PROJECT REPORT

ON

Online Students Faculty Interaction System

B.Tech (CE) Sem-V

In the Subject of

Advanced Technology

Kenil Gopani (CE-048)

Gopal Malaviya (CE-073)

Vimal Matholiya (CE-076)

Under the Guidance of

Prof. Prashant Jadhav

Department Of Computer Engineering

DHARMSINH DESAI UNIVERSITY

NADIAD-387001, GUJARAT

CERTIFICATE

This is to certify that the project entitled as “**Student Faculty Interaction System**” is a bonafide report of the work carried out by,

1) **Kenil Gopani (ID: 19CEUOS080)**
2) **Gopal Malaviya (ID: 19CEUES049)**
3) **Vimal Matholiya (ID: 19CEUOG120)**

of Department of Computer Engineering, semester V, under the guidance of **Prof. Prashant Jadhav** for the subject Advanced Technologies during the academic year 2021-2022.

Project Guide
Assistance Professor
Prof. Prashant Jadhav
Department of Computer Engineering,
Faculty of Technology,
Dharmsinh Desai University,

Nadiad
Date: 24/10/2021

Head of the Department
Dr. C.K Bhensdadia
Prof. & Head,
Department of Computer Engineering,
Faculty of Technology,
Dharmsinh Desai University,
Nadiad
Date: 24/10/2021

Contents

1. Introduction 4

2. Software Requirement Specifications 5

3. Design 8

4. Implementation Detail 11

5. Screen-Shots 15

6. Conclusion 24

7. Limitation and Future Extension 25

8. Bibliography 26

1. Introduction

This project is about interaction between students and faculty through groups which is created by faculties.

To use the system user must have to register themselves and then sign in to the system. After sign in into the system, admin(faculty) can create group to interact with students. Whenever new group is created a unique Group Code is generated and displayed on the Group info. Admin(faculty) can copy that code and share among students who are subjected to join, by getting group code students can join group accordingly.

After joining the group, students can interact with faculties through messages. If admin(faculty) wants to delete particular group or delete individuals student then he can do so. Admin(faculty) can view profile of students who are joined.

Technologies/Tools used:

- ReactJs
- NodeJs
- ExpressJs
- MongoDB
- Visual Studio Code 2019

2. Software Requirement Specifications

Users of the System :

- Admin
- Faculty
- Student

R.1 ManageAccounts

Description : User can create account, login and update his profile.

R.1.1 Create Account

Description :User can create account into the system by entering his required details.

Input : User enters details.

Output : Login page displayed.

R.1.2 Login

Description :User can login to the system using his email-id and password.

Input : User enters email-id and password.

Processing : Email-id and password are verified. If they are valid then, access is given otherwise access is denied.

Output : User logged in to the system and home page is displayed.

R.1.3 Update Profile

Input : User enter details that he/she wants to update

Output : Confirmation message is displayed.

R.1.4 Logout

Input : User selection.

Output : User is logged out from system.

R.2 Manage Groups

R.2.1 Create Group

Description : User can create group.

Input : User enters group name and group description.

Output : Group is created.

R.2.2 Join Group

Description : User can join any group using unique group-code.

Input : User clicks on join group and enters group-code.

Output : User is added to the group.

R.2.3 Delete Group

Description : User (who has created group) can delete group.

Input : User selection.

Output : Group is deleted.

R.2.4 Modify Group

Description :User (who has created group) can modify group information and remove any member.

Input : User selection.

Output : Modifiction made by user is reflected.

R.3 Manage Chats

R.3.1 Send Message

Input : User's message.

Output : Message is send.

R.3.2 Delete Message

Input : User selection.

Output : Message is deleted.

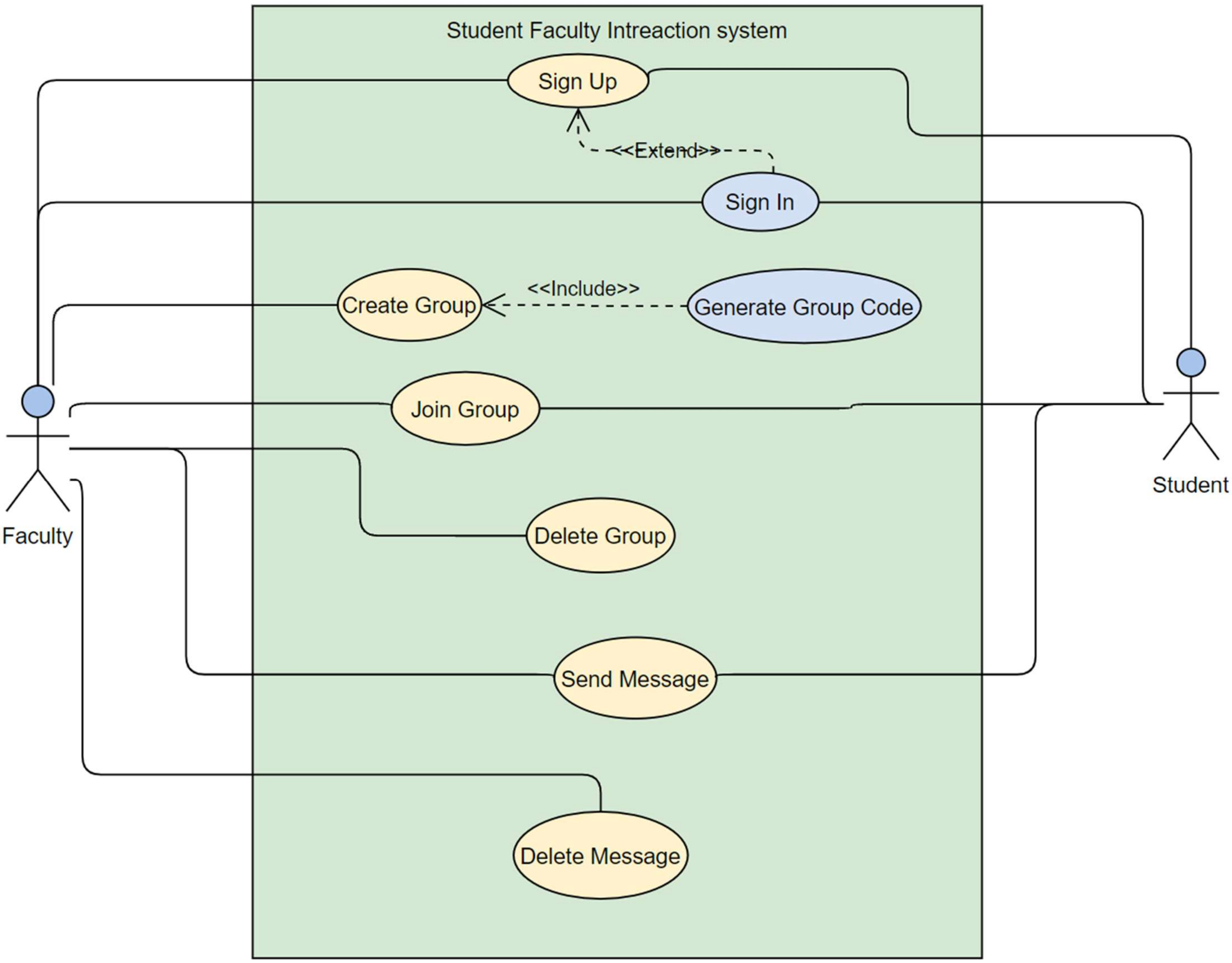
R.3.3 Send Document

Input : User's selected document.

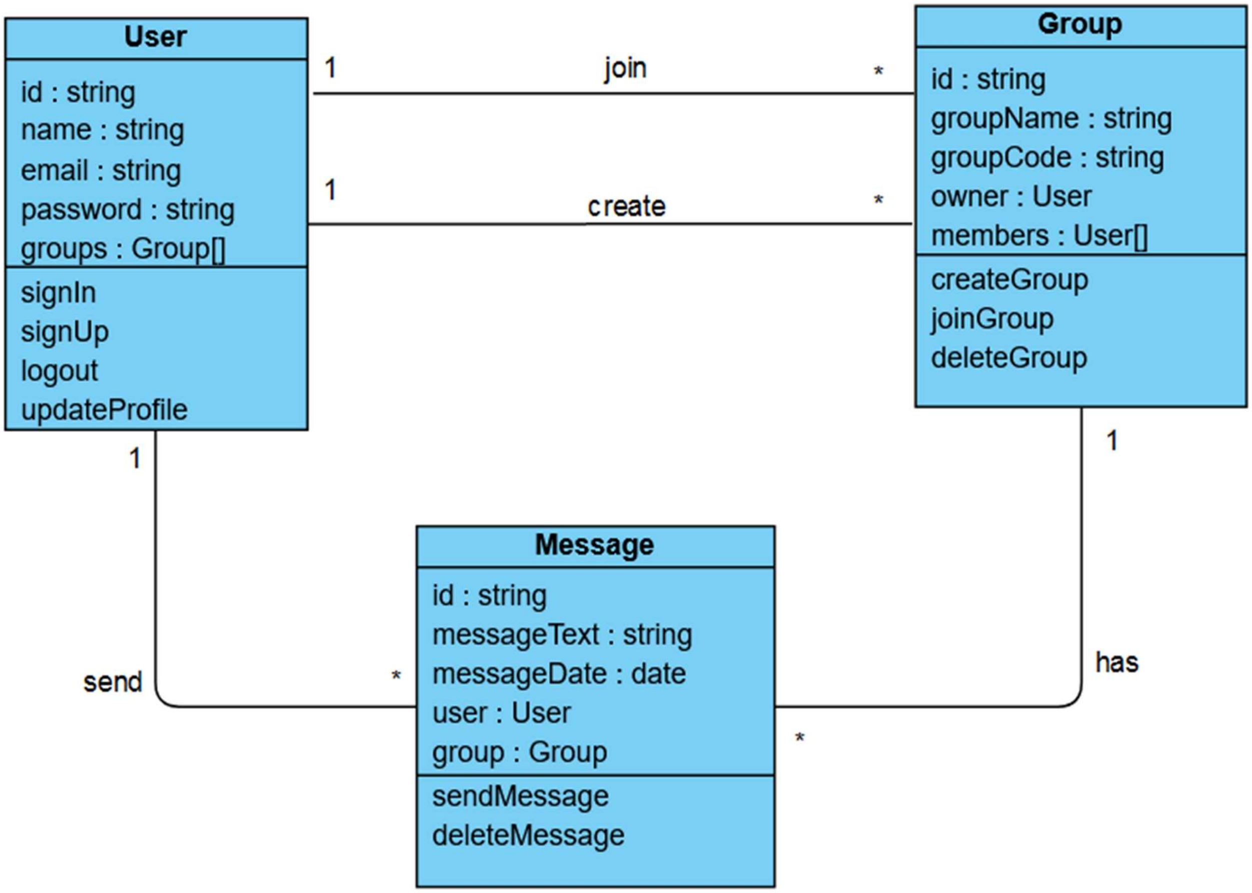
Output : Document is sent.

3. Design

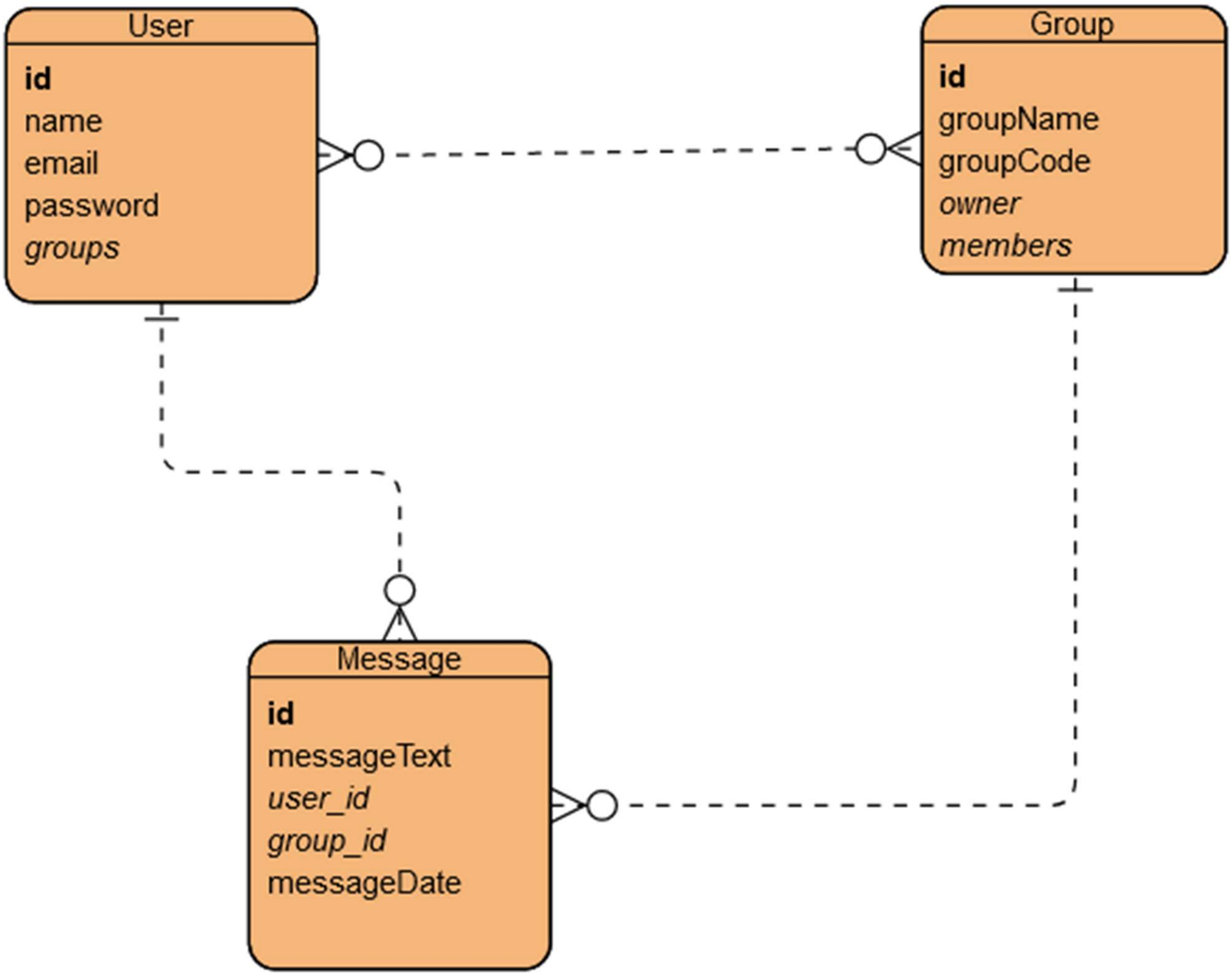
I) Use Case Diagram



II) Class Diagram



III) ER Diagram



4. Implementation Details

I) Modules created and brief description of each modules

Create Group :

Admin(Faculty) can create group using this module. When admin(faculty) create group, unique code will be generated, which they can share with students afterwards.

Join Group :

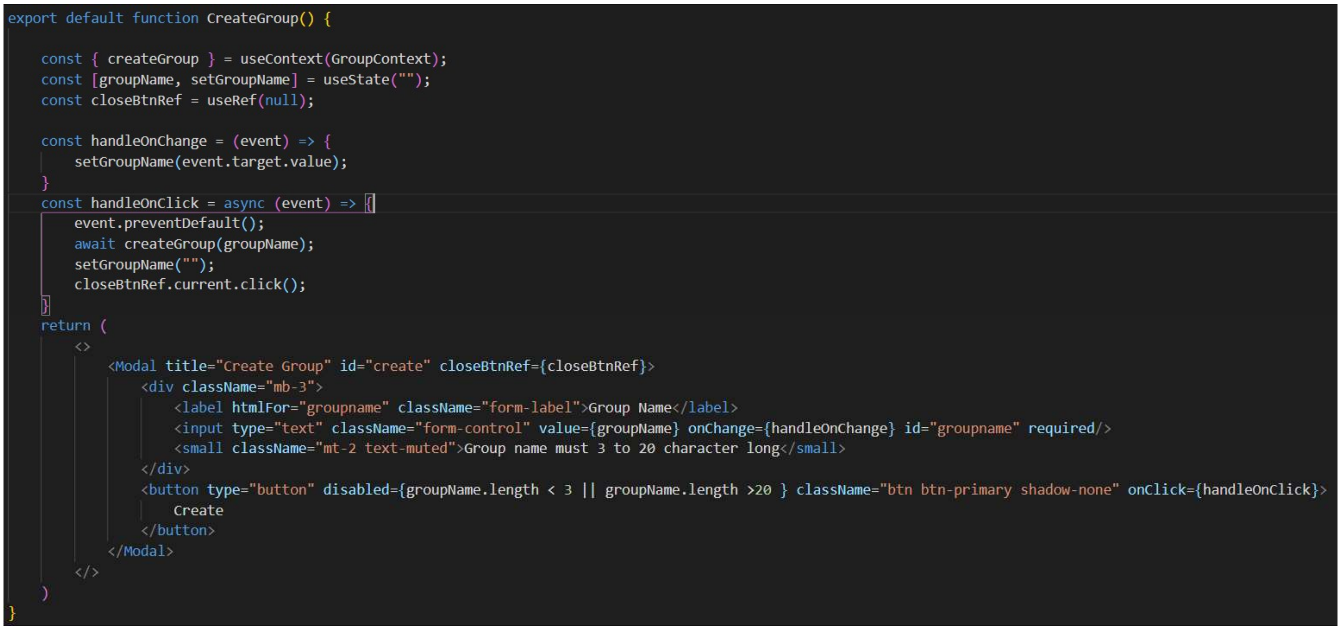
Student can join group using this module. To join group student has to enter group code, which is provided by their faculty.

Send Message :

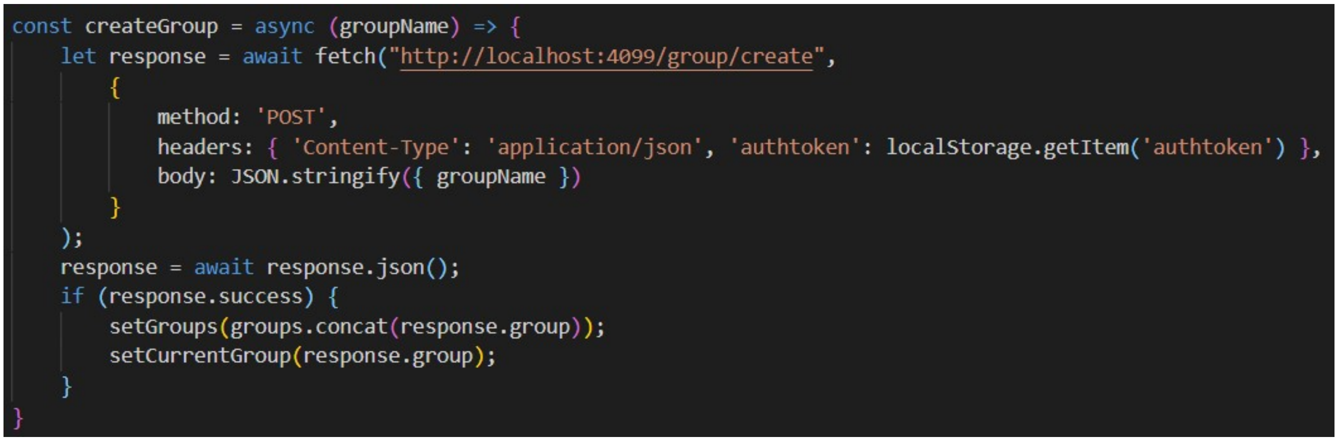
Student or Faculty can interact with each other by sending messages in the group.

II) Function prototypes which implements major functionality

Create Group :

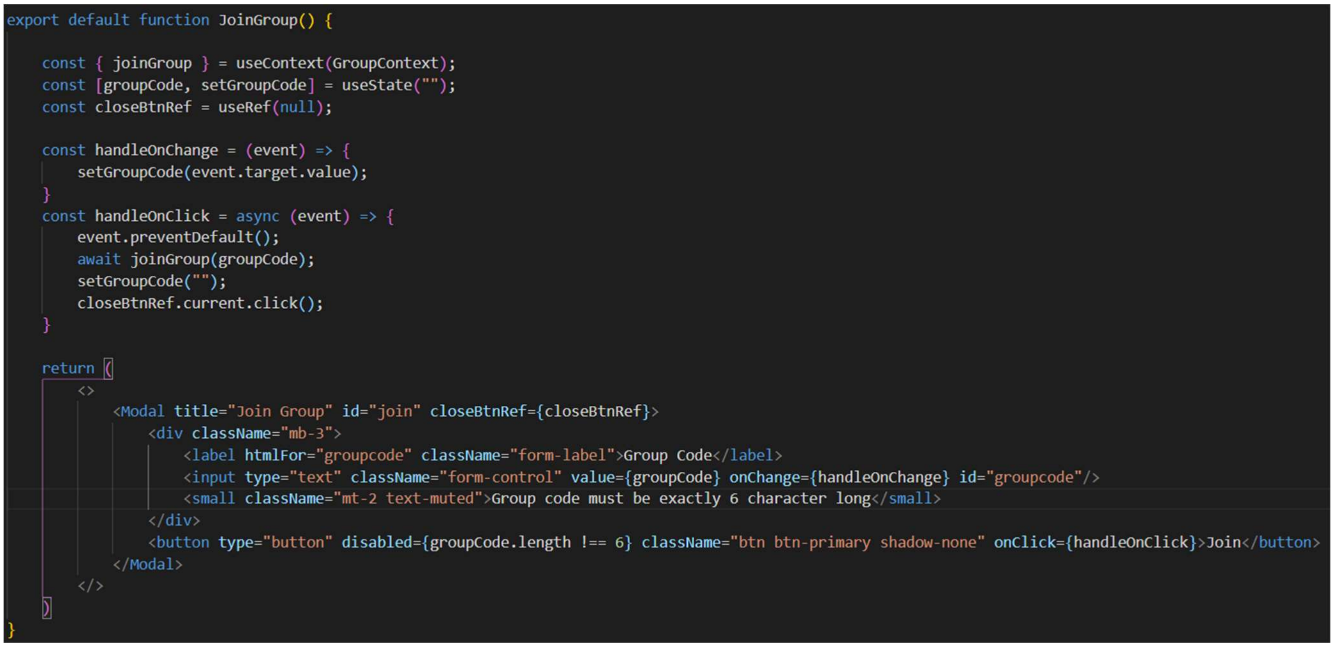


```
export default function CreateGroup() {

    const { createGroup } = useContext(GroupContext);
    const [groupName, setGroupName] = useState("");
    const closeBtnRef = useRef(null);

    const handleOnChange = (event) => {
        setGroupName(event.target.value);
    }
    const handleOnClick = async (event) => {
        event.preventDefault();
        await createGroup(groupName);
        setGroupName("");
        closeBtnRef.current.click();
    }
    return (
        <>
            <Modal title="Create Group" id="create" closeBtnRef={closeBtnRef}>
                <div className="mb-3">
                    <label htmlFor="groupname" className="form-label">Group Name</label>
                    <input type="text" className="form-control" value={groupName} onChange={handleOnChange} id="groupname" required/>
                    <small className="mt-2 text-muted">Group name must 3 to 20 character long</small>
                </div>
                <button type="button" disabled={groupName.length < 3 || groupName.length >20 } className="btn btn-primary shadow-none" onClick={handleOnClick}>
                    Create
                </button>
            </Modal>
        </>
    )
}
```

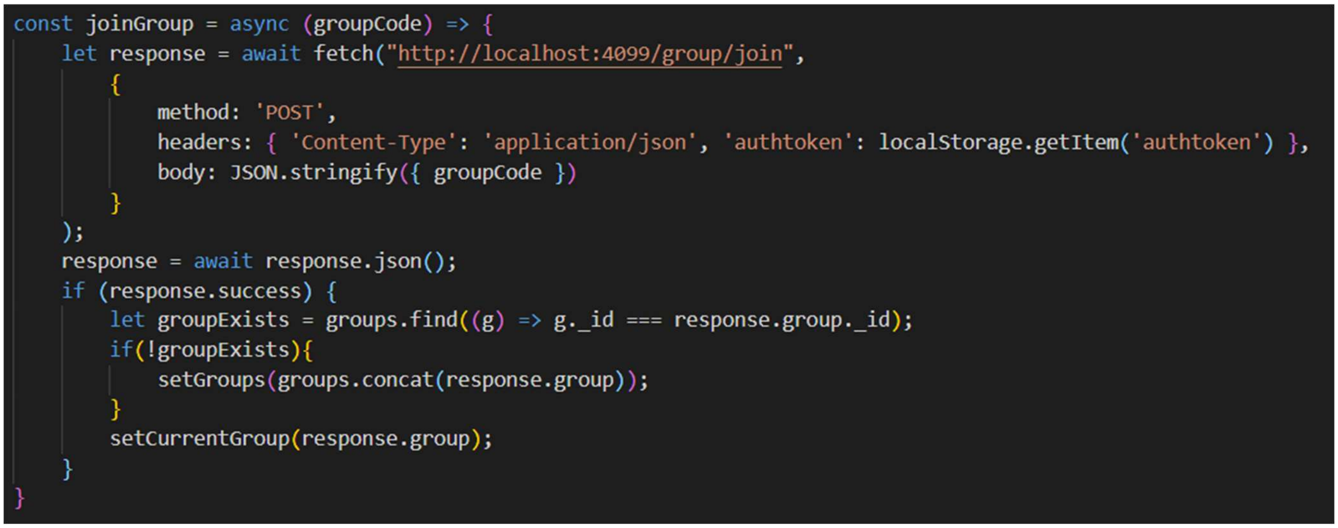


```
const createGroup = async (groupName) => {
    let response = await fetch("http://localhost:4099/group/create",
        {
            method: 'POST',
            headers: { 'Content-Type': 'application/json', 'authtoken': localStorage.getItem('authtoken') },
            body: JSON.stringify({ groupName })
        }
    );
    response = await response.json();
    if (response.success) {
        setGroups(groups.concat(response.group));
        setCurrentGroup(response.group);
    }
}
```

Join Group :

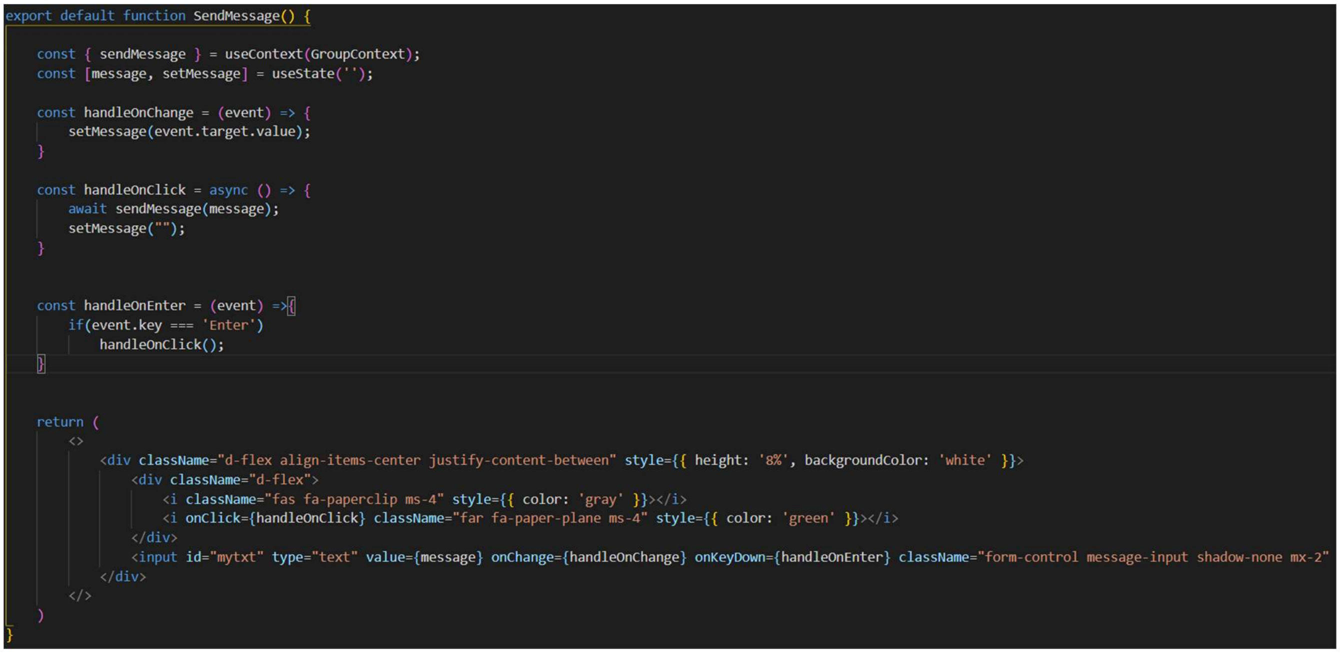


```
export default function JoinGroup() {

    const { joinGroup } = useContext(GroupContext);
    const [groupCode, setGroupCode] = useState("");
    const closeBtnRef = useRef(null);

    const handleOnChange = (event) => {
        setGroupCode(event.target.value);
    }
    const handleOnClick = async (event) => {
        event.preventDefault();
        await joinGroup(groupCode);
        setGroupCode("");
        closeBtnRef.current.click();
    }

    return (
        <>
            <Modal title="Join Group" id="join" closeBtnRef={closeBtnRef}>
                <div className="mb-3">
                    <label htmlFor="groupcode" className="form-label">Group Code</label>
                    <input type="text" className="form-control" value={groupCode} onChange={handleOnChange} id="groupcode"/>
                    <small className="mt-2 text-muted">Group code must be exactly 6 character long</small>
                </div>
                <button type="button" disabled={groupCode.length !== 6} className="btn btn-primary shadow-none" onClick={handleOnClick}>Join</button>
            </Modal>
        </>
    )
}
```

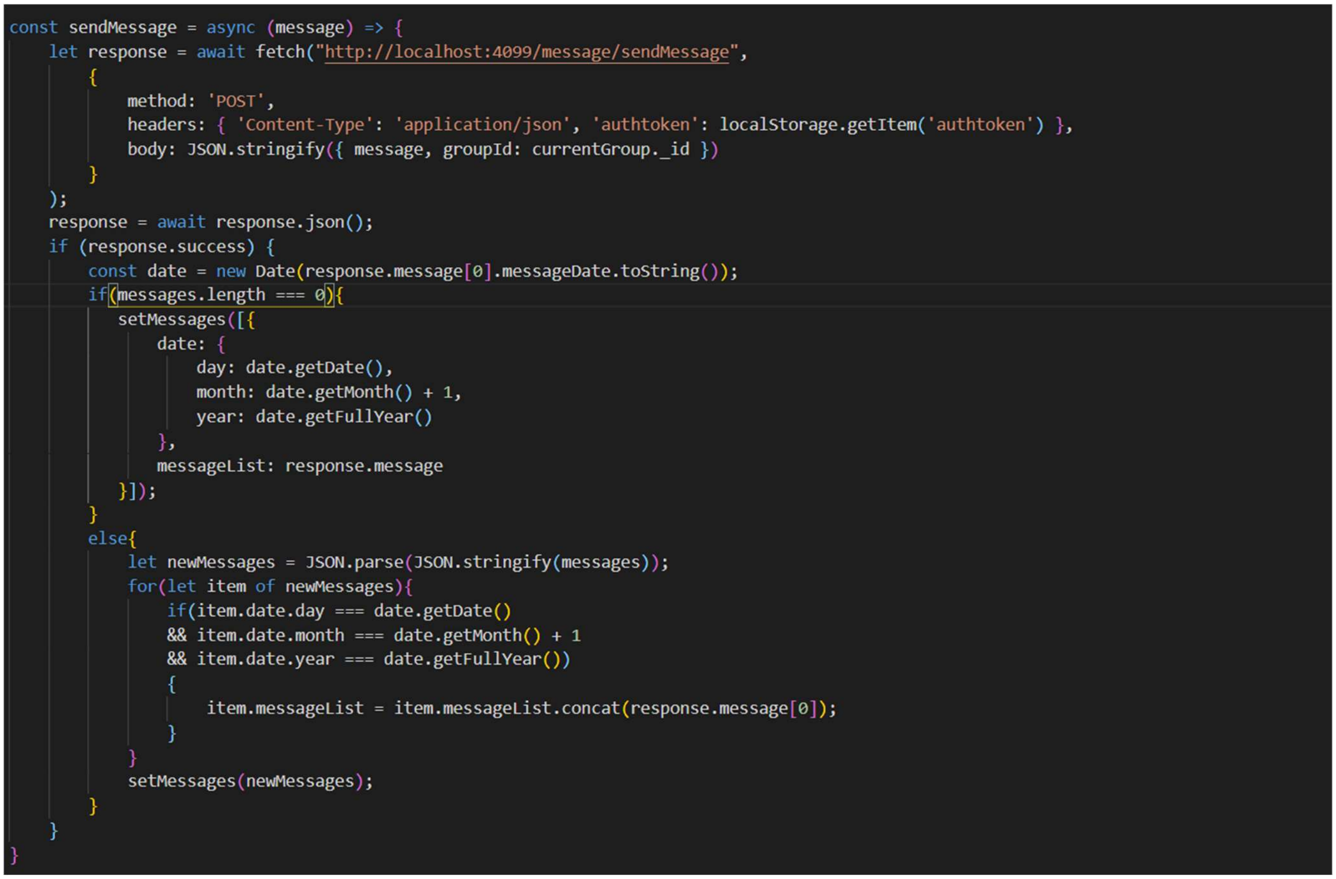


```
const joinGroup = async (groupCode) => {
    let response = await fetch("http://localhost:4099/group/join",
        {
            method: 'POST',
            headers: { 'Content-Type': 'application/json', 'authtoken': localStorage.getItem('authtoken') },
            body: JSON.stringify({ groupCode })
        }
    );
    response = await response.json();
    if (response.success) {
        let groupExists = groups.find((g) => g._id === response.group._id);
        if(!groupExists){
            setGroups(groups.concat(response.group));
        }
        setCurrentGroup(response.group);
    }
}
```

Send Message :

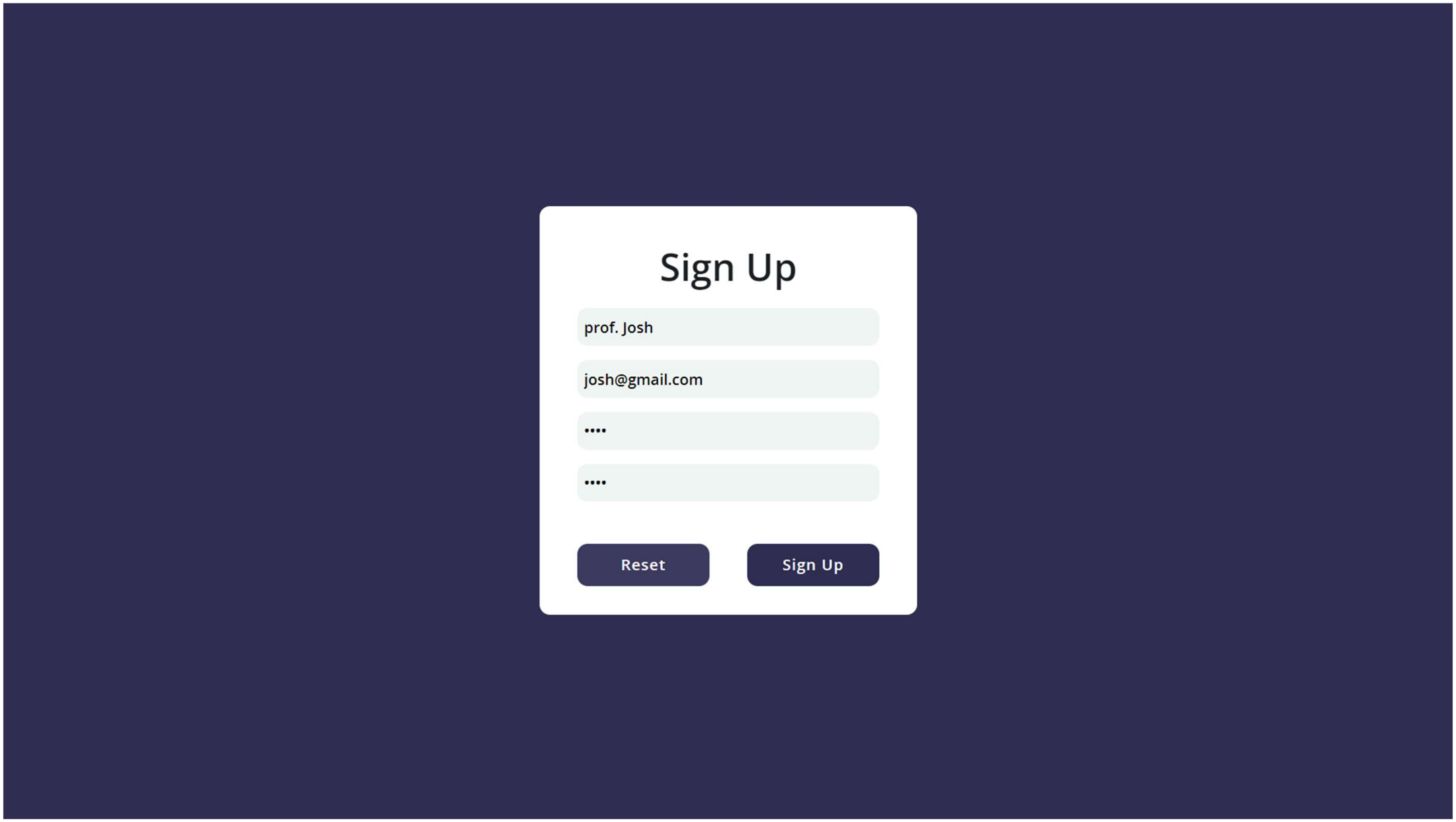


```
export default function SendMessage() {

    const { sendMessage } = useContext(GroupContext);
    const [message, setMessage] = useState('');

    const handleOnChange = (event) => {
        setMessage(event.target.value);
    }

    const handleOnClick = async () => {
        await sendMessage(message);
        setMessage("");
    }


    const handleOnEnter = (event) =>{
        if(event.key === 'Enter')
            handleOnClick();
    }


    return (
        <>
            <div className="d-flex align-items-center justify-content-between" style={{ height: '8%', backgroundColor: 'white' }}>
                <div className="d-flex">
                    <i className="fas fa-paperclip ms-4" style={{ color: 'gray' }}></i>
                    <i onClick={handleOnClick} className="far fa-paper-plane ms-4" style={{ color: 'green' }}></i>
                </div>
                <input id="mytxt" type="text" value={message} onChange={handleOnChange} onKeyDown={handleOnEnter} className="form-control message-input shadow-none mx-2"
            </div>
        </>
    )
}
```



```
const sendMessage = async (message) => {
    let response = await fetch("http://localhost:4099/message/sendMessage",
        {
            method: 'POST',
            headers: { 'Content-Type': 'application/json', 'authtoken': localStorage.getItem('authtoken') },
            body: JSON.stringify({ message, groupId: currentGroup._id })
        }
    );
    response = await response.json();
    if (response.success) {
        const date = new Date(response.message[0].messageDate.toString());
        if(messages.length === 0){
            setMessages([{
                date: {
                    day: date.getDate(),
                    month: date.getMonth() + 1,
                    year: date.getFullYear()
                },
                messageList: response.message
            }]);
        }
        else{
            let newMessages = JSON.parse(JSON.stringify(messages));
            for(let item of newMessages){
                if(item.date.day === date.getDate()
                && item.date.month === date.getMonth() + 1
                && item.date.year === date.getFullYear())
                {
                    item.messageList = item.messageList.concat(response.message[0]);
                }
            }
            setMessages(newMessages);
        }
    }
}
```

5. Screen-shots

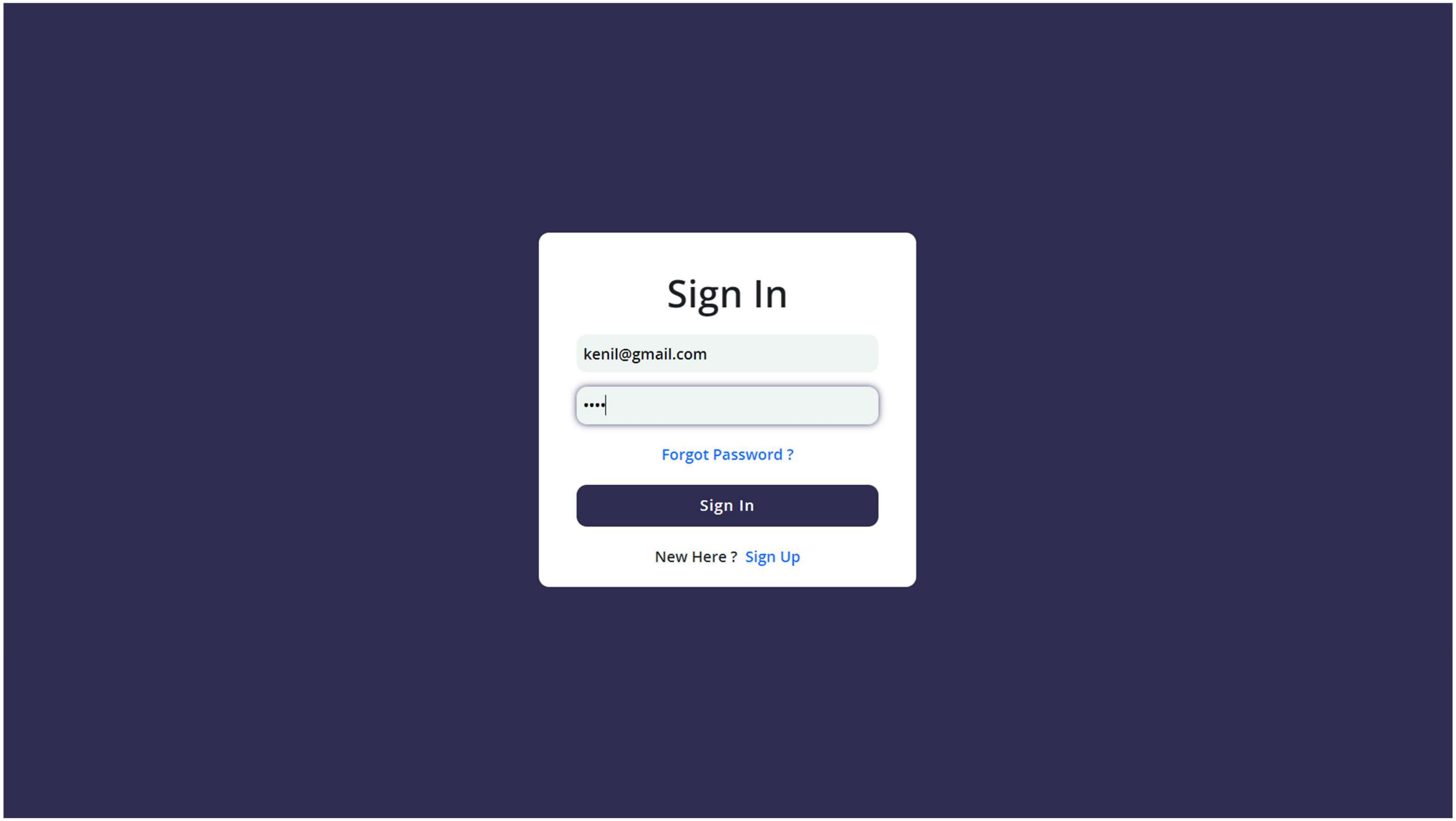


prof. Josh

josh@gmail.com

••••

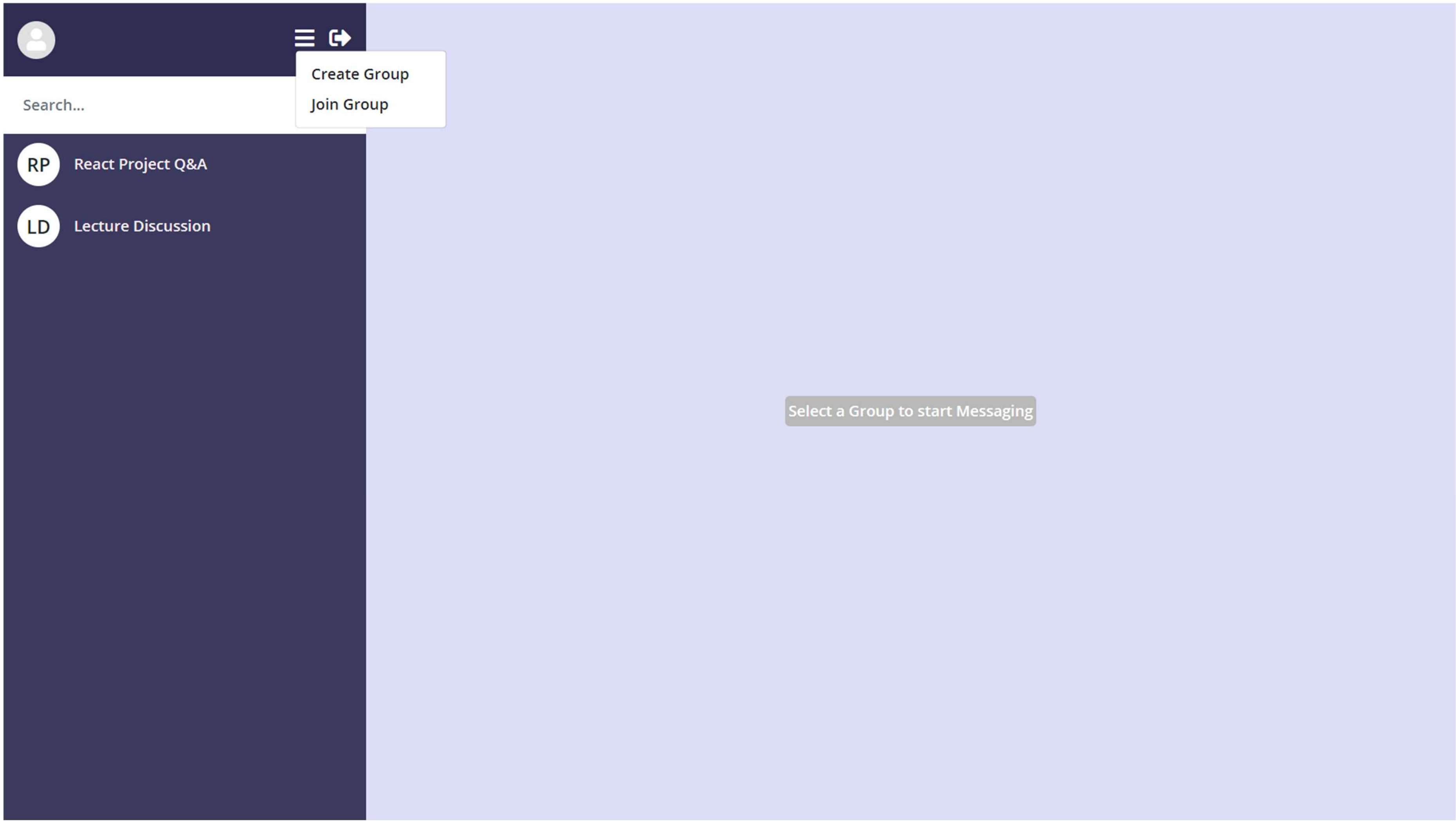
••••



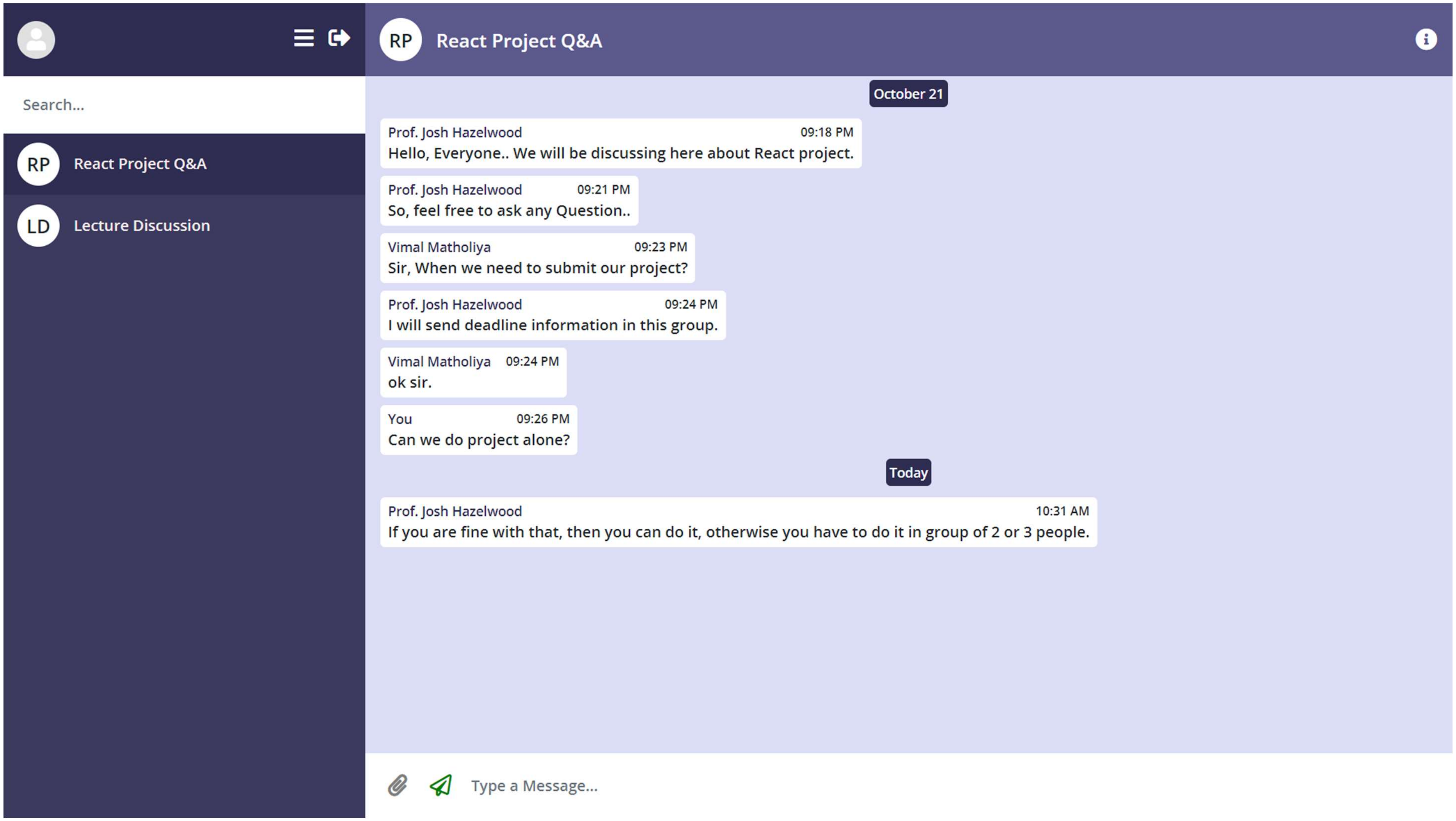
Sign In

kenil@gmail.com

••••



Select a Group to start Messaging



October 21

Prof. Josh Hazelwood 09:18 PM
Hello, Everyone.. We will be discussing here about React project.

Prof. Josh Hazelwood 09:21 PM
So, feel free to ask any Question..

Vimal Matholiya 09:23 PM
Sir, When we need to submit our project?

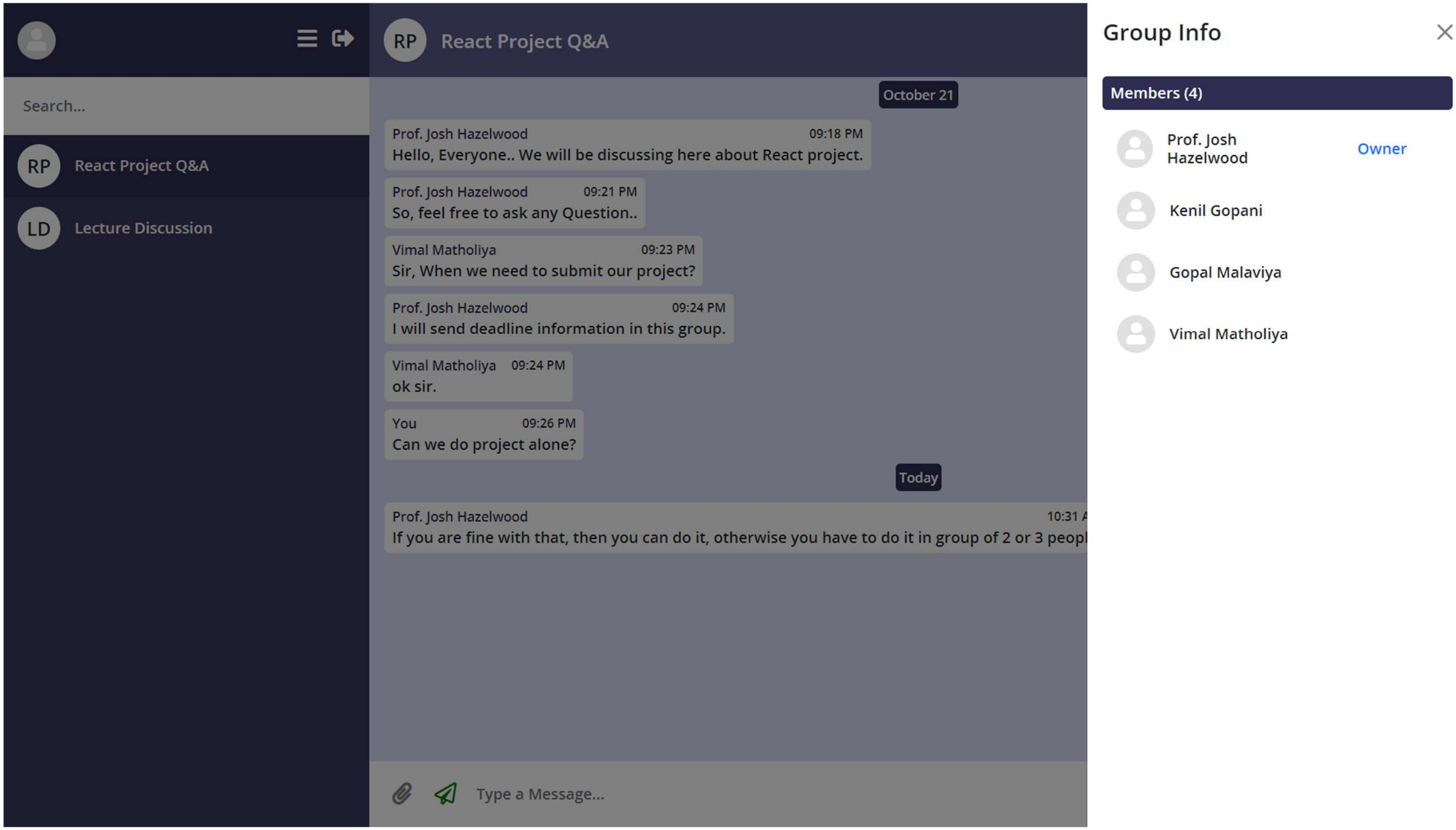
Prof. Josh Hazelwood 09:24 PM
I will send deadline information in this group.

Vimal Matholiya 09:24 PM
ok sir.

You 09:26 PM
Can we do project alone?

Today

Prof. Josh Hazelwood 10:31 AM
If you are fine with that, then you can do it, otherwise you have to do it in group of 2 or 3 people.



RP React Project Q&A

October 21

Prof. Josh Hazelwood 09:18 PM
Hello, Everyone.. We will be discussing here about React project.

Prof. Josh Hazelwood 09:21 PM
So, feel free to ask any Question..

Vimal Matholiya 09:23 PM
Sir, When we need to submit our project?

Prof. Josh Hazelwood 09:24 PM
I will send deadline information in this group.

Vimal Matholiya 09:24 PM
ok sir.

You 09:26 PM
Can we do project alone?

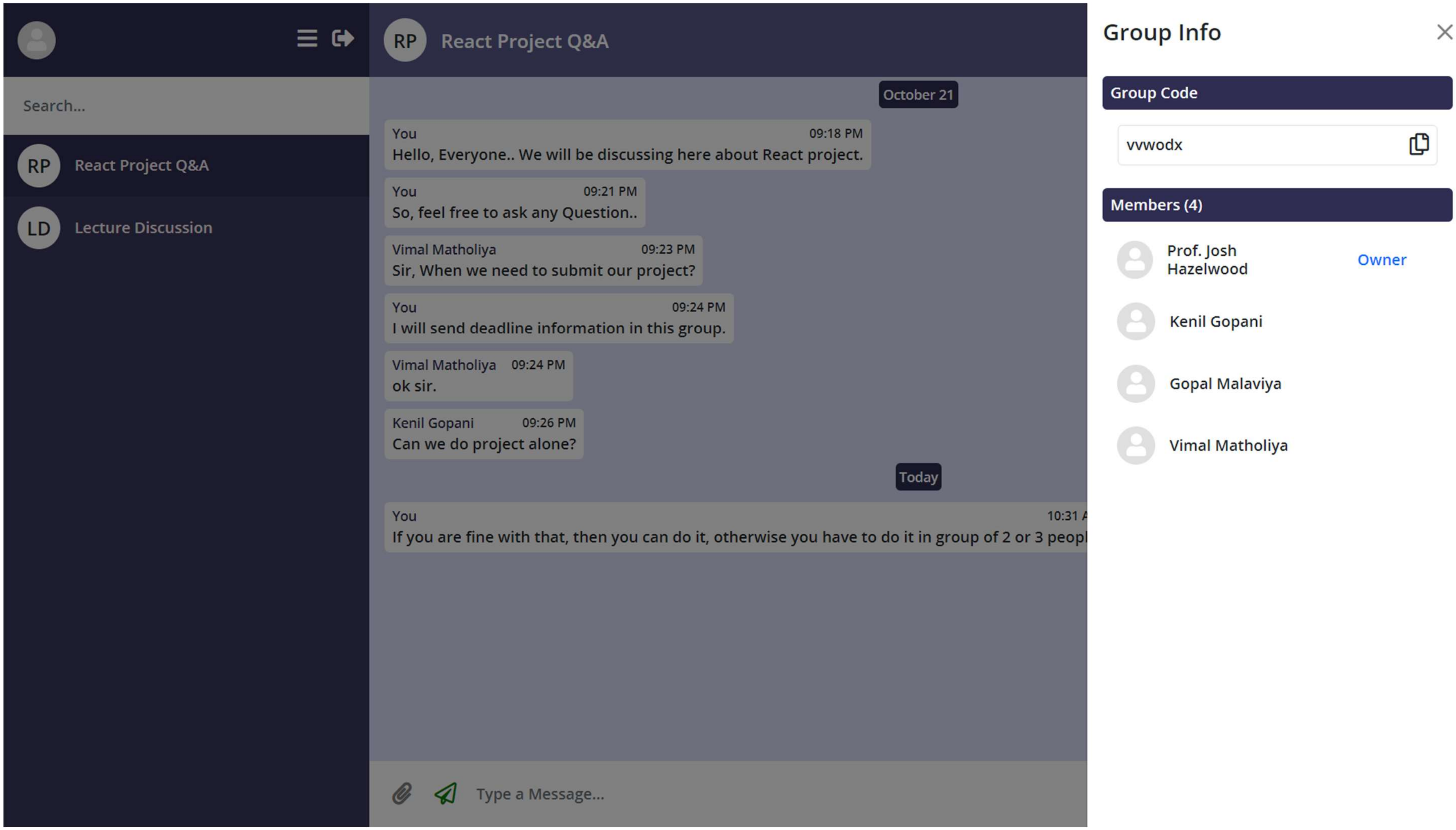
Today

Prof. Josh Hazelwood 10:31 A
If you are fine with that, then you can do it, otherwise you have to do it in group of 2 or 3 peopl

Group Info

Members (4)

Prof. Josh Hazelwood	Owner
Kenil Gopani	
Gopal Malaviya	
Vimal Matholiya	



October 21

You 09:18 PM
Hello, Everyone.. We will be discussing here about React project.

You 09:21 PM
So, feel free to ask any Question..

Vimal Matholiya 09:23 PM
Sir, When we need to submit our project?

You 09:24 PM
I will send deadline information in this group.

Vimal Matholiya 09:24 PM
ok sir.

Kenil Gopani 09:26 PM
Can we do project alone?

Today

You 10:31 A
If you are fine with that, then you can do it, otherwise you have to do it in group of 2 or 3 peopl

Group Info

Group Code

vvwodx

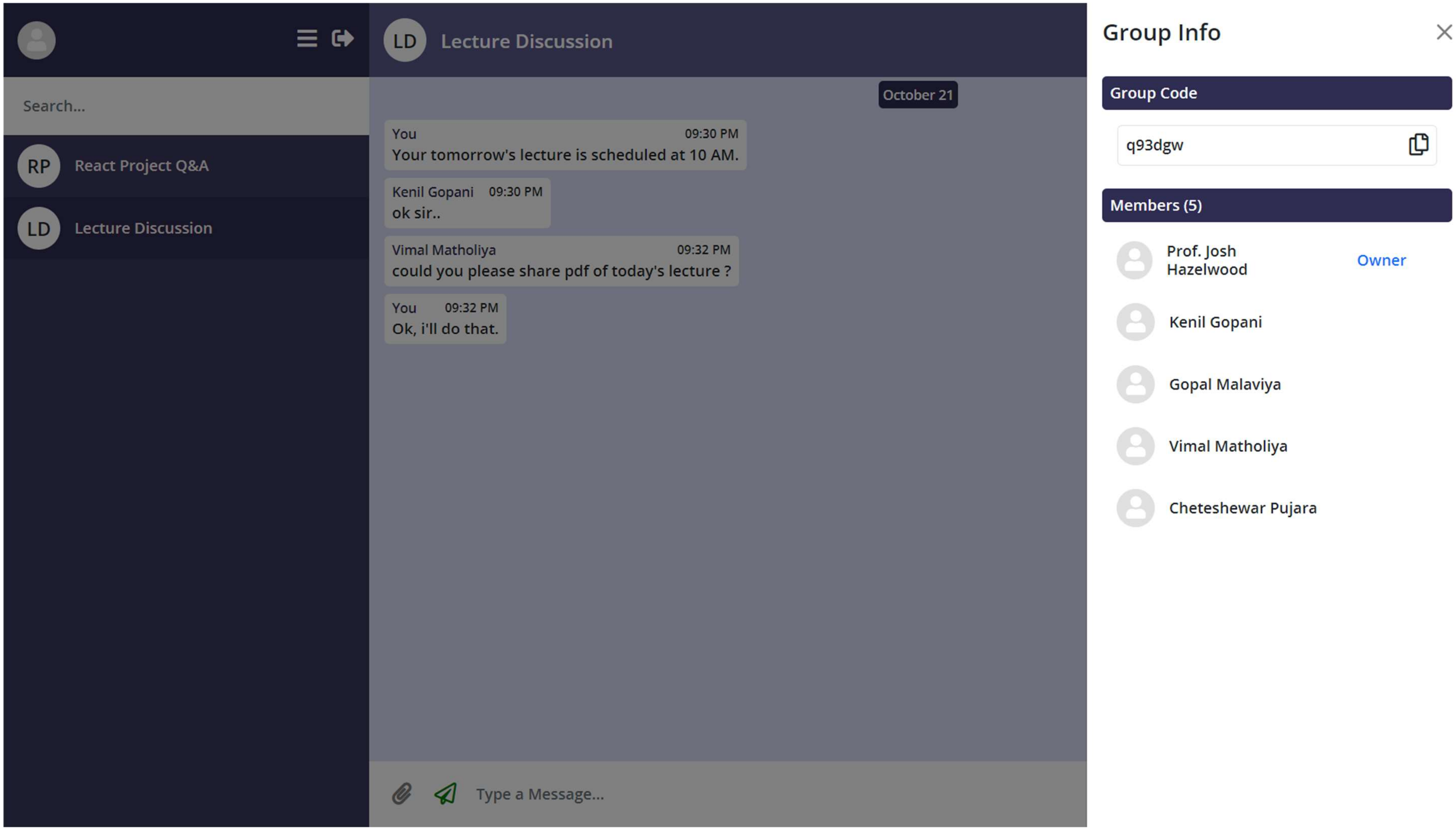
Members (4)

Prof. Josh Hazelwood Owner

Kenil Gopani

Gopal Malaviya

Vimal Matholiya



You 09:30 PM
Your tomorrow's lecture is scheduled at 10 AM.

Kenil Gopani 09:30 PM
ok sir..

Vimal Matholiya 09:32 PM
could you please share pdf of today's lecture ?

You 09:32 PM
Ok, i'll do that.

Group Info

Group Code

q93dgw

Members (5)

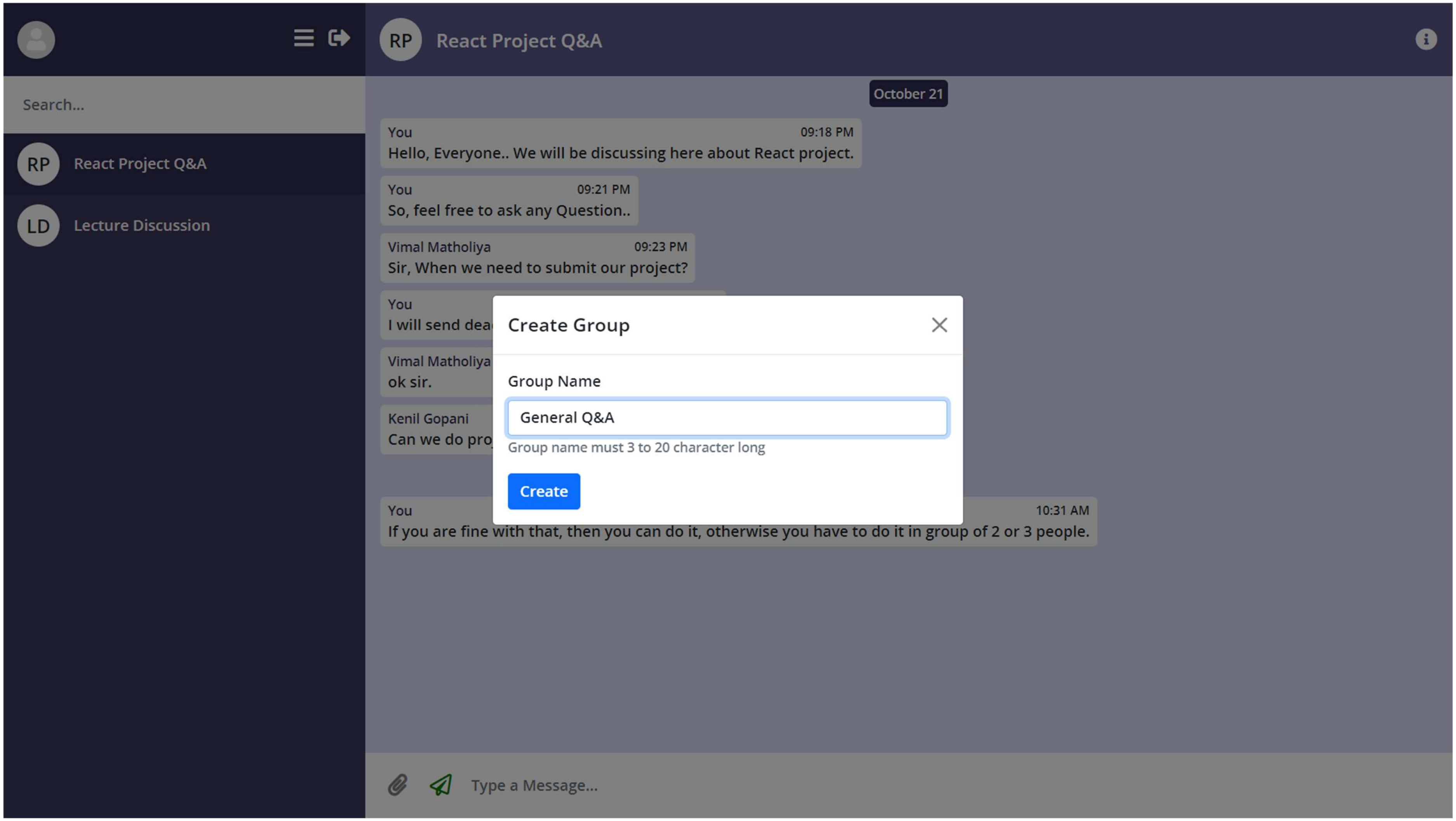
Prof. Josh Hazelwood Owner

Kenil Gopani

Gopal Malaviya

Vimal Matholiya

Cheteshewar Pujara



You 09:18 PM
Hello, Everyone.. We will be discussing here about React project.

You 09:21 PM
So, feel free to ask any Question..

Vimal Matholiya 09:23 PM
Sir, When we need to submit our project?

You
I will send dea

Vimal Matholiya
ok sir.

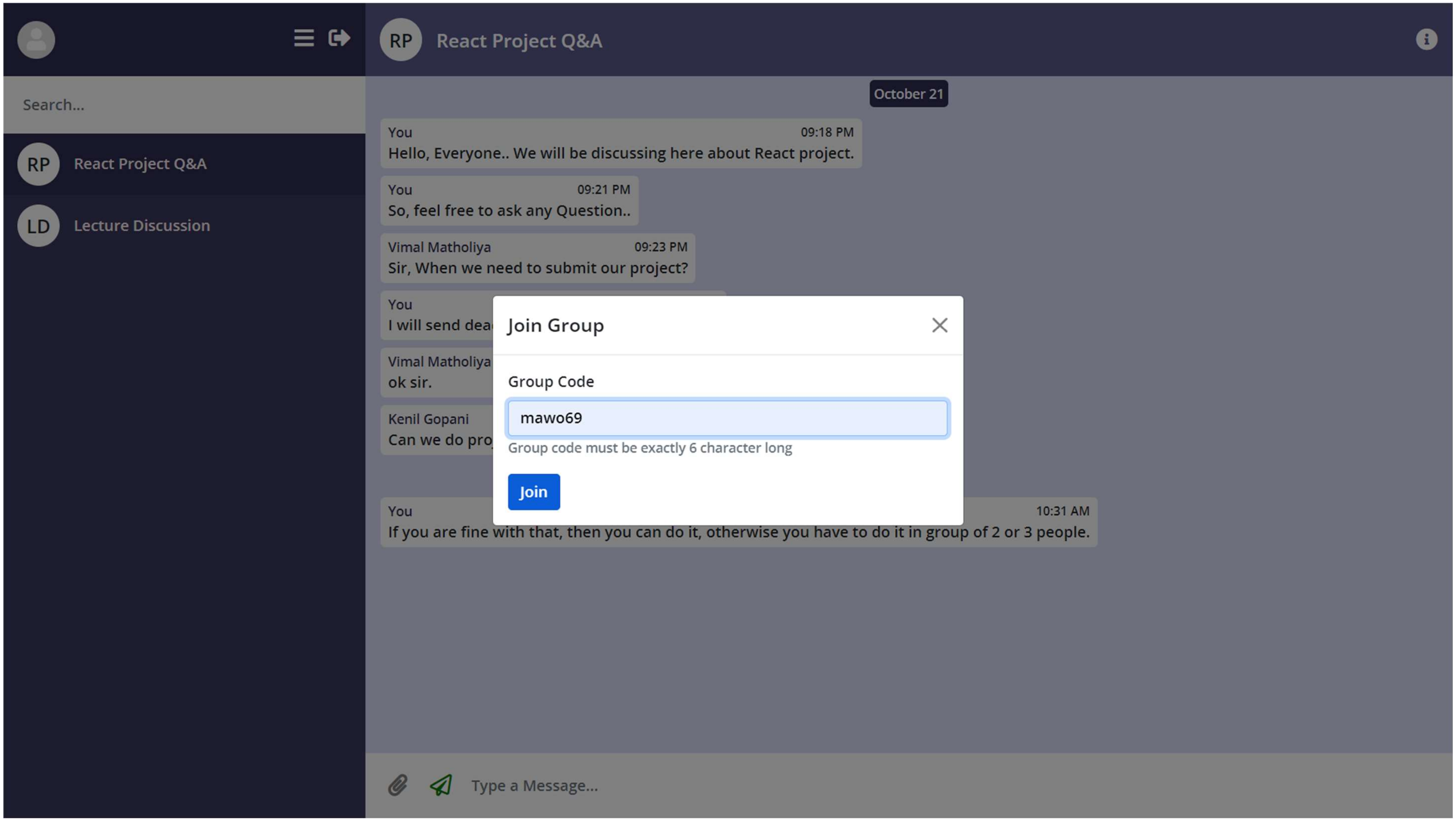
Kenil Gopani
Can we do pro

You 10:31 AM
If you are fine with that, then you can do it, otherwise you have to do it in group of 2 or 3 people.

Create Group ✕

Group Name: General Q&A

Group name must 3 to 20 character long



You 09:18 PM
Hello, Everyone.. We will be discussing here about React project.

You 09:21 PM
So, feel free to ask any Question..

Vimal Matholiya 09:23 PM
Sir, When we need to submit our project?

You
I will send dea

Vimal Matholiya
ok sir.

Kenil Gopani
Can we do pro

You 10:31 AM
If you are fine with that, then you can do it, otherwise you have to do it in group of 2 or 3 people.

Join Group

Group Code: mawo69

Group code must be exactly 6 character long

6. Conclusion

All the functionalities are implemented after understanding all models and diagrams of system. By the use of system all users can interact with each other.

Functionalities are successfully implemented are as below mentioned.

- Sign In
- Sign Up
- Logout
- Create Group
- Join Group
- Send Message
- View Group Information

After the implementation all functionalities were successfully tested, and working properly.

7. Limitations and Future Extensions

Limitation:

- User can’t delete a group or message.
- User can’t see or update their own profile.
- User has to refresh to see new messages.

Future Extension:

- User will be able to send files.
- User interface will be improved to provide better interaction with system.
- There will be more security for user’s data and personal information.
- Real time Message preview.

8. Bibliography

Referred links for project:

- www.reactjs.org.com
- www.stackoverflow.com
- www.getbootstrap.com/docs/5.0.com
- www.fontawesome.com