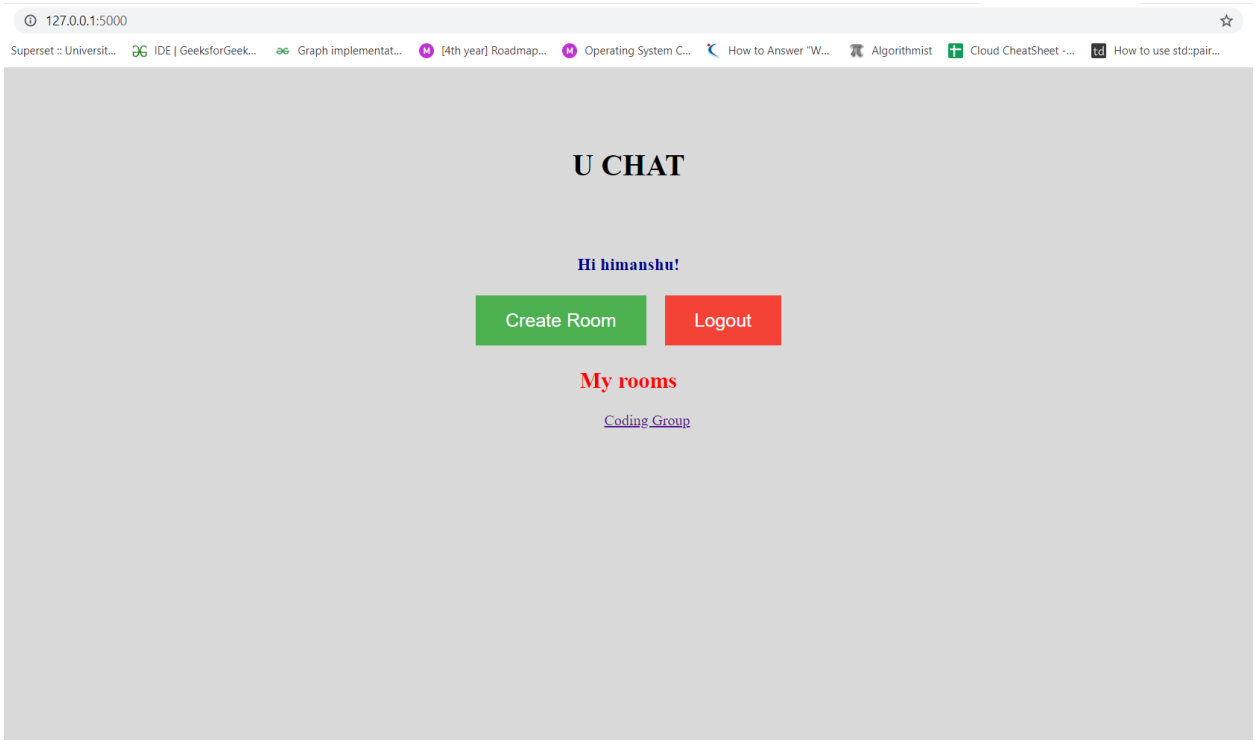
4.2.2 Register



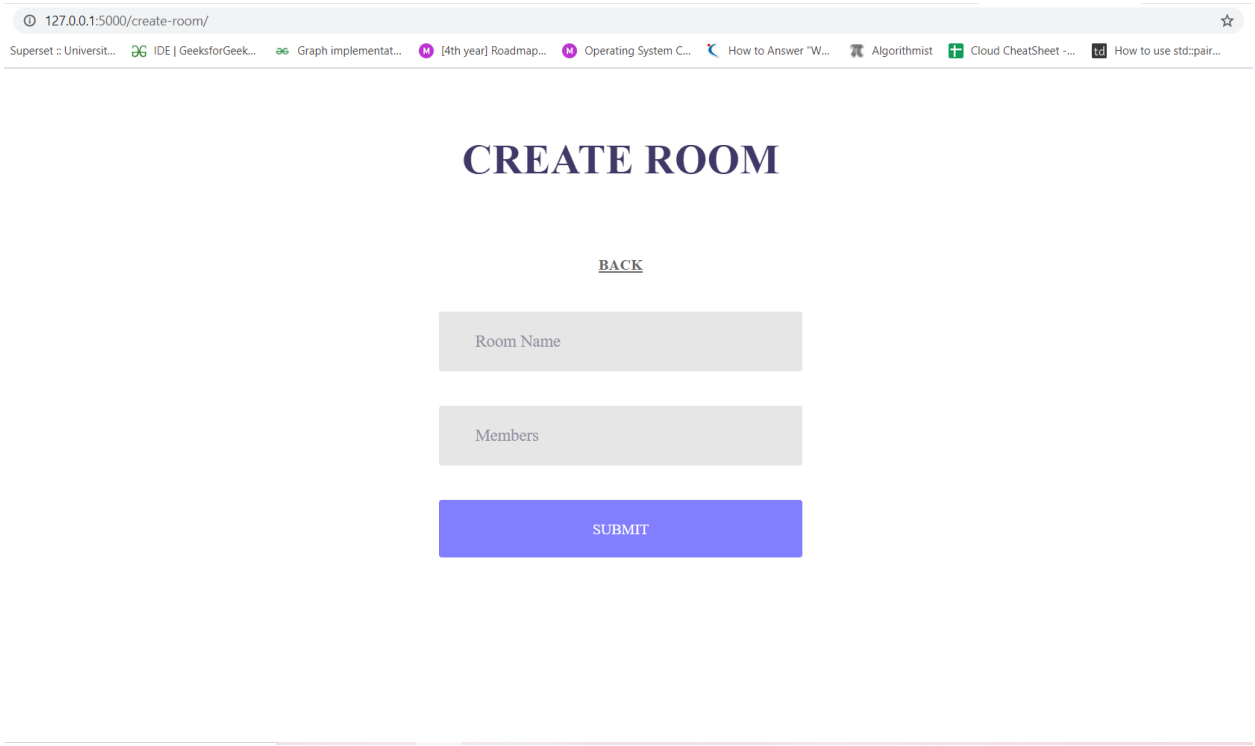
4.2.3 Login



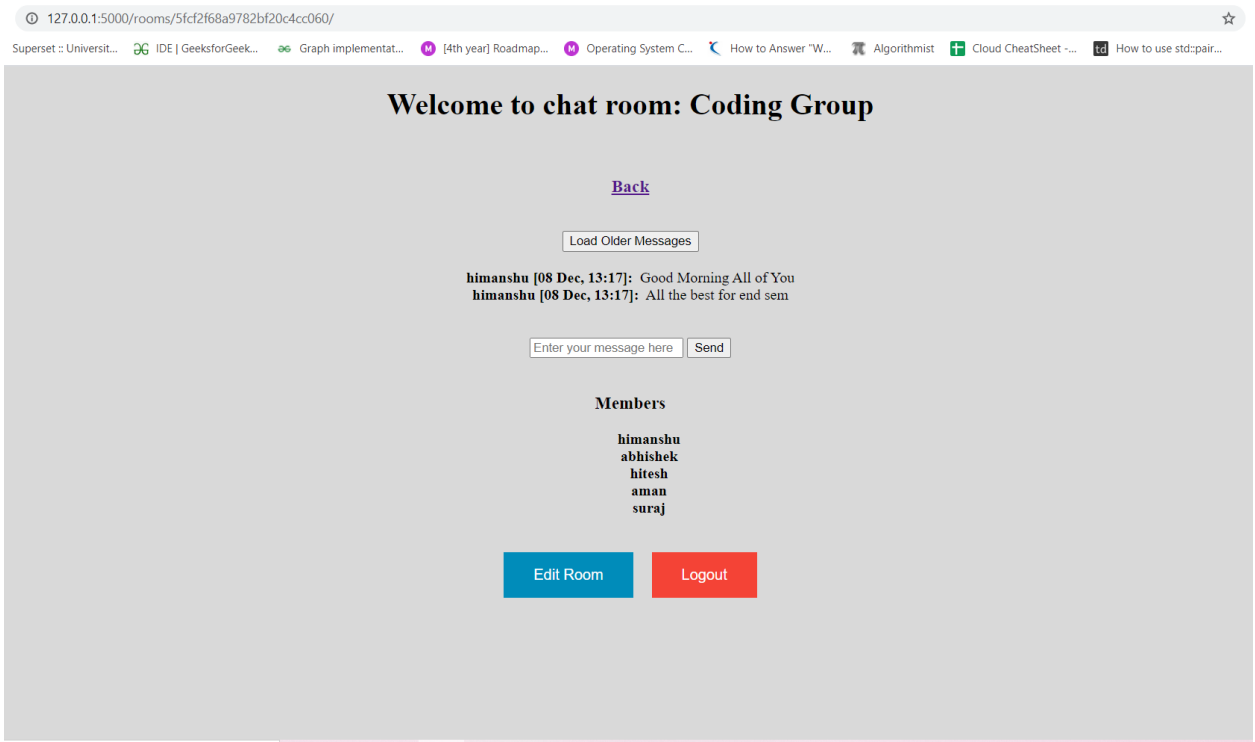
4.2.4 Dashboard



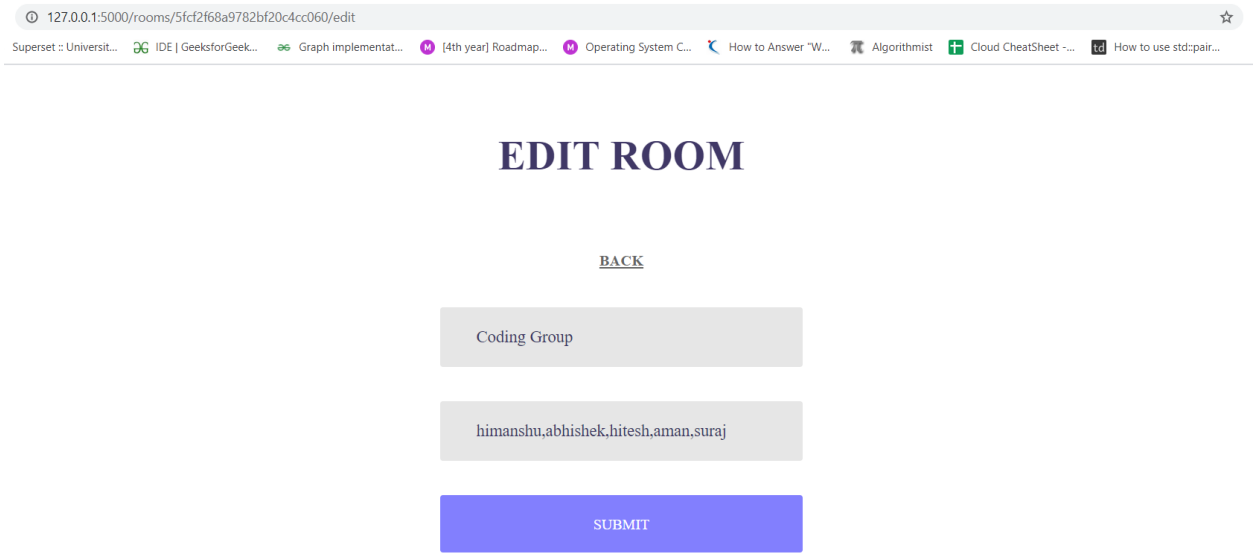
4.2.5 Create Room



4.2.6 My Room & Send Message



4.2.7 Edit Room



5. TESTING

5.1 TESTING GOAL

The goal of Employee Leave Management System Testing is to ensure that the system performs as per the functional requirements specified by client. Most cases tested here are done manually per module.

5.2 FUNCTIONAL REQUIREMENTS TESTING (BLACK BOX TESTING)

5.2.1 Registration for New User

Description: This test will ensure registration of new user			
Data Requirements: Connectivity to database			
Steps #	**Step Description**	**Expected Results**	**Remarks**
1	Open sign Up page	Sign Up page opens	Pass
2	Enter Username	Checked Existence of Username	Pass
3	Enter Email	Checked Existence of Username	Pass
4	Fill Registration Form Password and Click on register	Checked Type of Password and registration successful	Pass
5	Login using credential entered on Sign Up page	Successfully Logged in	Pass

5.2.2 Send Message by User to any Room

Description: This test will ensure that message send successfully			
Data Requirements: Connectivity to complaint Database			
Steps #	**Step Description**	**Expected Results**	**Remarks**
1	Sign In using username and password	Dashboard of Student which shows name of rooms available	Pass
2	If no room created earlier, click on Create Room	Create Room Opens	Pass
3	Choose Room and Open chosen Room	Type text in the text box	Pass
4	Submit form	Message Send Successfully and check by clicking History	Pass

5.3 Non- Functional Requirements Testing

5.3.1 Stability Testing

Stability testing checks to see if the software can continuously function well in or above an acceptable period. This activity of non-functional software testing is oftentimes referred to as load (or endurance) testing.

5.3.2 Usability

Usability testing is needed to check if the user interface is easy to use and understand. This test is used to verify if a user that never use the application can search and read result list within a reasonable time.

5.3.3 Security Testing

Security testing is essential for software which processes confidential data and to prevent system intrusion by hackers.

6. MAINTAINENCE

The Chat Application U CHAT is developed using Python, HTML, CSS, Flask and MangoDB. Thus, it provides an easy Maintainability and Optimization.

Software maintenance in software engineering is the modification of a software product after delivery to correct faults, to improve performance or other attributes. Maintenance covers about 60% of the phase in Software Development Life Cycle.

The U CHAT is the soul property of Himanshu Ranjan & Abhishek Kumar and is developed for all type of users of **the world**. The group shall provide regular updates for the maintenance of the U CHAT Application.

The Maintenance of U CHAT shall include:

- Error Correction
- Bug Fixes
- Enhancement of Capabilities
- Updating of Obsolete Capabilities
- Optimization

The U CHAT Application has been developed for simplifying the process of messaging, matters of the Users and thus it is not affordable to have bugs in the Application but nonetheless some sneaky bugs may find its way to the users. The users are requested to report the bugs as soon as they encounter it so that it could be fixed in the next update.

7. CONCLUSION

Overall, we would class this project as a success. Everything that was outlined in the early stages of the project was achieved and a lot of extra features were also added.

In this project we have used the basics of networking in python. We also learned how to make a GUI for our application. This project teaches us how to create a basic chat application by creating GUI using flask and socketio.