

PICTIONARY

Submitted in partial fulfillment of the requirements of the
degree

BACHELOR OF ENGINEERING IN COMPUTER ENGINEERING

By

Pratham Gawli 2213192

Vrajesh Hegde 2213209

Raghav Rathi 2113208

Supervisor

Prof. Vaibhav Ambhire



**Computer Engineering Department
Thadomal Shahani Engineering College**

Bandra(w), Mumbai - 400 050

University of Mumbai

(AY 2023-24)

CERTIFICATE

This is to certify that the Mini Project entitled “ **Pictionary** ” is a bonafide work of **Pratham Gawli (2213192), Vrajesh Hegde (2213209), Raghav Rathi (2113208)** submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of “**Bachelor of Engineering**” in “**Computer Engineering**”.

(Prof. Vaibhav Ambhire)

Supervisor

(Prof. Dr Tanuja Sarode)

Head of Department

(Prof. Dr. G.T. Thampi)

Principal

Mini Project Approval

This Mini Project entitled “ **Pictionary** ” By **Pratham Gawli (2213192)**, **Vrajesh Hegde (2213209)**, **Raghav Rathi (2113208)** is approved for the degree of **Bachelor of Engineering in Computer Engineering**.

Examiners

1.....
(Internal Examiner Name &Sign)

2.....
(External Examiner name &Sign)

Date:

Place:

Contents

Abstract	ii
Acknowledgments	iii
List of Figures	iv
1 Introduction	1
1.1 Introduction	1
1.2 Motivation	1
1.3 Problem Statement & Objectives	2
1.4 Organization of the Report	2
2 Literature Survey	3
2.1 Survey of Existing System	3
2.2 Limitation Existing system or research gap	4
2.3 Mini Project Contribution	5
3 Proposed System (eg New Approach of Data Summarization)	6
3.1 Introduction	7
3.2 Architecture/ Framework	9
3.3 Algorithm and Process Design	9
3.4 Details of Hardware & Software	11
3.5 Experiment and Results	11
3.6 Conclusion and Future work	15
References	16

Abstract

The 'Pictionary' project is an interactive multiplayer gaming platform built using the Flutter framework for the client-side and Node.js, Express, Socket.io, and MongoDB for the server-side. It offers an engaging and real-time gaming experience where users can create rooms, invite others to join, and participate in drawing and guessing games. The game encompasses a wide array of features, including a waiting lobby for player gatherings, a collaborative canvas for doodling, options to customize pen width and color, and the ability to clear the canvas. Random word generation, chat functionality within the rooms, and dynamic turn-switching mechanics further enhance the gaming experience. The project's server component leverages Socket.io to facilitate instant communication between players and a MongoDB database, using Mongoose for data storage. It manages essential game aspects like tracking correct word guesses, scoring, determining winners, and updating leaderboards. This comprehensive documentation serves as a vital resource for developers, offering insights into the project's architecture, implementation details, and customization options, making it accessible for potential contributors to enrich the gaming experience and adapt it for diverse applications.

Acknowledgement

We would like to express our gratitude and thanks to **Prof. Vaibhav Ambhire** for his valuable guidance and help. We are indebted for his guidance and constant supervision as well as for providing necessary information regarding the project. We would like to express our greatest appreciation to our principal **Dr. G.T. Thampi** and head of the department **Dr. Tanuja Sarode** for their encouragement and tremendous support. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of the project.

Pratham Gawli

Vrajesh Hegde

Raghav Rathi

LIST OF FIGURES

Sr No.	Figure No.	Name of Figure	Page No.
1	3.1	Block Diagram	9
2	3.2	Main Screen	11
3	3.3	Create Room Screen	12
4	3.4	Join Room Screen	12
5	3.5	Waiting Lobby Screen	13
6	3.6	Playing Area Screen	13
7	3.7	Ongoing Game	14
8	3.8	Leaderboard Screen	14

Chapter 1

Introduction

1.1 Introduction

Project Pictionary is a real-time multiplayer game developed using Flutter for the client-side and Node.js with Socket.io and MongoDB for the backend. It enables users to create and join game rooms where players take turns drawing randomly generated words while others guess. The game features interactive doodling, chat, scoring, and leaderboards. This documentation provides a comprehensive guide to set up and understand the project, including the architecture, features, and server-client communication, allowing developers to contribute or customize the game for various applications.

1.2 Motivation

The motivation behind the development of our Pictionary project stems from the desire to modernize and enhance the traditional drawing and guessing game experience. Traditional board games have always been a source of entertainment and social interaction, but they often require physical presence and can be limited in scope. In the digital age, there is a growing need for engaging and accessible gaming experiences that transcend physical boundaries. Pictionary, as a beloved draw and guess game, provided the perfect opportunity to leverage modern technologies like Flutter, MongoDB and Node.js to bring this classic game to a wider audience.

By developing a digital version of Pictionary, we aim to make the game more accessible, interactive, and enjoyable for players regardless of their physical location. This project also serves as an exploration of the capabilities of these technologies, highlighting their potential to transform traditional games into immersive digital experiences. Additionally, the global shift towards online entertainment and social interaction, particularly in light of recent events, underscores the relevance and timeliness of such a project. Ultimately, our motivation is to provide a platform that

fosters creativity, connection, and entertainment in a digital age while showcasing the possibilities of modern software development.

1.3 Problem Statement & Objectives

In the realm of online gaming, Pictionary aims to provide a virtual platform where players can participate in a draw and guess game seamlessly. This project addresses the need for accessible and engaging entertainment in the digital age, eliminating the need for physical presence and offering a convenient and fun experience for players. The primary objectives of this project are:

- To develop a digital version of the Pictionary game using modern technologies such as Flutter, MongoDB, and Node.js.
- To create an intuitive and user-friendly interface that allows players to draw, guess, and interact effortlessly.
- To offer a platform for players to enjoy Pictionary from anywhere, fostering creativity and social connections.
- To showcase the potential of these technologies in transforming traditional games into engaging digital experiences.

1.4 Organization of the Report

This report consists of three chapters. The first chapter deals with introduction of the topic, problem statement, motivation behind the topic and objectives. The second chapter is the Literature Survey. It includes all the research work done related to this topic. All information related to study of existing systems as well as learning of new tools is mentioned in this chapter. The third chapter is about the proposed system which is used in this project. The block diagram, techniques used, hardware and software used screenshots of the project are presented in this chapter. All the documents related to development of this project are mentioned in References.

Chapter 2

Literature Survey

This chapter explains the concepts used in this project, study of existing system and contribution of this project

2.1 Survey of Existing System

Our project team conducted a survey of existing digital drawing and guessing games to gain insights into the current landscape of similar applications.

Drawasaurus

It is a platform where artistic users with creative minds can collaborate and make different types of artwork for fun. Drawasaurus has an intuitive interface that responds seamlessly on every device, Android and iOS mobile phones and PC. This makes the platform easily accessible, and the tools are easy to use. Drawasaurus requires that participants guess the image while a player is drawing. And any participant that guesses correctly wins points for each correct guess. Users can join any of the existing chat rooms by choosing the desired nickname. There is also the option to create new chatrooms on Drawasaurus. When a user decides to create a new room, the user can choose the room privacy, the number of players allowed, room name, and room access password. Users can also set the room custom words, rounds per game, drawing time, and the number of word choices for the room. The link to a newly created room can then be shared to invite participants. Drawasaurus offers players tools such as draw, erase, fill, size, and color tools, and all these features are available for free.

Pictar.io

Pictar.io is a web-based gaming site for art-loving people. Users can join the platform by creating an account with the desired nickname, and other details, to unlock the full features. Players can also opt to play a game on Pictar.io as guests, but being a guest player gives them limited access.

Pictar.io is a drawing platform as well as a multiplayer puzzle video game platform. And it features superb gameplay with many features. The platform offers a chat system that allows users to send instant messages during gameplay to other players in a room. The chat system also allows users to write the word while watching the object that other players will draw during their turn. To become a winner, a player has to be the first to guess the word correctly, and such a player will receive in-game points. Other tools and features on Pictar.io include the ability to create private chat rooms and invite players, family members, and friends to participate. There are other prominent features such as mouse-driven controls, 2D graphics, and more.

Draw & Guess

Draw & Guess offers lots of impressive Pictography games for users to enjoy. Players of Draw & Guess have access to artistic tools on the platform that they can use to draw diagrams of all sorts. The goal of the game is to earn points by correctly guessing the diagram a player is drawing. There are so many tools and features on Draw & Guess that make the platform unique and fun to use. These features include the ability to create custom personalized avatars to make the player stand out among the group. It also has the classic game mode, with each gamer having a guessed word from a list of words. At the end of each round, all players vote, and the player with the most votes wins. Players can also activate other modes such as multi-color mode, speed mode, and the invisible mode which is the hardest mode. The invisible mode requires players to draw while the strokes turn invisible. There is no limit to how many games can be played on Draw & Guess, players are able to compete to win every day. The platform also provides players with different brushes for drawing, such as pens, spray brushes, and paint buckets. The interface is optimized to display perfectly on all screen sizes, and the games will play seamlessly on all devices.

2.2 Limitation of existing system

- 1) There are some limitations in Drawasaurus :
 - Limited Features: While Drawasaurus offers free drawing tools and a chat room for gameplay, it may lack the complexity and variety of tools that more dedicated graphic design or drawing software provides.
 - Mobile-Centric: It mentions Android and iOS mobile compatibility, but this may not be suitable for users who prefer working on larger screens with more advanced tools.

- User Base: The success of such platforms heavily depends on the active user base, which can sometimes be limited, affecting the availability of games and players to interact with.
- 2) There are some limitations in Pictar.io:
- Limited Features for Guest Users: Guest players on Pictar.io have restricted access to the platform's features. This can be frustrating for users who want to explore the platform before committing to creating an account.
 - Dependency on User-Generated Content: The quality of gameplay and word choices can depend on user-generated content, which may vary in quality and creativity.
 - Potential for Inactive Rooms: If users create private chat rooms and do not actively invite participants, it may result in inactive rooms or limited engagement.
- 3) There are some limitations in Draw & Guess :
- Limited Complexity: While it provides a fun and engaging drawing and guessing experience, it may not cater to professional artists or those seeking advanced drawing tools.
 - Dependency on User Participation: The enjoyment of the platform relies on having active players and participants. If there are too few users, it can limit the quality of the gaming experience.
 - Lack of Seriousness: Some users might find the platform's games less suitable for serious art or design work, as it is primarily focused on casual gaming.

2.3 Mini Project Contribution

Our Pictionary project seeks to address these limitations by providing a user-friendly and customizable digital drawing and guessing game. The key contributions of our project include:

- Intuitive User Interface: We have designed a straightforward and user-friendly interface to ensure players of all skill levels can enjoy the game.
- Diverse Gameplay Elements: Our Pictionary game incorporates a range of features and game modes to keep players engaged and entertained.

- Customization Options: We offer players the ability to customize game settings, including difficulty levels and drawing tools, providing a tailored gaming experience

Chapter 3

Proposed System

This chapter consists of detailed description about the methodology used, the hardware and software components, the tools used and also the screenshots of the project.

3.1 Introduction

Flutter :

Flutter is an open-source UI development framework developed by Google to build apps. It enables us to build cross-platform apps using a single codebase. A single codebase can build smooth, high-performance natively compiled applications for iOS and Android quickly.

Flutter provides a simple, powerful, efficient, and straightforward SDK for developing mobile applications in Google's language, Dart. Dart is an open-source, Object-Oriented programming language with C-style syntax used to power Flutter apps.[1]

Node JS:

Node.js is an open-source, cross-platform, JavaScript runtime environment built on the V8 JavaScript engine of Google Chrome. It allows developers to write server-side JavaScript code, which runs outside of the web browser. Node.js provides an event-driven, non-blocking I/O model that makes it efficient and scalable for building network applications. This architecture allows Node.js to handle a large number of concurrent connections with low overhead and high throughput.

Node.js is commonly used for building web applications, real-time chat applications, and server-side APIs. It comes with a large set of built-in modules, such as HTTP, HTTPS, and FileSystem, that make it easy to develop server-side applications. Node.js also has a large and active community of developers who contribute to the ecosystem of modules and tools. One of the advantages of Node.js is its ability to work with the JavaScript language on both the server-side and client-side, providing a unified

programming language for the full stack. This allows developers to share code between the front-end and back-end, simplifying development and maintenance.[2]

Socket.io:

Socket.IO is an event-driven library for real-time web applications. It enables real-time, bi-directional communication between web clients and servers. It consists of two components: a client, and a server. Both components have a nearly identical API.

Socket.IO is also a protocol, where different complying implementations of the protocol can communicate with each other. The main implementation consists of two parts: a client that runs in the browser and a server for Node.js. Apart from the main implementation, there are multiple implementations, for example, the official Deno (JavaScript), C++, Java , and Swift servers.

Socket.IO primarily uses the WebSocket protocol with polling as a fallback option, while providing the same interface. Although it can be used simply as a wrapper for WebSockets, it provides many more features, including broadcasting to multiple sockets, storing data associated with each client, and asynchronous I/O.

It can be installed with the Node Package Manager (NPM).[6]

MongoDB:

MongoDB is an open-source document-oriented database that is designed to store a large scale of data and also allows you to work with that data very efficiently. It is categorized under the NoSQL (Not only SQL) database because the storage and retrieval of data in the MongoDB are not in the form of tables.

The MongoDB database is developed and managed by MongoDB.Inc under SSPL(Server Side Public License) and initially released in February 2009. It also provides official driver support for all the popular languages like C, C++, C#, and .Net, Go, Java, Node.js, Perl, PHP, Python, Motor, Ruby, Scala, Swift, Mongoid. So, that you can create an application using any of these languages. Nowadays there are so many companies that used MongoDB like Facebook, Nokia, eBay, Adobe, Google, etc. to store their large amount of data.[3]

3.2 Architecture

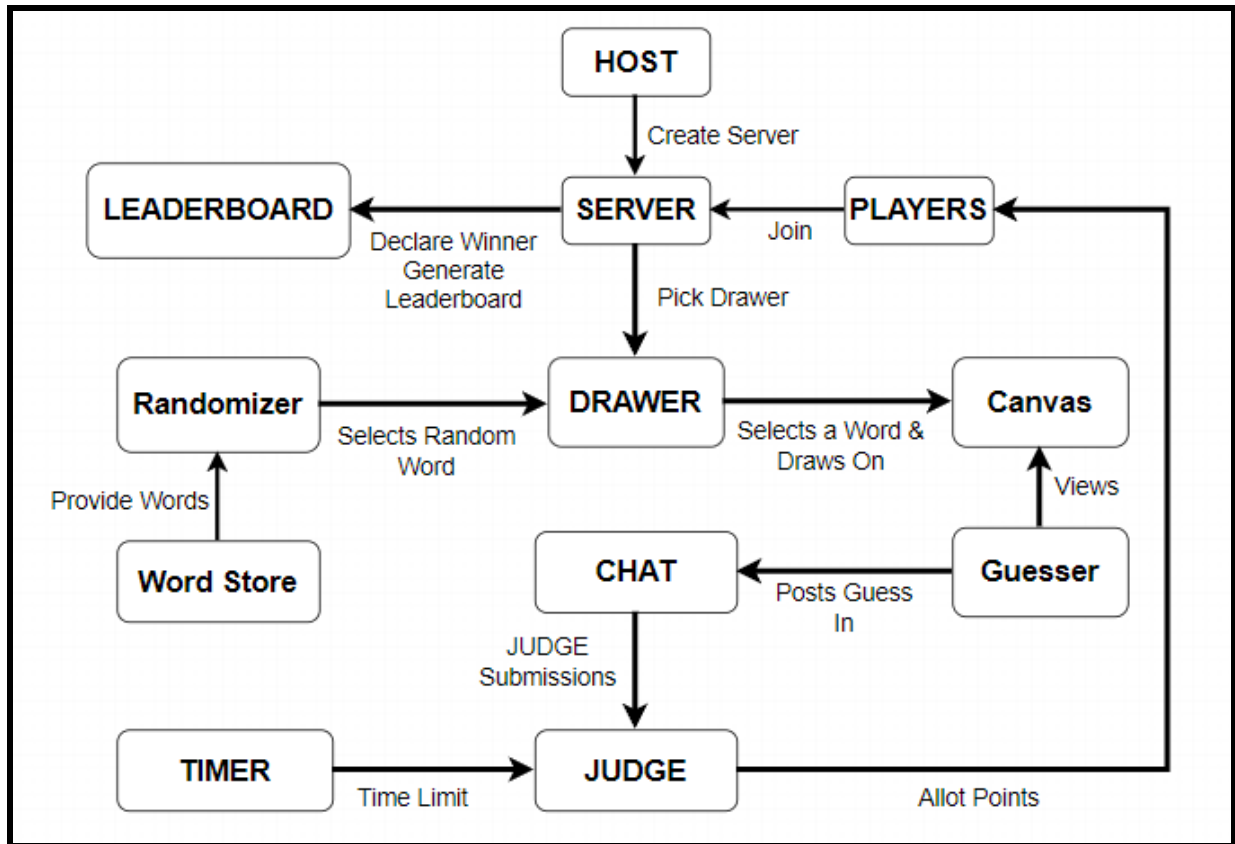


Fig 3.1 Block Diagram

The user can create the live server and make room for other players to join and play. The server chooses a sketcher who will be provided a random word which they need to draw on the canvas. The canvas is live and visible to all the players and others need to guess the word. The randomizer is responsible to select a random word from the word store. Participants enter their guesses in the chat which are judged by the judge which takes into account the timer and allots points to players accordingly. The judge then manages the leaderboard according to the points and then declare winner after all rounds finish.

3.3 Algorithm and Process Design

The development of the Pictionary project was guided by a structured algorithm and a carefully planned process design to ensure a systematic and efficient development journey. The key phases of this algorithm and process design are outlined below:

1. Problem Statement Formulation:

Our project's inception involved extensive discussions among team members and an in-depth analysis of existing drawing and guessing games. This collaborative effort led to the formulation of a precise problem statement that defined the project's scope and objectives.

2. Framework and Requirements Assessment:

Once the problem statement was established, the team deliberated on selecting the most appropriate technologies and frameworks for our project. We considered the group's familiarity with these technologies and their suitability for fulfilling the project's requirements. Ultimately, we chose to leverage Flutter for the front-end and Node.js for the back-end to deliver a dynamic and cross-platform gaming experience.

3. Tool and Technology Selection:

In addition to technologies and frameworks, we carefully selected the tools and software that would facilitate our development process. Tools like Visual Studio Code and various web browsers for testing and debugging were chosen to ensure a smooth development workflow.

4. Feature Finalization:

With a clear project overview and wireframes in place, our team collectively identified and finalized the features to be integrated into the application. These features were designed to enhance the gameplay and overall user experience of Pictionary.

5. Development Phase:

To expedite development, we divided the project into modules, with each group member taking responsibility for specific components. This approach allowed us to harness the diverse skills and expertise of our team members effectively.

6. Testing and Issue Resolution:

Rigorous testing procedures were implemented to identify and resolve any issues within our modules. Regular communication and collaboration among team members were crucial for successful issue resolution.

7. Evaluation and Documentation:

In the final stages of the project, we documented every aspect of our development journey. This documentation includes details on the project's development, testing, and outcomes, providing a comprehensive record of our work

3.4 Details of Hardware & Software

Sr. No.	Name of Equipment	Specification
1	Desktop with Windows 10 & above	4 GB RAM
2	Flutter	3.10.6
3	MongoDB	7.0
4	Node.js	18.18.0
5	Socket.io	4.7.2
6	VS Code	1.83.1

3.5 Results

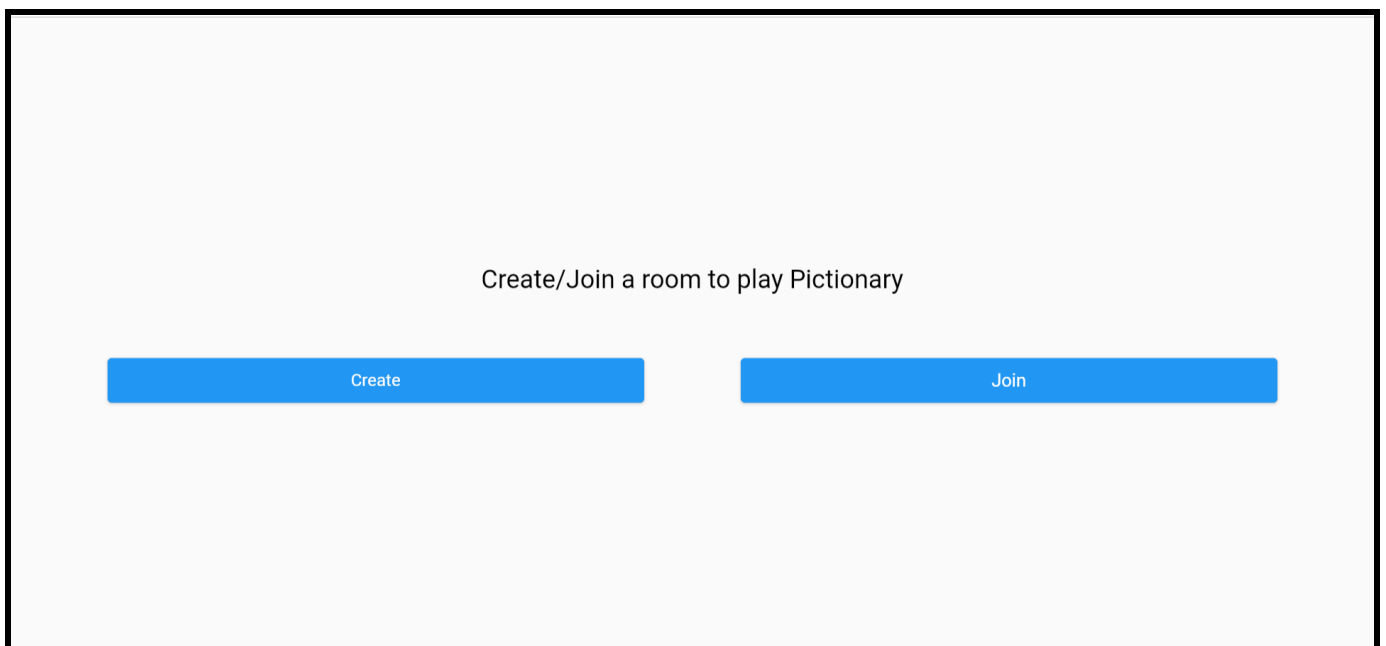
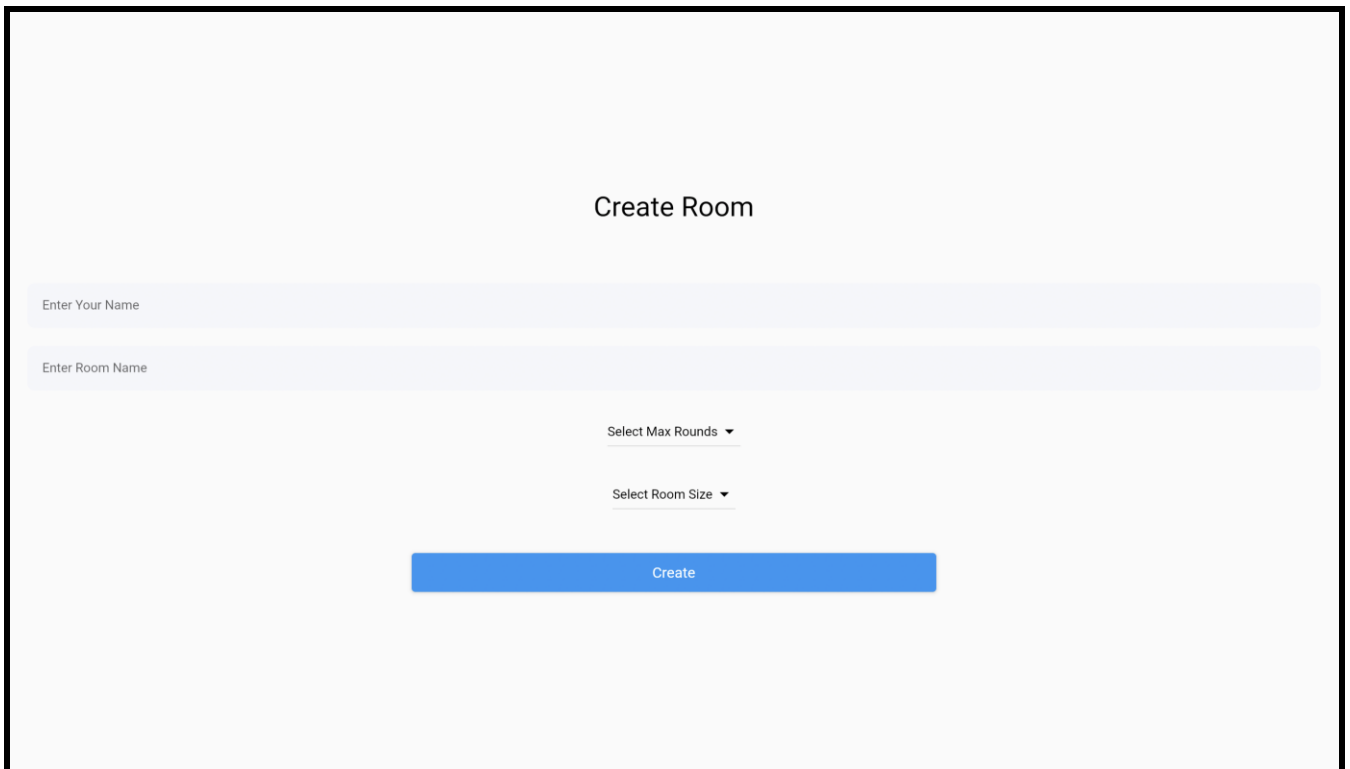


Fig 3.2 Main Screen

The Fig 3.2 shows the main screen where the user will first interact with the application. The user here can create a new room or choose to enter already created room.

The image shows a web interface for creating a room. At the top, the title "Create Room" is centered. Below it, there are two text input fields: "Enter Your Name" and "Enter Room Name". Under these fields, there are two dropdown menus: "Select Max Rounds" and "Select Room Size". At the bottom, there is a blue button labeled "Create".

Create Room

Enter Your Name

Enter Room Name

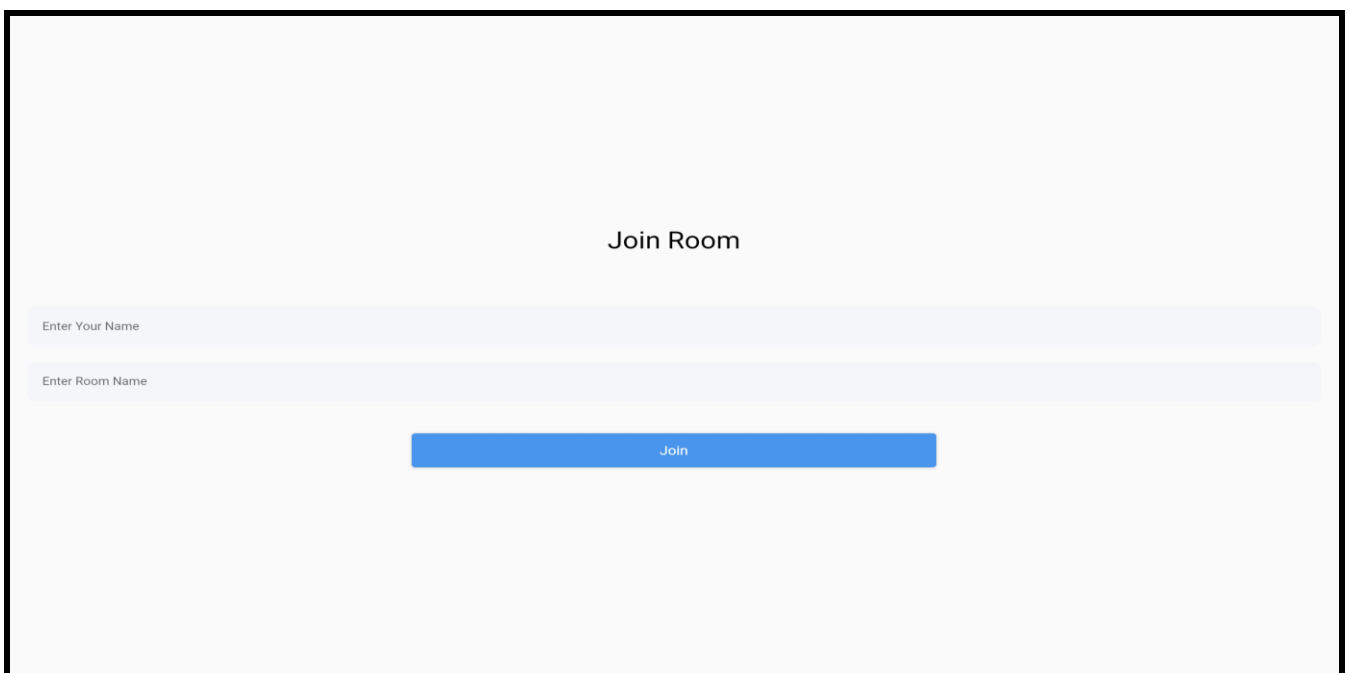
Select Max Rounds ▼

Select Room Size ▼

Create

Fig 3.3 Create Room Screen

The fig 3.3 is the Create Room Screen where the user can create a new room. The user will enter its name and select a room name. The select the number of players and number of rounds he wishes the game to go on.

The image shows a web interface for joining a room. At the top, the title "Join Room" is centered. Below it, there are two text input fields: "Enter Your Name" and "Enter Room Name". At the bottom, there is a blue button labeled "Join".

Join Room

Enter Your Name

Enter Room Name

Join

Fig 3.4 Join Room Screen

The fig 3.4 is the Join Room Screen where the user can join existing room using the room name.

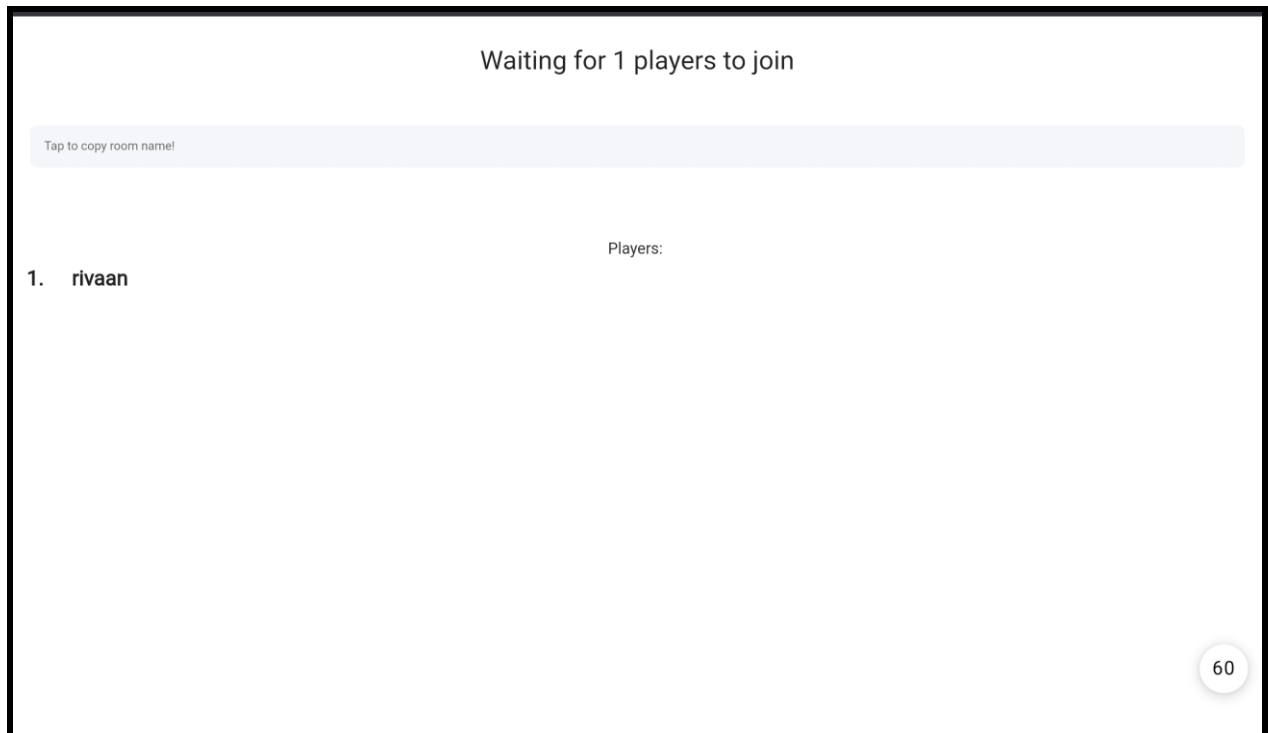


Fig 3.5 Waiting Lobby Screen

The fig 3.5 is the Waiting Lobby Screen where players wait while more players are yet to join the room.

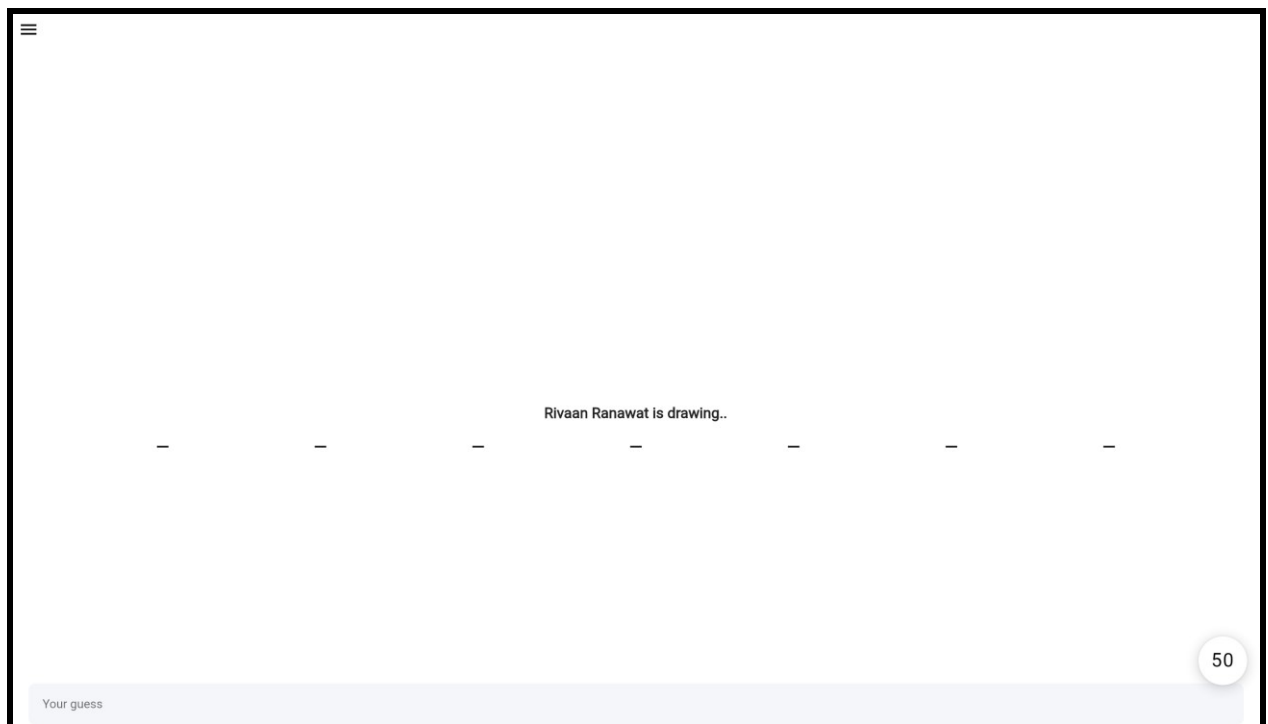


Fig 3.6 Playing Area Screen

The fig 3.6 is the Playing Area Screen where user has access to live canvas and chat area to enter guesses and also see the timer and number of letters in the word.

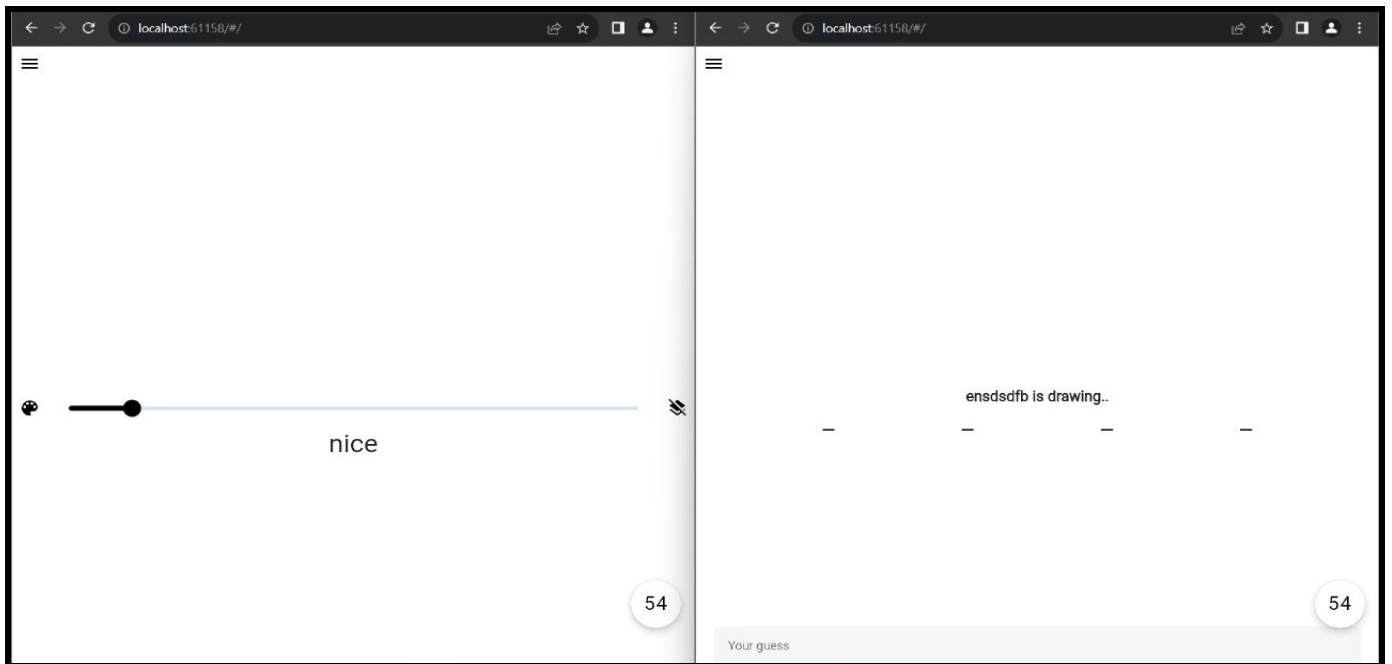


Fig 3.7 Ongoing Game

The fig 3.7 shows an ongoing game between two players where the word to guess is “nice”.

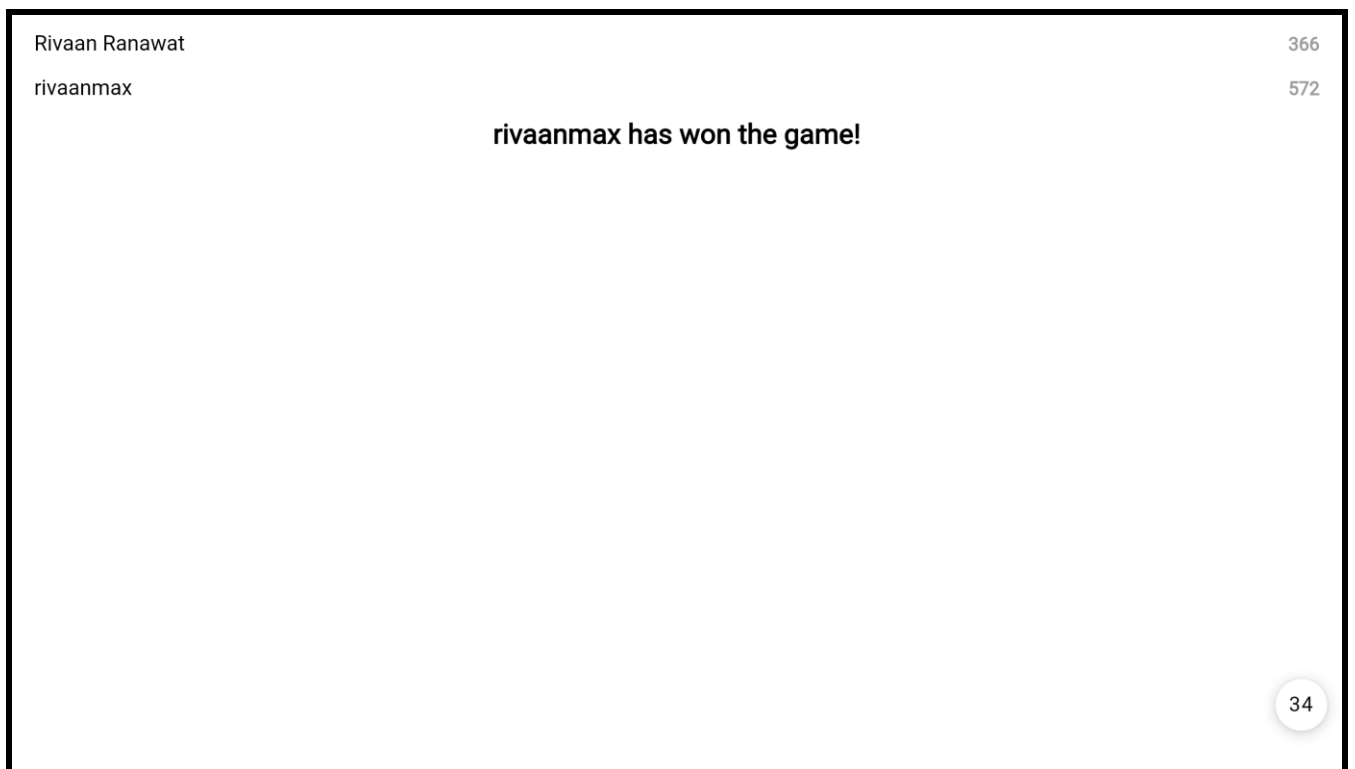


Fig 3.8 Leaderboard Screen

The fig 3.7 shows the leaderboard screen where the rankings can be seen.

3.6 Conclusion and Future Work

The Pictionary project has successfully realized the vision of creating an engaging and interactive digital drawing and guessing game, providing users with an enjoyable gaming experience.

Our project's core objective was to bridge the gap between the classic board game and the digital world, fostering creativity, teamwork, and fun among players.

Through the collaborative efforts of our team, we have not only developed a captivating game but also gained valuable insights into modern app development, UI/UX design, and teamwork dynamics.

The initial phase of our Pictionary project being concluded, we envisioned several exciting avenues for future development and enhancement:

- **Multiplatform Accessibility:** Expanding the game's compatibility to various platforms, including web browsers and desktop applications, will enable a broader audience to enjoy Pictionary.
- **Community and Social Integration:** Incorporating social features like player profiles, leaderboards, and real-time multiplayer interactions will enhance the social aspect of the game.
- **Advanced Gameplay Features:** We can introduce advanced gameplay features, such as themed drawing categories, tournament modes, and interactive challenges, to keep players engaged and entertained.
- **User-Generated Content:** Allowing users to create and share their custom word lists, drawings, and game modes will foster a vibrant and creative Pictionary community.

References

- [1] “All about Flutter | Felexiple” : <https://flexiple.com/flutter/deep-dive>
- [2] “About | Node.js” : <https://nodejs.org/en/about/>
- [3] “What is MongoDB - Working and Features – GeeksforGeeks” :
<https://www.geeksforgeeks.org/what-is-mongodb-working-and-features/>
- [4] “MongoDB: The Developer Data Platform” : <https://www.mongodb.com/>
- [5] “Introduction | Socket.IO” : <https://socket.io/docs/v4/>
- [6] “Wikipedia | Socket.IO” : <https://en.wikipedia.org/wiki/Socket.IO>