# ECHOVERSE

# MAJOR PROJECT REPORT

Submitted by

**NETHRA HARINI.M**
**21BCA036**

Under the Guidance Of

**Dr. C. ARUNPRIYA M.Sc., M.Phil., Ph.D**
**Associate Professor**
Department of Computer Applications (UG)

In partial fulfillment of the requirement of the award of the Degree of
**BACHELOR OF COMPUTER APPLICATIONS**
of Bharathiar University

**DEPARTMENT OF COMPUTER APPLICATIONS(UG)**

**PSG College of Arts & Science**
An Autonomous college - Affiliated to Bharathiar University
Accredited with 'A++' Grade by NAAC (4th cycle)
College with potential for excellence
(Status Awarded by the UGC)
Star College Status Awarded by DBT- MST
An ISO 9001:2015 certified Institution
Coimbatore- 641014

**APRIL 2024**

# CERTIFICATE

**DEPARTMENT OF COMPUTER APPLICATIONS**

**PSG College of Arts & Science**

An Autonomous college - Affiliated to Bharathiar University
Accredited with 'A++' Grade by NAAC (4th cycle)
College with potential for excellence
(Status Awarded by the UGC)
Star College Status Awarded by DBT- MST
An ISO 9001:2015 certified Institution
Coimbatore- 641014

# Certificate

This is to certify that this major project work entitled **Echoverse** is a bonafide record of work done by **Nethra Harini. M (21BCA036)** in partial fulfillment of the requirement for the award of Degree of **Bachelor of Computer Applications** of Bharathiar University.

**Faculty Guide** **Head of the Department**

Submitted for Viva - Voce Examination held on

**Internal Examiner** **External Examiner**

# DECLARATION

# DECLARATION

I **Nethra Harini. M (21BCA036)**, hereby declare that this major project work entitled **Echoverse**, is submitted to PSG College of Arts & Science (Autonomous), Coimbatore in partial Fulfillment for the award of **Bachelor of Computer Applications**, is a record of original work done by me under the supervision and guidance of **Dr. C. Arunpriya M.Sc., M.Phil., Ph.D** Associate Professor in Department of Computer Applications(UG), PSG College of Arts & Science, Coimbatore.

This project work has not yet been submitted by me for the award of any other Degree / Diploma / Associative ship / Fellowship or any other similar degree to any other University.

**Place :** Coimbatore **Nethra Harini. M**

**Date :** **(21BCA036)**

# ACKNOWLEDGEMENT

# ACKNOWLEDGEMENT

With great gratitude I would like to acknowledge the help of those who contributed their valuable suggestions and timely assistance to complete this work.

First and foremost, I would like to extend my heartfelt gratitude and place my sincere thanks to **Thiru L. Gopalakrishnan** managing Trustee, PSG & SONS Charities, Coimbatore for providing all sorts of support and necessary facilities throughout the course.

I express my deep sense of gratitude to Secretary, **Dr. T. Kannaian M.Sc., MTech., Ph.D.** for permitting me to undertake this work.

I thank our Principal, **Dr. D. Brindha M.Sc., M.A(Yoga)., MPhil., Ph.D.** for her support and constant source of inspiration through the course of the project.

I would like to thank our Vice Principals, **Dr. A. Anguraj M.Sc., M.Phil., PhD.** and Faculty-In-Charge (Student Affairs), **Dr. M. Umarani, M.Com, M.Phil.,** for their support.

I owe my deepest gratitude to **Dr. R. Sudha MCA., M.Phil., Ph.D.** Associate professor and Head, Department of Computer Applications(UG), PSG College of Arts & Science, Coimbatore, for her encourage me to pursue my goals.

My sincere thanks to **Dr. C. Arunpriya M.Sc., M.Phil., Ph.D.** Associate professor, Department of Computer Applications(UG), PSG College of Arts & Science, Coimbatore for her valuable suggestions, support and guidance as my internal guide, without which my work would not have reached the present form.

Last but not the least, I am greatly indebted to my parents and friends for their kind co-operation in each and every step I took in this project.

# ABSTRACT

# ABSTRACT

Echoverse stands as a pioneering web platform,working for the online communication and collaboration. Its foundation lies in Express.js, ensuring streamlined server-side processing and routing for optial performance. this is Socket.IO, seamlessly enabling real-time bidirectional communication between clients and servers, thus ensuring instantaneous message delivery. With HTML providing the structural backbone, Echoverse boasts compatibility across devices and browsers, while CSS enhances its visual presentation with customizable styling options. JavaScript drives interactivity, facilitating dynamic features such as live chat, notifications, and user interactions. Through Echoverse, users seamlessly join servers, engage in real-time conversations, and explore diverse communities. By This modern web technologies, Echoverse is set to redefine online connectivity and interaction, and engaging digital community that works on collaboration and connection.

# TABLE OF CONTENTS

# Table of Contents

# 1. INTRODUCTION

## 1.1 Overview

In a digital landscape where collaboration and communication are paramount, Echoverse emerges as a groundbreaking platform designed to transform the way teams interact and engage with one another.

Echoverse is a fusion of the best features from server collaboration platforms like Discord, enriched with purpose-driven functionalities, setting a new standard for online discourse.

At its core, Echoverse provides a dynamic and intuitive environment for individuals, teams, and communities to converge, communicate, and collaborate seamlessly.

Whether it's for business educational initiatives, or recreational pursuits, Echoverse offers a versatile space where users can connect and Echoverse redefines the landscape of collaborative communication by offering a platform that transcends traditional boundaries.

# 2. SYSTEM SPECIFICATION

System requirements are the minimum and/or maximum hardware and software specifications that a system or application must meet in order to function properly. In order to determine the system requirements for a particular system or application, it's important to carefully analyze the functional and non-functional requirements of the system or application, and to identify any constraints or other factors that may affect its performance.

## 2.1 Hardware Specification

Hardware specifications are technical description of the computer's components and capabilities. Processor speed, model and manufacturer, etc.., So, the hardware components required for the proposed system are

| | |
|---|---|
| Processor | : AMD Ryzen 5 500H |
| Ram | : 4.00 GB |
| Hard Disk Drive | : 500 GB HDD |
| Key Board | : 104 Keys Standard |
| Mouse | : Optical Mouse |
| Monitor | : 15'' Color Monitor |

## 2.2 Software Specification

Software requirement specification is a description of a software system to be developed. It lays out functional and non-functional requirements and it also describes the operating system and tool used in the system and they are

| | |
|---|---|
| Operating System | : Windows |
| Development Tool | : Visual Studio Code |
| Front End | : React |
| Back End | : NodeJS |

## 2.3 Software Description

### Visual Studio

Microsoft Visual Studio is an IDE made by Microsoft and used for different types of software development such as computer programs, websites, web apps, web services, and mobile apps. It contains completion tools, compilers, and other features to facilitate the software development process.

Enable additional languages, themes, debuggers, commands, and more. VS Code's growing community shares their secret sauce to improve your workflow.

### Advantages

- Excellent support for multiple programming languages: Visual Studio Code offers great support for a wide range of programming languages, including Java, Python, C++, JavaScript, and more. It includes features like syntax highlighting, code completion, and language-specific tools that make it a great choice for developers working with multiple languages.
- A wide range of features: Visual Studio Code offers many features that make it a powerful code editor. Some of the key features include Git integration, debugging tools, and extensions that allow you to customize your workflow.
- High customizability: Visual Studio Code is highly customizable, allowing you to configure the interface and keyboard shortcuts to your liking. This makes it a great choice for developers who want to tailor their coding environment to their specific needs.
- Large community of developers: Visual Studio Code has a large community of developers who create and maintain extensions and plugins that add new functionality to the editor. This means you can find a wide range of extensions to help with your coding workflow.
- Fast and efficient: Visual Studio Code is fast and efficient, with a small footprint. This makes it a great choice for developers who want a code editor that won’t slow down their computer.

## React

A powerful JavaScript library for building dynamic user interfaces. React's component-based architecture allows developers to create modular, reusable UI elements, facilitating efficient development and maintenance workflows. Through React's virtual DOM (Document Object Model), Echoverse achieves optimal rendering performance, enabling smooth navigation, fast interactions, and real-time updates without sacrificing responsiveness.

## Nodejs

A versatile runtime environment for executing JavaScript code on the server-side. Node.js enables efficient event-driven architecture, asynchronous I/O operations, and non-blocking concurrency, making it ideal for handling concurrent connections and real-time communication in Echoverse. With Node.js, Echoverse achieves high performance, scalability, and responsiveness, crucial for supporting the platform's growing user base and dynamic interactions.

## Express

The Echoverse utilizes the Express.js framework to streamline server-side development, routing, and middleware management. Express.js simplifies the creation of RESTful APIs, authentication mechanisms, and request handling, enabling developers to build robust backend services efficiently.

## Socket IO

In your project, Socket.IO is utilized for real-time communication between the client-side and server-side components. It enables bidirectional, event-based communication, allowing instant updates and interactions between users without the need for manual page refreshes. Socket.IO facilitates features like instant messaging, live notifications, and collaborative functionalities, providing users with a seamless and interactive experience. By leveraging Socket.IO, your project ensures efficient and responsive communication, enhancing user engagement and satisfaction.

# 3. SYSTEM ANALYSIS

## 3.1 Existing System

The existing system is a react based solution designed to chat and collab within servers. Using This platform such as react and Socket.io, the chat processes chat room with multiple servers.

## Functionality

- Real-time messaging: Users can exchange messages instantly within server rooms and also it uses real time messaging.
- Server collaboration: Users can create, join, and collaborate within servers dedicated to specific topics or projects.
- Temporary servers: Users can create temporary servers for short-term collaborations or discussions.

## Disadvantages

- Limited interactivity and responsiveness due to reliance on traditional HTML/CSS-based frontend.
- Complexity and maintenance challenges as the codebase grows, leading to longer development cycles and higher maintenance costs.
- Scalability concerns due to the lack of modern frontend frameworks optimized for handling large-scale applications.
- Cross-browser compatibility issues may arise, affecting the consistency of the user experience across different browsers and devices.
- Reduced development efficiency without modern frontend development tools and methodologies, hindering rapid iteration and feature development.

## 3.2 Proposed System

The proposed system enhances multiple server collaboration simultaneously and it has temporary database there will be no data breaching in this system.

## Key Features

- Modern frontend framework: Adopting a modern frontend framework such as React or Vue.js to enhance interactivity, responsiveness, and development efficiency.
- Component-based architecture: Implementing a component-based architecture to facilitate code reusability, maintainability, and scalability.
- Real-time communication: Integrating WebSocket technology for real-time messaging and collaboration, ensuring instant updates and seamless communication.
- Enhanced security measures: Implementing robust security measures, including encryption, authentication, and access control, to protect user data and privacy.

## Advantages

- Enhanced Interactivity: Utilizing modern frontend frameworks like React or Vue.js would enable the implementation of advanced interactivity features, such as real-time updates, dynamic content rendering, and seamless user interactions, resulting in a more engaging user experience.

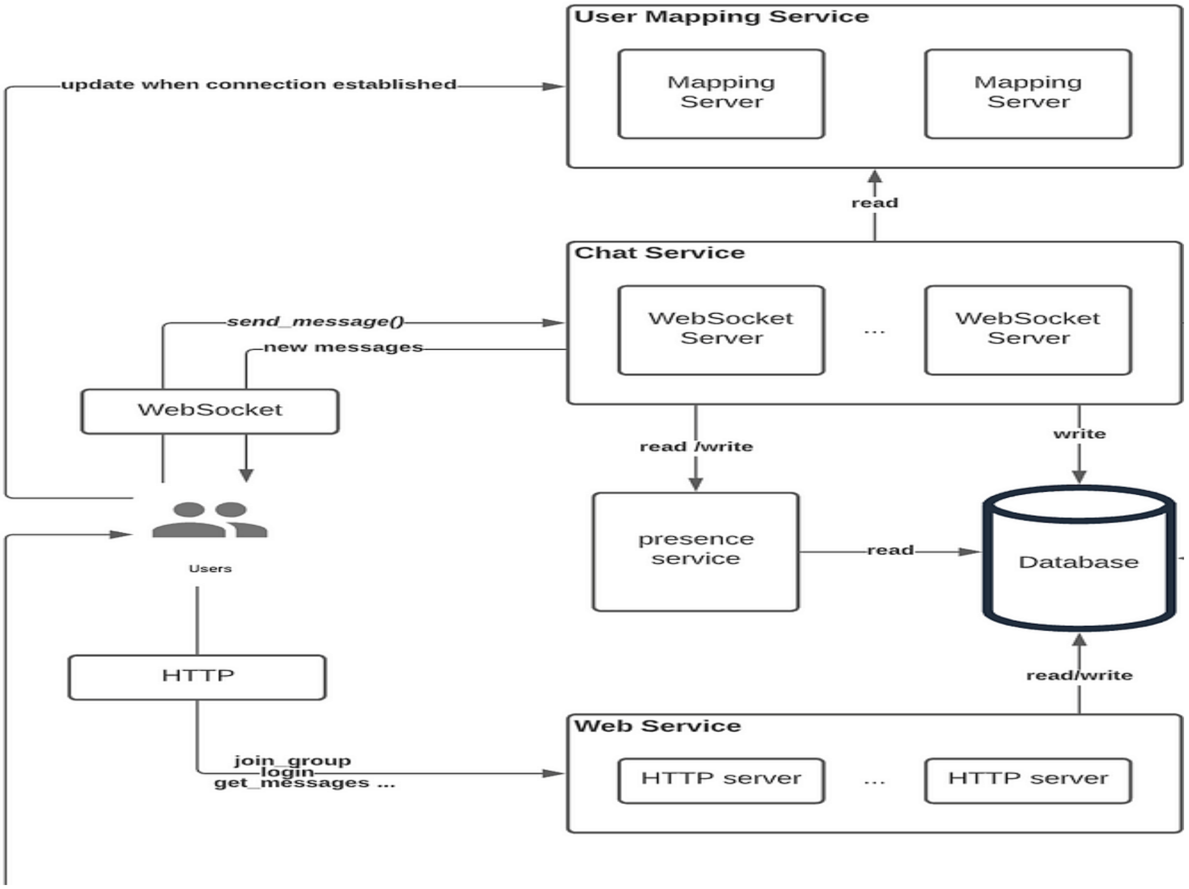
- Simplified Maintenance: Adopting a component-based architecture and modern development practices would streamline code organization and maintenance, making it easier to manage and update the frontend codebase as the application evolves.

- Improved Scalability: Modern frontend frameworks offer robust state management, performance optimization, and scalability features, allowing the application to scale more effectively to accommodate a growing user base or increased traffic. This ensures that the application remains responsive and performant even under high loads.

# 4. SYSTEM DESIGN

System design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. It involves translating user requirements into a detailed blueprint that guides the implementation phase.

## 4.1 Process diagram



The process of this project is followed as

- Users is joined and how the chat service works.
- Web socket acts as a server for all chat servives.
- The chat is read and write is handled by temporary database.
- The HTTP server maintains getting and join group messages.

## 4.2 Input Design

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

## Objectives

- Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.
- It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.
- When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

## 4.3 Output Design

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

- Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.
- Select methods for presenting information.
- Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

- Convey information about past activities, current status or projections of the
- Future.
- Signal important events, opportunities, problems, or warnings.
- Trigger an action.
- Confirm an action.

# 5. SYSTEM TESTING AND IMPLEMENTATION

## 5.1 System Testing

Software testing is a critical element of software quality assurance and represents the ultimate reviews of specification, design and coding. Testing represents interesting anomaly for the software. During earlier definition and development phases, it was attempted to build software from an abstract concept to tangible implementation.

The testing phase involves the testing of the developed system using various test data. Preparation of the test data plays a vital role in the system testing. After preparing the test data the system under study was tested using those test data. While testing the system, errors were found and corrected by using the following testing steps and corrections are also noted for future use. Thus, a series of testing is performed for the proposed system, before the system was ready for the implementation. The various types of testing done on the system are:

### Unit Testing

Unit testing is a crucial aspect of ensuring the reliability and functionality of the codebase. Unit tests are implemented to validate individual components, functions, or units of code in isolation, verifying that they behave as expected under various conditions.

### Integration Testing

Integration testing in Echoverse is essential for verifying the interactions and interoperability between different components, modules, and systems within the application. This testing phase evaluates how individual units work together as a cohesive whole, ensuring that they integrate seamlessly and function correctly in concert.

## Validation Testing

Validation testing in Echoverse is a critical phase aimed at ensuring that the software meets the requirements and expectations of its users. This testing process evaluates the application's functionality, usability, and performance in real-world scenarios to validate that it fulfills its intended purpose effectively.

## Functionality Testing

Functionality testing is a crucial aspect of ensuring that your project meets the specified requirements and delivers the intended features and capabilities to users. This testing phase focuses on verifying the correctness, completeness, and reliability of the system's functionalities.

## Security Testing

Security testing plays a crucial role in identifying and mitigating potential vulnerabilities, threats, and risks within your project's system. This testing phase focuses on evaluating the effectiveness of security controls, mechanisms, and countermeasures implemented to protect sensitive data, prevent unauthorized access, and ensure the confidentiality, integrity, and availability of information. Here's an in-depth look at security testing in your project:

## Output Testing

Output testing in Echoverse is essential for verifying the correctness and reliability of the system's outputs, such as chat messages, notifications, and user interface elements. This testing phase focuses on ensuring that the information presented to users is accurate, consistent, and aligned with the expected behavior of the application.

## 5.2 System Implementation

System implementation is the important stage of project where the theoretical design is turned into the practical system. The main stages in the implementation are as follows:

- Planning
- Training
- System testing and
- Changeover Planning is the first task in the system implementation.

Planning means deciding on the method and the time scale to be adopted. At the time of implementation of any system people from different departments and system analysis involve they are confirmed to practical problem of controlling various activities people outside their own data processing departments. The line managers will be controlled through an implementation coordinating committee. The committee considers ideas, problems and complaints of user department, it must also consider the following:

- The implication of system environment
- Self-selection and allocation implementation task
- Consultation with unions and resources available
- Standby facility and channels of communication

After finishing the development of any computer-based system the next complicated time consuming process is system testing. During the time to testing only the development company know that, how far the user requirement, have been met out

# 6. Conclusion

In Echoverse represents a transformative leap forward in the realm of online communication and collaboration. Through the integration of cutting-edge technologies such as Express.js, Socket.IO, HTML5, CSS3, and JavaScript, Echoverse delivers a seamless and immersive platform where users can engage in real-time conversations, seamlessly join servers, and explore diverse communities. By addressing the limitations of existing solutions and providing a unified environment for interaction, Echoverse aims to redefine online connectivity, fostering vibrant and engaging digital communities. With its intuitive design, responsive functionality, and commitment to user-centric innovation, Echoverse stands poised to revolutionize the way people connect, collaborate, and thrive in the digital age.

# 7. SCOPE FOR FUTURE ENHANCEMENT

These are some areas for potential future enhancements of the proposed system for Echoverse:

- **Advanced Collaboration Tools:** In the future, Echoverse could introduce advanced collaboration tools to enrich user experiences. This might include integrating features such as video and voice calling functionalities, allowing users to engage in real-time conversations with voice and video support.

- **Artificial Intelligence Integration:** The integration of artificial intelligence (AI) technologies holds promise for enhancing various aspects of Echoverse. For instance, personalized recommendations based on user preferences and behavior could enhance user engagement and content discovery within the platform.

- **Enhanced Security Measures:** To protect user data and privacy, Echoverse could implement enhanced security measures in future updates. This could include integrating additional authentication methods, such as two-factor authentication (2FA), to strengthen account security.

- **Accessibility Improvements:** Ensuring accessibility for all users is essential for Echoverse's inclusivity. Future enhancements could focus on improving accessibility features to accommodate users with disabilities.

# 8. APPENDIX

## 8.1 Bibliography

### Referred websites

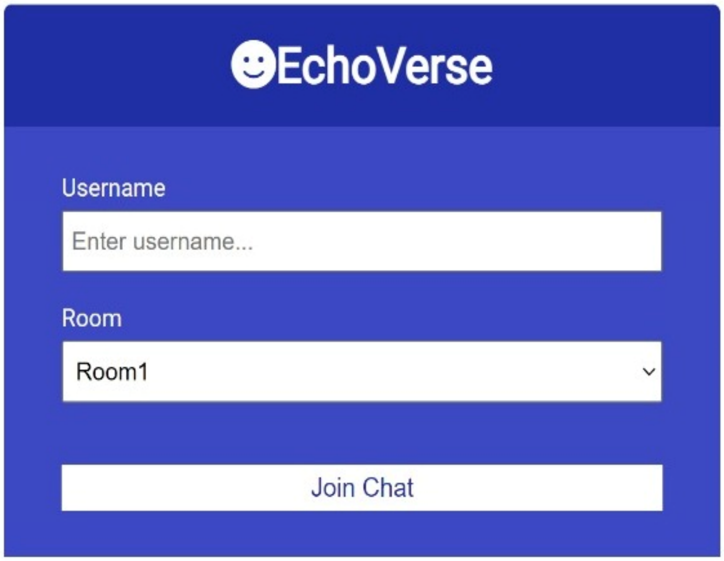
- https://socket.io/
- https://www.discordf.org/
- https://github.com/
- https://www.freecodecamp.org/

### Referred books

- Ramesh Bangia (2017) “Computer Vision: Algorithms and Applications”.
- Mario Casciaro (2014) “Node.JS Design Patterns”.
- Eric Sarrion (2022) “JavaScript from Frontend to Backend”.

## 8.2 Screenshots

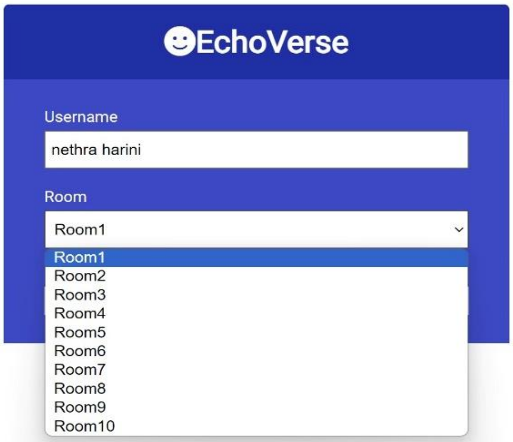
### Image-1



Room

Room1

## Image-2



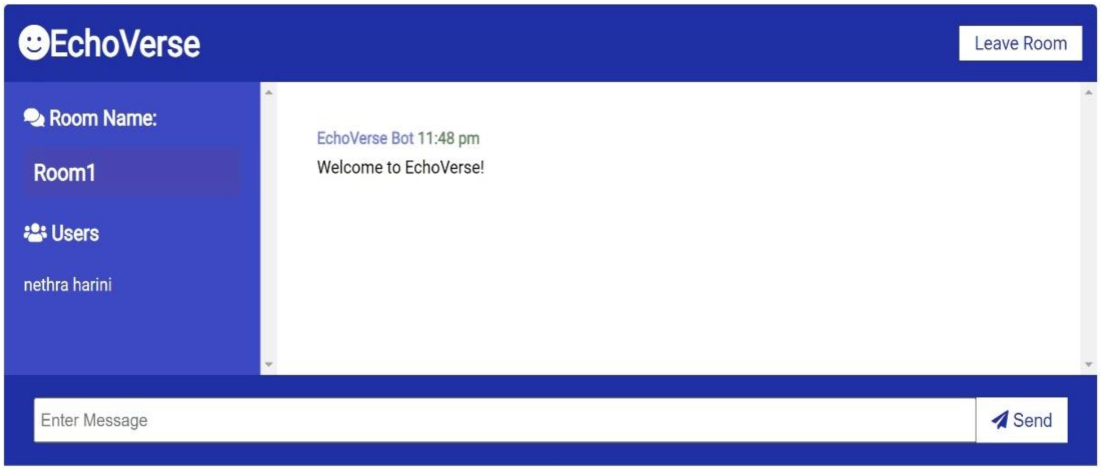
Username

nethra harini

Room

Room1

## Image-3



Room Name:

Room1

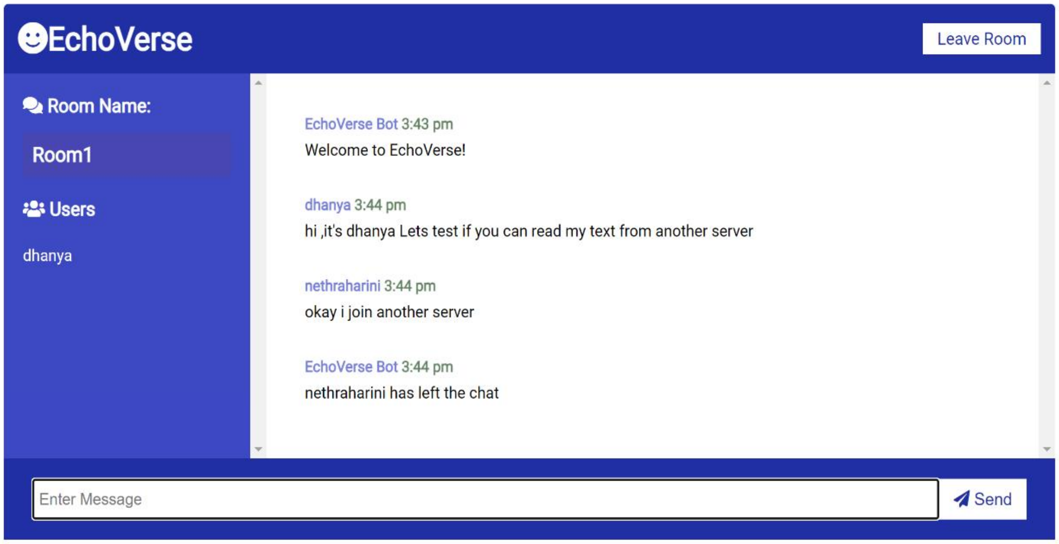
Users

nethra harini

EchoVerse Bot 11:48 pm

Welcome to EchoVerse!

## Image-4



Room1

dhanya

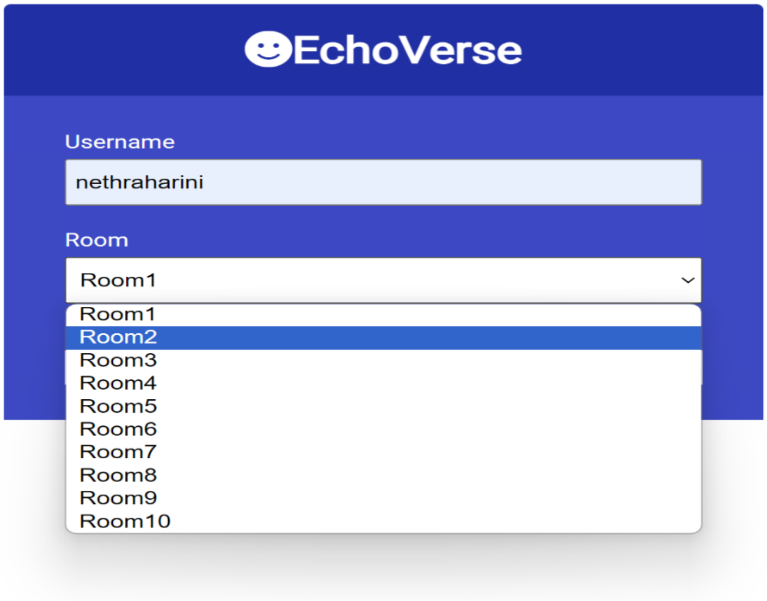
EchoVerse Bot 3:43 pm
Welcome to EchoVerse!

dhanya 3:44 pm
hi ,it's dhanya Lets test if you can read my text from another server

nethraharini 3:44 pm
okay i join another server

EchoVerse Bot 3:44 pm
nethraharini has left the chat

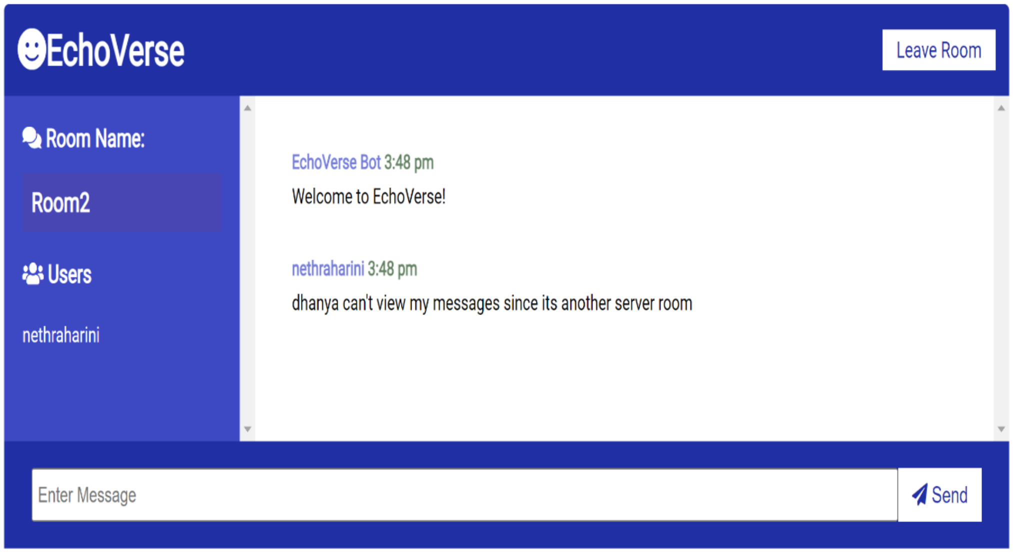
## Image-5



Username: nethraharini

Room: Room1

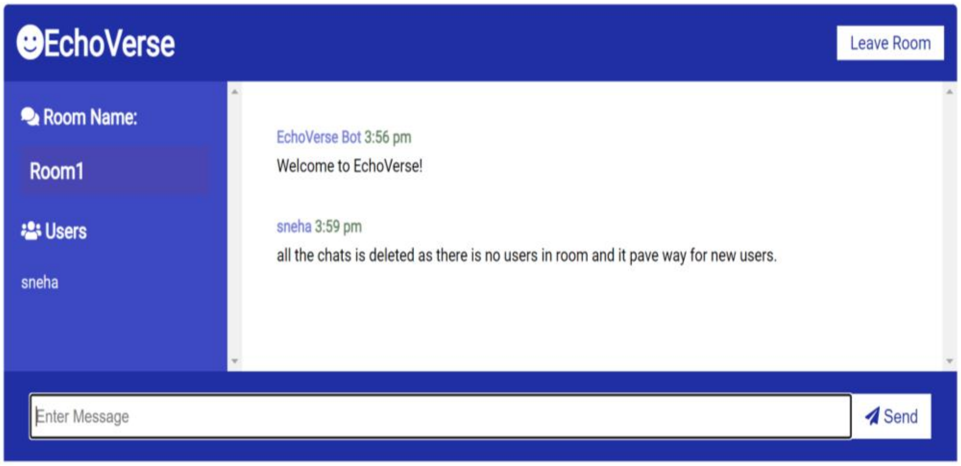
## Image-6



EchoVerse Bot 3:48 pm
Welcome to EchoVerse!

nethraharini 3:48 pm
dhanya can't view my messages since its another server room

## Image-7



EchoVerse Bot 3:56 pm
Welcome to EchoVerse!

sneha 3:59 pm
all the chats is deleted as there is no users in room and it pave way for new users.

### 8.3 Sample Code

```
const path = require('path');
const http = require('http');
const express = require('express');
const socketio = require('socket.io');
const formatMessage = require('./utils/messages');
const {
userJoin,
getCurrentUser,
userLeave,
getRoomUsers
} = require('./utils/users');
const app = express();
const server = http.createServer(app);
const io = socketio(server);

//Set static folder
app.use(express.static(path.join(__dirname, 'public')));


const botName = 'EchoVerse Bot'
// Run when client connect
io.on('connection', socket => {
socket.on('joinRoom', ({username, room}) => {
const user = userJoin(socket.id, username, room);
socket.join(user.room);

// Welcome current user
socket.emit('message',formatMessage(botName, 'Welcome to EchoVerse!'));
```

```
// Broadcast when a user connect
socket.broadcast.to(user.room).emit('message', formatMessage(botName,`${user.username}  has joined the chat`));


    // Send users and room info
io.to(user.room).emit('roomUsers', {
room: user.room,
users: getRoomUsers(user.room)

});
});
// Listen for chatMessage
socket.on('chatMessage', msg => {
const user =getCurrentUser(socket.id);
io.to(user.room).emit('message',formatMessage(user.username, msg));
});
// Runs when client disconnects
socket.on('disconnect', () => {
const user = userLeave(socket.id);

if(user){
io.to(user.room).emit('message',formatMessage(botName, `${user.username} has left the chat`));


// Send users and room info
io.to(user.room).emit('roomUsers', {
room: user.room,
users: getRoomUsers(user.room)

});
}
});

});
const PORT = 3000 || process.env.PORT;

server.listen(PORT, () => console.log(`Server running on port ${PORT}`));
```

**Messages.JS:**

```
const moment = require('moment');

function formatMessage(username, text) {
return{
   username,
   text,
   time: moment().format('h:mm a')
}
}

module.exports = formatMessage;
```

# ABSTRACT

Echoverse stands as a pioneering web platform,working for the online communication and collaboration. Its foundation lies in Express.js, ensuring streamlined server-side processing and routing for optial performance.  this is Socket.IO, seamlessly enabling real-time bidirectional communication between clients and servers, thus ensuring instantaneous message delivery.  With HTML providing the structural backbone, Echoverse boasts compatibility across devices and browsers, while CSS enhances its visual presentation with customizable styling options. JavaScript drives interactivity, facilitating dynamic features such as live chat, notifications, and user interactions. Through Echoverse, users seamlessly join servers, engage in real-time conversations, and explore diverse communities. By This modern web technologies, Echoverse is set to redefine online connectivity and interaction, and engaging digital community that works on collaboration and connection.

I am NETHRA HARINI .M (21BCA036) doing my final year Computer Applications in  PSG College of Arts & Science, batch (2021 – 2024)