

# **L. D. COLLEGE OF ENGINEERING**

**(Affiliated to Gujarat Technological University)**

**Navrangpura, Ahmedabad-380015**

A Mini Project Report

On

**“Peer 2 Peer”**

B.E. III, SEMESTER – VI

(COMPUTER ENGINEERING)

*Submitted by*

- |                         |                       |
|-------------------------|-----------------------|
| <b>1. Jeel Gajera</b>   | <b>(200280107080)</b> |
| <b>2. Ketan Lakum</b>   | <b>(200280107075)</b> |
| <b>3. Nitin Mujpara</b> | <b>(210280107526)</b> |

**Department of Computer Engineering**

**L. D. College of Engineering**

**Ahmedabad-380015**

**May-2023**

# **L. D. College of Engineering**

Department of Computer Engineering

2022-2023

## **CERTIFICATE**

**Date: 17/05/2023**

This is to certify that the mini project entitled “**Peer 2 Peer**” has been carried out by **Jeel Gajera (200280107080)**, **Ketan Lakum (200280107075)** and **Nitin Mujpara (210280107526)** under my guidance in partial fulfillment of the course **Advance Java Programming** in 6<sup>th</sup> Semester of Bachelor of Computer Engineering during the academic year 2022-23.

**Prof. Payal Prajapati**

Assistant Professor

Department of Computer Engineering

L. D. College of Engineering

## CANDIDATE'S DECLARATION

We have finished our mini project report entitled "**Peer 2 Peer**" and submitted to our respective guide. We are in 6<sup>th</sup> semester and we have tried to give our best. We have done our work honestly and in a good way.

First Candidate's

Name : Gajera Jeel  
Branch : Computer  
Enrollment No. : 200280107080  
Signature

Second Candidate's

Name : Lakum Ketan  
Branch : Computer  
Enrollment No. : 200280107075  
Signature

Third Candidate's

Name : Nitin Mujpara  
Branch : Computer  
Enrollment No. : 210280107526  
Signature

Submitted to: Prof. Payal Prajapati

L. D. College of Engineering,  
Ahmedabad.

Affiliated to: Gujarat Technological University

# INDEX

<b>Sr No</b>	<b>Title</b>		<b>Page No</b>
	<b>Acknowledgement</b>		5
	<b>Abstract</b>		6
<b>1</b>	<b>Introduction</b>		7
	1.1	Introduction to System	7
	1.2	Limitation of Existing System	8
	1.3	Objective of the new system	9
	1.4	Problem Definition	10
<b>2</b>	<b>Proposed Systems</b>		11
	2.1	Tools and Technology used	11
	2.2	Users and Roles	12
	2.3	Functionality/Modules	13
<b>3</b>	<b>Database Design Details</b>		14
<b>4</b>	<b>Project Result and Discussion</b>		15
<b>5</b>	<b>Future Enhancement</b>		19
<b>6</b>	<b>Conclusion</b>		20
	<b>References</b>		21

## ❖ Acknowledgement

We would like to express our sincere gratitude to all those who have contributed to the successful completion of this project. Their support, guidance, and expertise have been invaluable throughout the development process.

First and foremost, we would like to thank our project supervisor Prof. Payal Prajapati for their continuous support and valuable insights. Their guidance and supervision have been instrumental in shaping the direction of this project.

We are grateful to L. D. College Of Engineering for providing us with the necessary resources, facilities, and infrastructure to carry out this project. Their assistance has been crucial in ensuring the smooth execution of the project.

We would like to extend our appreciation to the developers and contributors of the advanced Java technologies used in this project. Their hard work and dedication have provided us with a solid foundation to build upon and create a robust and efficient web application.

We would also like to acknowledge the help and support of our fellow team members who have collaborated and worked together throughout the development process. Their teamwork, creativity, and problem-solving skills have been vital in overcoming challenges and achieving our goals.

Furthermore, we extend our thanks to the participants who volunteered their time and provided valuable feedback during the testing phase of the application. Their input and suggestions have greatly contributed to improving the functionality and user experience of the application.

Last but not least, we express our heartfelt gratitude to our family and friends for their constant encouragement and understanding during the project's development. Their support has been invaluable in helping us stay motivated and focused.

In conclusion, we would like to express our appreciation to all individuals and organizations who have contributed to the successful completion of this project. Their involvement has played a significant role in making this web application a reality.

## ❖ Abstract

The primary objective of this project is to develop an advanced web application utilizing Java technologies that empowers users to effortlessly search for and establish connections with other individuals, enabling seamless sharing of files and messages, all while eliminating the need for data storage on a centralized server. Leveraging a peer-to-peer communication model, this application enables direct transmission of files and messages from the sender to the intended recipient.

This web application encompasses a wide range of functionalities, including user registration and profile creation. Users have the ability to create personalized profiles that showcase their interests, skills, and other relevant information. Leveraging this profile data, users can connect with other individuals who share similar interests or possess complementary skills, fostering meaningful connections and collaborations.

The application also includes a robust messaging system that facilitates seamless communication between users. This messaging system enhances the user experience by providing a convenient and secure platform for exchanging messages and fostering real-time interactions.

Furthermore, the application incorporates a feature that allows users to maintain a comprehensive friend list based on their past connections. This feature enables users to easily manage their network of connections and stay connected with individuals who have played a significant role in their digital ecosystem.

By harnessing the power of advanced Java technologies, this web application aims to provide a user-friendly, secure, and efficient platform that empowers users to connect, collaborate, and share files and messages seamlessly, all while maintaining control over their own data

# 1. Introduction

## 1.1. Introduction to System

In today's digital age, the ability to easily connect and collaborate with others is crucial for individuals and businesses alike. The emergence of peer-to-peer communication models has revolutionized the way individuals interact and share information, eliminating the need for centralized data storage and providing greater control over personal data.

In this context, our team has developed an advanced web application utilizing cutting-edge Java technologies that allows users to connect and share files and messages directly with other users, without relying on a centralized server. Our application leverages a peer-to-peer communication model, which enhances user privacy and security by enabling direct transmission of data between users.

Our application's core features include user registration and profile creation. Leveraging this profile data, our application enables users to connect with other individuals who share similar interests or possess complementary skills, fostering meaningful connections and collaborations.

Furthermore, our application includes a robust messaging system that facilitates seamless communication between users, enabling users to exchange messages and foster real-time interactions. The messaging system is designed to provide a secure and convenient platform for users to communicate with each other while maintaining full control over their data.

Our application also includes a feature that allows users to maintain a comprehensive friend list based on their past connections, enabling them to easily manage their network of connections and stay connected with individuals who have played a significant role in their digital ecosystem.

In conclusion, our web application provides users with a user-friendly, secure, and efficient platform to connect, collaborate, and share files and messages seamlessly, all while maintaining control over their own data. By utilizing advanced Java technologies and leveraging a peer-to-peer communication model, our application sets a new standard for efficient and secure digital collaboration.

## 1.2.Limitation of Existing System

**Dependency on User Connectivity:** As our application relies on a peer-to-peer communication model, the seamless transmission of files and messages is contingent upon the connectivity and availability of both the source and destination users. In situations where either user experiences network issues or is offline, the real-time exchange of data may be hindered.

**Limited Functionality for Offline Users:** Since our application prioritizes direct communication between users, offline users may face limitations in accessing shared files and messages. Users who are temporarily disconnected from the network may miss out on important updates and information until they regain connectivity.

**Potential Security Risks:** While our application emphasizes user privacy and security by eliminating the need for centralized data storage, there may still be inherent security risks associated with peer-to-peer communication. Users must exercise caution and adopt secure practices to safeguard their data and protect against potential security breaches.

**Lack of Scalability:** The peer-to-peer communication model employed in our application may face scalability challenges when dealing with a large user base. As the number of users increases, the direct transmission of files and messages may become more complex and resource-intensive, potentially impacting the overall system performance.

**Limited Collaboration Features:** While our application provides a platform for users to connect and share files and messages, it may lack certain advanced collaboration features found in centralized systems. Collaborative functionalities such as real-time document editing or task management may be limited or absent in our current implementation.

**User Dependency for Profile Matching:** The success of establishing connections between users based on profile matching relies heavily on the accuracy and completeness of user-provided profile information. Inaccurate or incomplete profiles may result in ineffective connection recommendations, reducing the overall user experience.

It is important to recognize these limitations as areas for further improvement and development. Despite these limitations, our web application still provides a valuable solution for direct peer-to-peer communication and file/message sharing, emphasizing user control and privacy.



### 1.3.Objective of the new system

**Enable Direct Peer-to-Peer Communication:** The new system aims to facilitate direct communication between users, allowing them to connect and interact with each other without relying on a centralized server. By leveraging a peer-to-peer communication model, users can directly transmit files and messages, enhancing privacy, and reducing reliance on third-party storage.

**Seamless File and Message Sharing:** The new system aims to provide users with a seamless and efficient platform for sharing files and messages. Users will be able to exchange various types of files, such as documents, images, and videos, directly with other users, ensuring quick and secure transmission.

**User Registration and Profile Creation:** The new system will include a user registration process and profile creation functionality. Users will have the ability to sign up, create personalized profiles, and provide relevant information about their interests, skills, and preferences. This information will be utilized to facilitate meaningful connections and enhance the overall user experience.

**Enhanced Communication Features:** The new system will incorporate a comprehensive messaging system that enables real-time communication between users. Users will have access to features such as instant messaging, group chats, and notifications to foster effective and efficient communication.

**Robust Friend List Management:** The new system will provide users with the ability to manage their connections effectively through a comprehensive friend list feature. Users will be able to view and organize their connections based on past interactions, facilitating ongoing communication and collaboration.

**User-Friendly Interface and Experience:** The new system will prioritize a user-friendly interface, ensuring that users can easily navigate and utilize the application's features. A responsive and intuitive design will enhance the overall user experience, making it seamless for users to connect, share files, and communicate with others.

**Privacy and Security:** The new system will prioritize user privacy and security. By eliminating the need for centralized data storage, users will have greater control over their personal information. The system will incorporate robust security measures to protect user data during transmission and ensure the confidentiality and integrity of shared files and messages.

**Scalability and Performance:** The new system will be designed with scalability and performance in mind. It will aim to efficiently handle a growing user base, ensuring that the application remains responsive and reliable, even with increased usage and data transmission.

## **1.4.Problem Definition**

In today's digital landscape, there is a need for a web application that enables users to connect, share files, and communicate directly with others while maintaining control over their data. Existing systems often rely on centralized servers for data storage, which may compromise user privacy and pose security risks. Additionally, collaboration features and scalability limitations hinder the seamless exchange of files and messages.

To address these issues, our project aims to develop an innovative web application using advanced Java technologies. The objective is to create a peer-to-peer communication model where users can establish direct connections, share files and messages, and maintain control over their personal data. The application will also provide a user-friendly interface, robust friend list management, and enhanced communication features.

The development of this system poses several challenges. Firstly, ensuring a secure and reliable peer-to-peer communication model that eliminates the need for centralized data storage is crucial. Maintaining the privacy and integrity of user data during transmission is a key concern. Secondly, implementing efficient file and message sharing functionalities without relying on a central server requires careful design and optimization. Scalability should be considered to handle a growing user base and maintain optimal performance.

Furthermore, the project entails overcoming limitations such as potential connectivity issues between users and limited functionality for offline users. Addressing these challenges will enhance the user experience and ensure uninterrupted communication and file sharing capabilities. Additionally, providing comprehensive collaboration features and accurate profile matching based on user-provided information will contribute to the success of the system.

Overall, the problem at hand is to develop a web application that enables users to connect, share files, and communicate directly, while addressing the limitations of existing systems. By creating a peer-to-peer communication model, enhancing user control and privacy, and implementing robust features, our system aims to revolutionize the way individuals collaborate and share information in a secure and efficient manner.

## 2. Proposed Systems

### 2.1. Tools and Technology used

#### Backend:

- **Java:** The project utilizes Java as the primary programming language for the backend development. Java provides a robust and versatile foundation for building web applications, allowing for efficient and secure code implementation.
- **Spring Boot:** Spring Boot is utilized as a framework for developing the backend of the web application. It offers a streamlined development experience, simplifying the process of building and deploying Java-based applications.

#### Frontend:

- **React JS:** The frontend of the web application is developed using React JS, a popular JavaScript library for building user interfaces. React JS provides a component-based approach, enabling the creation of reusable and interactive UI components.
- **Tailwind CSS:** Tailwind CSS is used for styling and designing the frontend of the application. It offers a utility-first approach, providing a comprehensive set of pre-designed CSS classes that can be easily applied to create responsive and visually appealing interfaces.

#### Authentication:

- **Firebase:** Firebase is utilized for authentication purposes in the web application. Firebase offers a secure and easy-to-implement authentication system, allowing users to sign up, log in, and manage their accounts with features like email/password authentication, social media sign-in, and account management.

#### Database:

- **MongoDB:** The project employs MongoDB as the database management system. MongoDB is a NoSQL database that provides flexibility and scalability, making it suitable for handling unstructured data and accommodating future growth.

#### IDE:

- **VS Code:** Visual Studio Code (VS Code) is used as the Integrated Development Environment (IDE) for the project. VS Code is a lightweight and versatile code editor that offers various extensions and features, enhancing productivity and facilitating smooth development workflows.

By leveraging these tools and technologies, the project aims to achieve a robust, efficient, and user-friendly web application that enables seamless communication, file sharing, and user connections while maintaining privacy and security.

## 2.2.Users and Roles

The web application developed for this project encompasses multiple user roles, each with distinct functionalities and responsibilities. The following are the main user roles identified for the system:

### **Registered Users:**

- Registered users are individuals who have signed up for an account in the web application.
- They have the ability to create their profiles.
- Registered users can view their friend list for connect with other users based on profile info.
- They can exchange messages and share files directly with their connections using the peer-to-peer communication model.
- Registered users can also maintain a friend list, organizing their connections and staying connected with important contacts.

### **Unregistered Users:**

- Unregistered users are visitors who have not yet signed up for an account.
- However, unregistered users have restricted functionalities and cannot engage in direct communication, file sharing, or connection management until they create an account.

### **Administrator/Management:**

- The administrator or management role is responsible for overseeing and maintaining the overall functioning of the web application.
- They have administrative privileges and can manage user accounts, handle system configurations, and monitor user activities.

The defined user roles ensure that registered users can fully utilize the web application's features for communication, file sharing, and connection management, while unregistered users can access limited information. The administrator or management role ensures smooth operation and maintenance of the system.

## 2.3. Functionality / Modules

The web application developed for this project consists of several key functionalities or modules that enable users to connect, share files, and communicate effectively. The following modules have been identified:

### **User Registration and Authentication:**

- This module allows users to sign up for an account by simply providing the google Account.
- Profile creation enables users to showcase their details and helps in finding relevant connections.
- It ensures that only registered users with valid credentials can access the system and its features.
- Firebase authentication utilized to provide a secure and seamless authentication process.

### **User Connection:**

- This module enables users to connect with other registered users based on provide by simply others registered mail id.
- Users can view profiles info of potential connections and send connection requests to establish meaningful connections.

### **Direct Messaging:**

- The direct messaging module allows users to communicate with their connections through instant messaging.
- Users can exchange text messages, media files, and other types of information in real-time.
- The messaging module include features like conversation organization So, user can easily identify incoming msg flow. By alignment of text.

### **File Sharing:**

- This module enables users to share files directly with their connections.
- Users can upload and send various file types, such as documents, images, or videos.
- The file sharing module ensures secure and efficient transmission of files between users.

### **Friend List Management:**

- The friend list management module provides users with the ability to organize and manage their connections.
- Users can view their list of connections, categorize them, and access basic user information.

### 3. Database Design Details

#### Collection: Users

This collection stores user-related information, including user profiles and account details.

Fields:

**\_id:** A unique identifier for each user document (e.g., "645b3eeddaece92df691fda9").

**name:** The name of the user (e.g., "Jeel Gajera").

**email:** The email address associated with the user (e.g., "jeelgajera200@gmail.com").

**profileImg:** The URL or path to the user's profile image (e.g.,

"https://lh3.googleusercontent.com/a/AGNmyxMHf...").

**friendList:** An array field that stores the user's friend list, containing references to other user documents.

The above database design captures the necessary information to support the functionality of the application. The Users collection stores user details, including their name, email, profile image, and an array field for the friend list.

#### Peerdb.peer

5 DOCUMENTS 1 INDEXES

The screenshot shows the MongoDB Compass interface for the Peerdb.peer collection. At the top, there are tabs for Documents, Aggregations, Schema, Explain Plan, Indexes, and Validation. The Documents tab is active. Below the tabs, there is a search bar with a filter icon and a query input field containing "{ field: 'value' }". To the right of the search bar are buttons for Reset, Find, and More Options. Below the search bar, there are buttons for ADD DATA and EXPORT COLLECTION. The main area displays three documents in a list view. Each document is a JSON object with the following fields: \_id (ObjectId), name, email, profileImg, friendList (Array), and \_class (dev.ajava.peer2peer.model.PeerModel). The first document is for Jeel Gajera, the second for Prkalpa Dev, and the third for ketan lakum.

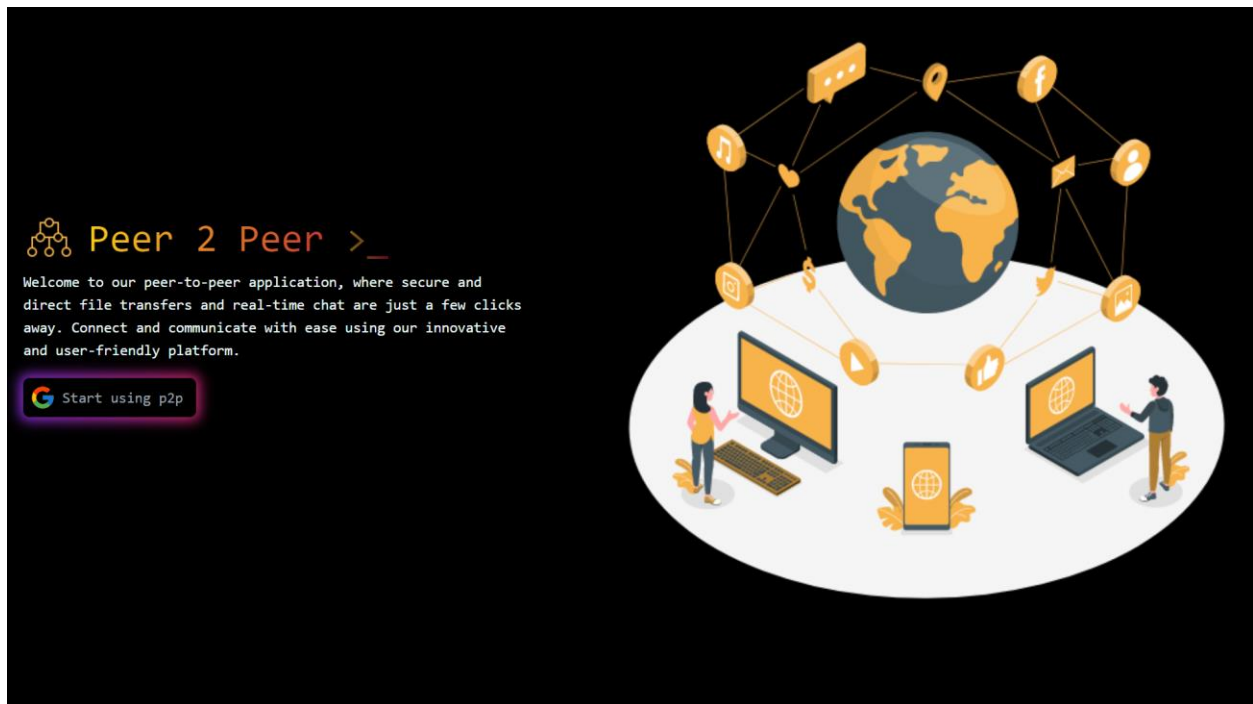
```
{
  "_id": ObjectId('645b3eeddaece92df691fda9'),
  "name": "Jeel Gajera",
  "email": "jeelgajera200@gmail.com",
  "profileImg": "https://lh3.googleusercontent.com/a/AGNmyxa9io-hYDoybXyxED0p20W1jhKMhf...",
  "friendList": Array,
  "_class": "dev.ajava.peer2peer.model.PeerModel"
}
```

```
{
  "_id": ObjectId('6460a9b52b426e092542eefa'),
  "name": "Prkalpa Dev",
  "email": "prakalpa.dev@gmail.com",
  "profileImg": "https://lh3.googleusercontent.com/a/AGNmyxaM7mHLIy9V2UKUFtmc-aBARgmLST...",
  "friendList": Array,
  "_class": "dev.ajava.peer2peer.model.PeerModel"
}
```

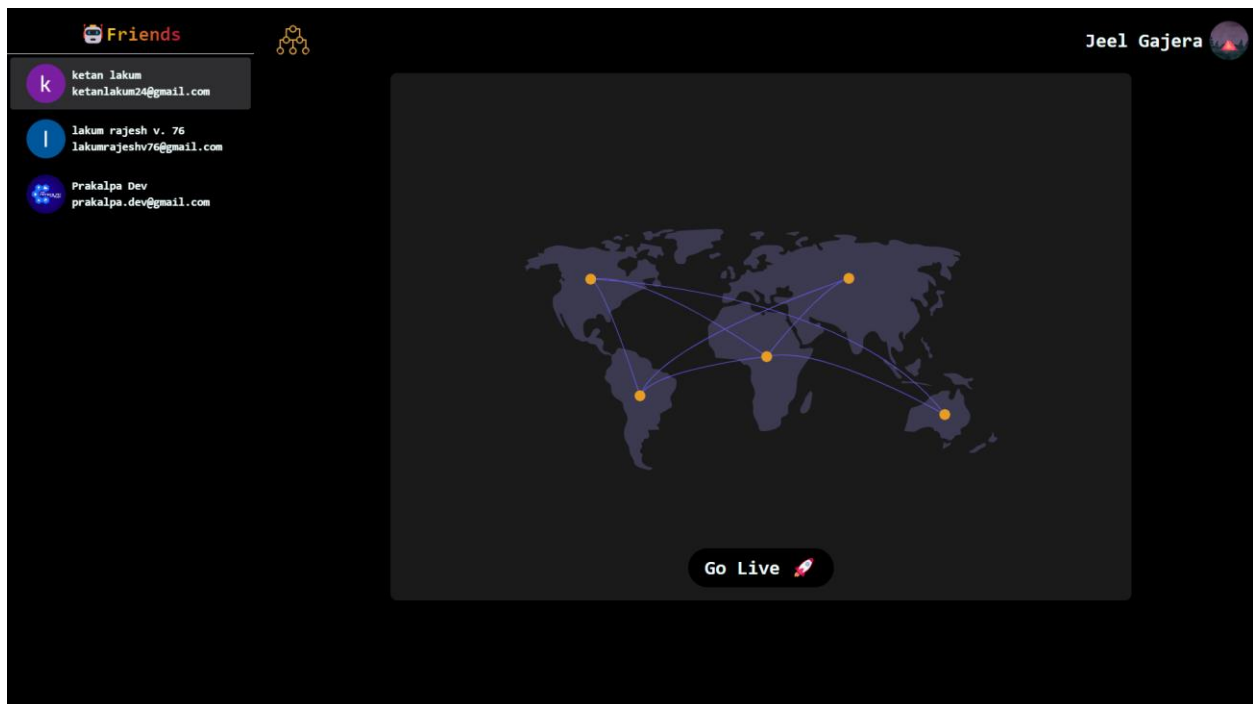
```
{
  "_id": ObjectId('6461cb81b3ba79447f3e3976'),
  "name": "ketan lakum",
  "email": "ketanlakum24@gmail.com",
  "profileImg": "https://lh3.googleusercontent.com/a/AGNmyxb4gqqaudkaPAxfy4mYhSAzbT5SFx...",
  "friendList": Array,
  "_class": "dev.ajava.peer2peer.model.PeerModel"
}
```

## 4. Project Result and Discussion

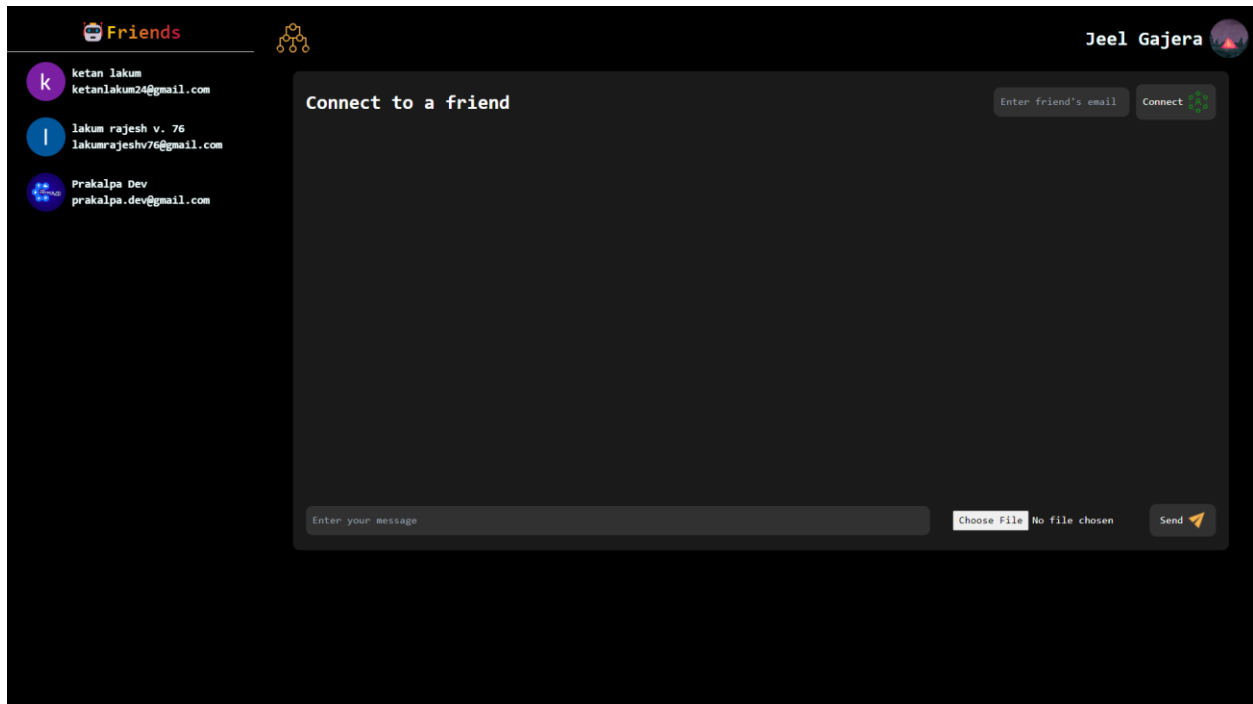
### ➤ Login Page



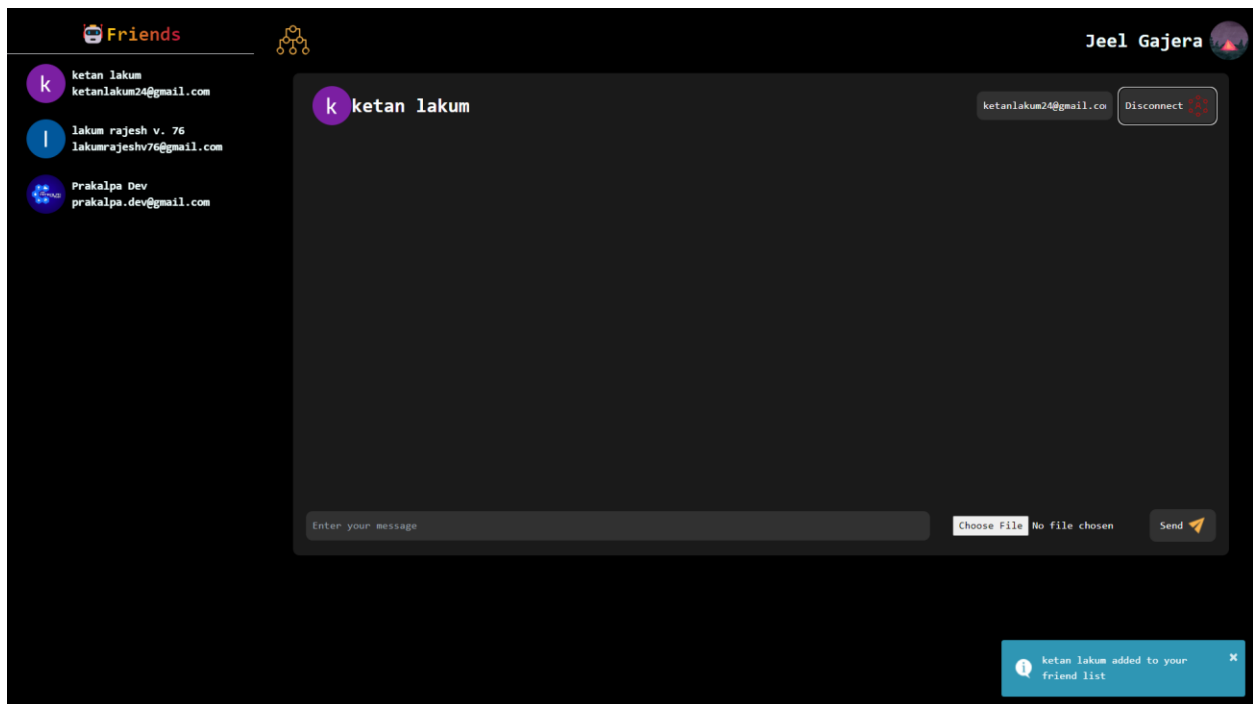
### ➤ Home Page



➤ Start User Connection by Click on GO LIVE



➤ Connect with Other by enter EMAIL



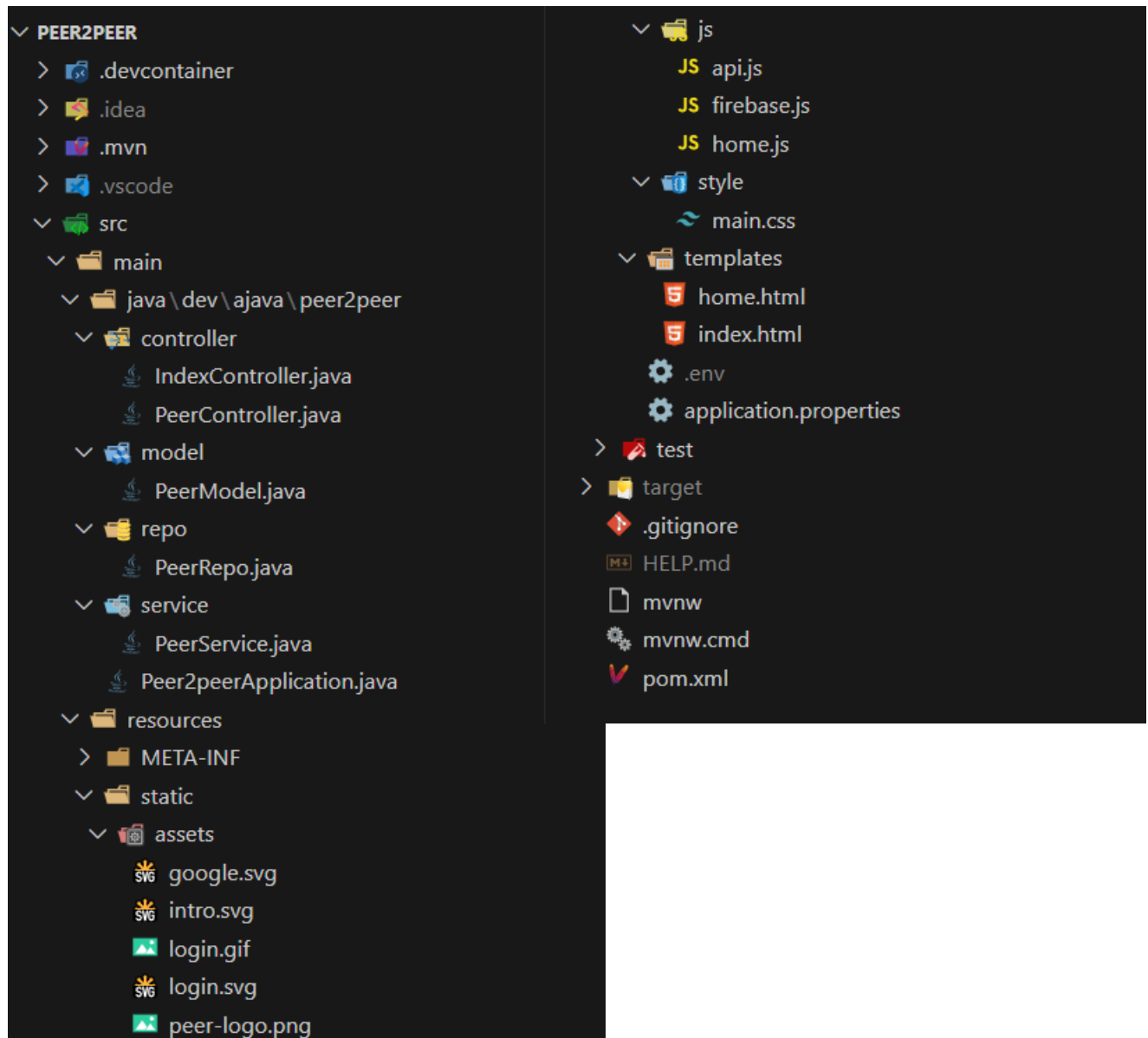


## ➤ Conversation demo

The screenshot shows a chat interface with a dark theme. On the left, a sidebar titled "Friends" lists three contacts: Prakalpa Dev (prakalpa.dev@gmail.com), ketan lakum (ketanlakum24@gmail.com), and lakum rajesh v. 76 (lakumrajeshv76@gmail.com). The main chat area is titled "Jeel Gajera" and shows a conversation with ketan lakum. The messages are: ketan lakum: "hii jeel", "please send me a practical list of AJava", "Thankyou"; Jeel Gajera: "hello ketan", "Sure, just a min", "Practical\_list\_AJP\_3160707\_2022-23-Even Term (1).pdf". At the bottom, there is a text input field "Enter your message", a "Choose File" button, and a "Send" button with a paper plane icon.

The screenshot shows the same chat interface from a different perspective. The sidebar "Friends" now lists ketan lakum (ketanlakum24@gmail.com), Ketan Lakum (ketanlakum72@gmail.com), lakum rajesh v. 76 (lakumrajeshv76@gmail.com), and Jeel Gajera (jeelgajera200@gmail.com). The main chat area is titled "ketan lakum" and shows a conversation with Jeel Gajera. The messages are: Jeel Gajera: "hii jeel", "please send me practical list of AJava", "Thank you"; ketan lakum: "hello ketan", "Sure, just a min", "Practical\_list\_AJP\_3160707\_2022-23-Even Term (1).pdf". At the bottom, the "Choose File" button now says "No file chosen".

## ➤ Project Structure



## 5. Future Enhancements

While the initial implementation of the web application covers the essential features, there are several potential areas for future enhancements and improvements. Some possible future enhancements for the project include:

**Group Messaging:** Introduce the capability for users to create and participate in group conversations. This enhancement would enable users to communicate and collaborate with multiple individuals simultaneously, facilitating teamwork and group discussions.

**Advanced Search and Recommendation System:** Enhance the user search functionality by incorporating advanced search algorithms and recommendation systems. This could involve utilizing machine learning techniques to suggest potential connections based on user profiles, interests, and past interactions.

**User Notifications:** Implement a notification system to keep users informed about new connection requests, messages, or updates related to their network. Notifications can improve user engagement and ensure timely responses to important activities within the application.

**File Versioning and Revision History:** Extend the file sharing functionality to include versioning and revision history. This enhancement would allow users to track changes made to shared files, revert to previous versions, and view a comprehensive history of modifications.

**Integration with Cloud Storage Services:** Integrate the application with popular cloud storage services, such as Google Drive or Dropbox, to provide users with the option to store and share files directly from their preferred cloud storage accounts.

**Real-time Presence Indicator:** Implement a real-time presence indicator to display the online/offline status of users. This feature can help users identify active connections and determine the best time to initiate communication.

**Advanced User Analytics:** Introduce analytics capabilities to gather insights on user behavior, interactions, and preferences. This data can be utilized to improve the user experience, personalize recommendations, and optimize system performance.

**Mobile Application:** Develop a mobile application version of the web application to extend its reach and accessibility to users on mobile devices. This would provide a seamless user experience and ensure compatibility across different platforms.

**Integration with Third-Party APIs:** Integrate the application with third-party APIs and services to enhance functionality. For example, integrating with social media platforms could allow users to import contacts or share updates from the application.

**User Privacy Controls:** Enhance user privacy controls by implementing granular settings that allow users to customize the visibility of their profile information, control message visibility, or restrict access to shared files.

## 6. Conclusion

In conclusion, the development of a web application with advanced Java technologies, utilizing a peer-to-peer communication model, provides a solution to the limitations of existing systems in terms of data privacy and security. By implementing features such as direct file sharing, messaging, and user connections, the application aims to revolutionize the way individuals collaborate and communicate.

The project's database design incorporates the necessary collections, such as Users to store user information and manage friend connections effectively. The use of technologies like Java, Spring Boot, React JS, Tailwind CSS, Firebase for authentication, and MongoDB as the database management system ensures a robust and efficient development process.

While the initial implementation covers the core functionalities, there are several avenues for future enhancements and improvements. These include implementing features like group messaging, advanced search algorithms, user notifications, file versioning, and integration with cloud storage services. Furthermore, incorporating real-time presence indicators, advanced user analytics, and developing a mobile application can enhance the user experience and accessibility.

Overall, the project's goal is to provide users with a secure, efficient, and user-friendly web application for connecting, sharing files, and communicating. By addressing the limitations of existing systems and considering future enhancements, the project aims to meet the evolving needs of users and provide a platform that fosters collaboration and seamless communication in a privacy-focused manner.

## 7. References

- Oracle. Java Documentation. Retrieved from <https://docs.oracle.com/en/java/>
- Spring Boot Documentation. Retrieved from <https://docs.spring.io/spring-boot/docs/current/reference/html/>
- ReactJS Documentation. Retrieved from <https://reactjs.org/docs/>
- PeerJS Documentations. Retrieved from <https://peerjs.com/docs/#api>
- Tailwind CSS Documentation Retrieved from <https://tailwindcss.com/docs>
- Firebase Documentation. Retrieved from <https://firebase.google.com/docs>
- MongoDB Documentation. Retrieved from <https://docs.mongodb.com/>
- Visual Studio Code Documentation. Retrieved from <https://code.visualstudio.com/docs>
- GitHub Repository for Source Code: <https://github.com/JeelGajera/peer2peer>