PROFORMA FOR THE APPROVAL PROJECT PROPOSAL

PNR **No.: 2020016400776276** Roll no**:** **730**

1. Name of the Student

   Siddhi Santosh Kotre

2. Title of the Project

   Bears & Bulls

3. Name of the Guide

   Miss. Pooja Amin

4. Teaching experience of the Guide_____

5. Is this your first submission? Yes ☐ No ☐

Signature of the Student Signature of the Guide

Date: …… Date: …….

Signature of the

Coordinator Date:

……

BEARS & BULLS

A Project Report
Submitted in partial fulfillment of the
Requirements for the award of the Degree of

BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)

By

Ms. Siddhi Santosh Kotre

A730

Under the esteemed guidance of

Miss Pooja Amin

DEPARTMENT OF INFORMATION TECHNOLOGY

CHIKITSAK SAMUHA'S

S.S & L.S PATKAR COLLEGE OF ARTS & SCIENCE & V. P. VARDE COLLEGE OF COMMERCE & ECONOMICS.

An Autonomous College

Affiliated To University Of Mumbai

Goregaon (W), Mumbai – 400 062

CHIKITSAK SAMUHA'S

S.S & L.S PATKAR COLLEGE OF ARTS & SCIENCE & V. P. VARDE COLLEGE OF COMMERCE & ECONOMICS.

An Autonomous College

DEPARTMENT OF INFORMATION TECHNOLOGY

CERTIFICATE

This is to certify that the project entitled, **"Bears & Bulls"**, is bonafide work of **Siddhi Santosh Kotre** bearing Seat.No: **730** submitted in partial fulfillment of the requirements for the award of degree of BACHELOR OF SCIENCE in INFORMATION TECHNOLOGY from University of Mumbai.

Internal Guide **Coordinator**

External Examiner

Date: **CollegeSeal**

COMPANY CERTIFICATE

(ONLY FOR LIVE PROJECTS)

IF APPLICABLE

ABSTRACT

We can see how word games may have effortlessly evolved, challenging individuals's language ability or expertise. Because games are an excellent pastime, they have probably transformed and have been a vital part of competitions and academic outlets later.

The core ideology behind this project is to build a word guessing game, which could be a good brain tease and create an itch to expand our vocabulary. This project is a simple but prudent game to check the intuition, pattern recognition and tactics of a player in guessing the word in the least amount of time.

As technology continues to flourish, word games progress hand-in-hand with it. Ever since we had the knowledge about language, whether to read, write, speak or understand, we have played around with words and today we have created similar games for enjoying such leisure activities as our project also has multiplayer mode so we can engage with other people and to play with them as the outcome is not predictable when you play against different player.

ACKNOWLEDGEMENT

This project would not have been successful without the ceaseless cooperation of our teachers. Their constant guidance and encouragement has been my biggest motivation. I am utterly grateful to Patkar Varde College for bestowing upon me this opportunity to bring forth my craft and competency.

This endeavor would not have been possible without my project guide Miss. Pooja Amin, her enlightening mentoring and constructive suggestions were helpful throughout the project. I would also like to thank all my colleagues and family members for their suggestions through different stages of my project.

I would like to extend my deepest regards to our Principal and CEO of Patkar Varde College, Dr. Mala P. Kharkar for presenting such a favor and liberty to work on my memorandum. I would also like to extend my welcome to our department coordinator Mrs. Namrata Kawale Shinde for her inveterate support. I would be remiss in not mentioning all the teaching and non teaching staff for their kind approach, counseling words of wisdom and inspiring to strive forward.

I feel blessed to have such strong and erudite forces guiding me throughout my entire journey as your help made all the difference.

DECLARATION

I hereby declare that the project entitled, "**Bears&Bulls**" done at **Patkar Varde College**, has not been in any case duplicated to submit to any other university for the award of any degree. To the best of my knowledge other than me, no one has submitted to any other university.

The project is done in partial fulfillment of the requirements for the award of degree of **BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)** to be submitted as a final semester project as part of our curriculum.

Name and Signature of the Student

Date :

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1 : INTRODUCTION

1. Theoretical Background

The core idea behind this project is to prompt a vigorous learning process and the players must study and enrich their vocabulary on their own time to succeed at these challenges. There have been similar games like Hangman and Scrambles but each with their own set of rules and protocols.

As long as there are words to form, there will be ways to play with them. Bears & Bulls is a word guessing game where you have to deduce the n-lettered word and if the word you have guessed has similar letters to the superword on the server we have acquired n-bears and if the position of the letter is same we have acquired n-bulls. We have to deduce and strategize meaningful words and if the postulated word matches the superword we have won the game.

2. Objectives of the Project

Few of the objectives are listed below:

- To enhance one's cognitive skills.
- To improve concentration and memorising ability.
- To help boost brain connection and memory.
- To expand our vocabulary.
- To promote self competition.
- To identify pattern formation

3. Purpose, Scope & Applicability of the Project

a. Purpose

- The intent of this project is to encourage healthy word-play games which can boost competition and a bit exasperating as word guessing games can be a simple but backbreaker to master and it also helps to calm their mind unconsciously so when the challenge is over their daily worries don't seem so daunting anymore.

b. Scope

- This game is user-friendly and also for all age groups as one can never fathom the possibilities of the words as it is an ever increasing universe and new words get added to the dictionary on a daily basis. Although sometimes it can get into two completely opposite scenarios wherein it can become addictive to the player or after playing it for long enough people start losing their interest. Still it will always help us to learn and to hype up our brains.

c. Applicability

- Bears & Bulls can be used as a model to help students in their English classes to learn vocabulary in a more thought-provoking way as it will be a productive as well as a fascinating way to learn. Teachers can make groups and give clues to students to help them solve and strategize the game which would be beneficial for their cognitive development.

4. Expected Achievements

- Even overlooking the fact about how much fun and entertaining they are, the advantages of playing these games are quite remarkable. It would be great if this game peaks the interest of all age groups and could be accessible to everyone. This game should be accessible to anyone on any platform and without many dependencies. I aspire that people will play this game and it will prove to be a resource of learning. “Getting 1% better everyday might not be noticeable but it will be notable” - James Clear, Atomic Habits

CHAPTER 2 : SURVEY OF TECHNOLOGIES

1. Description of Available Technologies

The following are the technologies which will be used in development of this project.

1. Python

Python is a programming language that permits you to work quickly and integrate systems more efficiently. Python is vastly Versatile, Easy to read, learn and write. Python's clean object-oriented design provides enhanced process control, and the language is equipped with excellent text processing and integration capabilities, as well as its own unit testing framework, which makes it more efficient.

2. SQLite

SQLite is a C-language library that implements a small, fast, self-contained, high-reliability, full-featured, SQL database engine. SQLite is the most used database engine in the world. SQLite is built into all mobile phones and most computers and comes bundled inside countless other applications that people use every day.

3. PyCharm IDE

PyCharm is a popular Integrated Development Environment (IDE) used for Python development. It is developed by JetBrains and is available in both free and paid versions. PyCharm provides many features that help developers write, test, and debug Python code more efficiently.

4. JavaScript, HTML, CSS

JavaScript is the programming language, we use HTML to structure the site, and we use CSS to design and layout the web page.

2. Comparative Analysis of Technologies in Chosen Area

A descriptive analysis of the languages is provided in the below table.

Table 2.1 Comparative Analysis between Technologies

Parameters	Java	Python	Kotlin
Ease of use	Hard	Simple	Simple
Performance	High Level	High Level	Moderate Level
Popularity	High Level	High Level	Moderate Level
Scalability	Moderate Level	High Level	High Level
Community Support	High Level	High Level	Moderate Level
Cross-Platform Support	Supports limited platforms	Support multiple platforms	Support multiple platforms
Documentation	Good but easy to understand	Good but easy to understand	Good but hard to understand
Library	Large	Huge	Limited

3. Chosen Project Domain

The domain of my project is Full-Stack Web development as it is to equip learners with the unique skills they need to build database-backed APIs and web applications. Full stack development supports cross platform optimization and uses planned codes and produce specifics to incessantly boost the software.

4. Technologies to be used

a. Front End

- HTML
- CSS
- JavaScript

b. Back End

- Python

c. Framework

- Django

d. Other Development Tools

- SQLite
- Pycharm IDE

5. Reason Supporting the use of above selected technologies

HTML provides the structure of the web page.CSS allows styling and presentation of the page. JavaScript provides interactivity and dynamic behavior on the page. Python is a flexible and powerful programming language, well suited for back-end development. PyCharm IDE is a popular and feature-rich Integrated Development Environment (IDE) for Python development. SQLite is a lightweight and simple relational database that is often used for small and local projects. Django is a high-level Python web framework that makes it easier and faster to build web applications. It provides a lot of built-in functionality, such as an ORM for database interaction, user authentication and authorization, URL routing, and more.

Together, these tools provide a complete solution for web development, from front-end to back-end, and from development to deployment.

CHAPTER 3 : REQUIREMENTS & ANALYSIS

1. Problem Statement and Problem Definition

Helping students to develop a strong reading vocabulary requires more than having them look up words in a dictionary. Especially for the people whose first language isn't English.
One of the most effective ways for young children to gain knowledge and develop crucial abilities is through play. So, if you're in search of some fun and educational activities to do at home, word games for kids are the perfect option to keep them engaged.

This game also makes learning at home enjoyable and lays the foundation for successful school learning.The intent of this project is to improve self-competition, as word-play games can get highly competitive and a bit exasperating on not getting the desired outcome and it is also friendly to all age groups.

2. Requirements Specification

What is requirement analysis?

The requirements should be documented, actionable, measurable, testable, traceable, related to identified business needs or opportunities, and defined to a level of detail sufficient for system design.

a. Functional Requirements

In software engineering and systems engineering, a functional requirement defines a function of a system or its component, where a function is described as a specification of behavior between outputs and inputs.

Functional requirements are as follows :

- Register
- Login
- Word generation on start of a new game
- Validation of input words
- Logical Implementation of pseudocode
- Leaderboard

b. Non-functional Requirements

Non functional requirements include beautification of user interface and ease of use of the game and it should run ceaselessly without unnecessary intervention.

In systems engineering and requirements engineering, a non-functional requirement (NFR) is a requirement that specifies criteria that can be used to judge the operation of a

system, rather than specific behaviors. They are contrasted with functional requirements that define specific behavior of functions.

Non-functional requirements are as follows :

- Compatibility
- Security
- Usability
- Recoverability
- Performance
- Availability
- Maintainability

c. User Requirements

What is the user requirement?

User requirements are as follows :

- Easy to understand protocols
- Friendly environment for all age groups
- Clear and concise
- Versatile

d. Hardware Requirements

What are hardware requirements?

Hardware requirements are as follows :

- 8 GB RAM
- Intel Core i5 processor

e. Software Requirements

What are software requirements?

Software requirements are as follows :

- Pycharm IDE
- Python3 or higher
- HTML, CSS, JavaScript
- SQLite
- Windows 8 or higher

3. Feasibility

A well-designed feasibility study should provide a historical background of the business or project, a description of the product or service, accounting statements, details of the operations and management, marketing research and policies, financial data, legal requirements and tax obligations. Generally, feasibility studies precede technical development and project implementation.

a. Operational Feasibility

It explains how well the proposed project solves the problem, how it takes into account the advantages and how it satisfies the requirements specified.

- Affordability
- Producibility
- Sustainability
- Usability
- Reliability

b. Technical Feasibility

The technical feasibility assessment is focused on gaining an understanding of the present technical resources of the organization and their applicability to the expected needs of the proposed system.

The technical requirements for my project include a working computer system which supports application development, internet to download all the necessary modules and prerequisites and sufficient amount of storage. All these are satisfied and readily available with me and hence this project is technically feasible.

c. Economic Feasibility

The purpose of an economic feasibility study (EFS) is to demonstrate the net benefit of a proposed project for accepting or disbursing electronic funds/benefits, taking into consideration the benefits and costs to the agency, other state agencies, and the general public as a whole.

All the modules and applications required for this application development are open source and available for free download hence the project is very economically feasible. Though we can also buy the professional edition which provides additional plugins for Eclipse IDE which is roughly 750 Rs and also windows licensing which is approximately 1000 Rs per month.

4. Planning and Scheduling

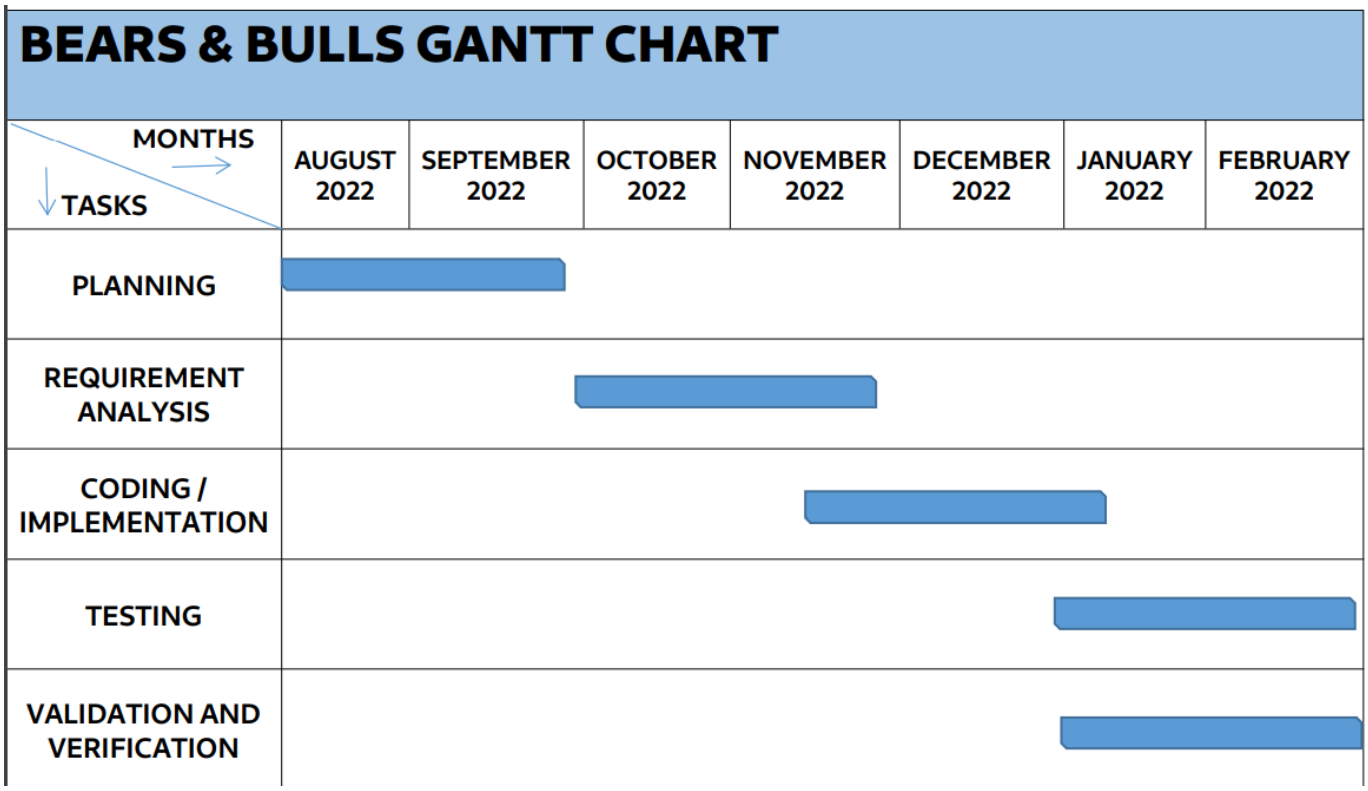
What is planning?

Planning is generally concerned with explaining and defining and the actual sequence of intermediate results. It is an event where we identify how much of the backlog we can commit to delivering during an upcoming project. A good judgment of the intricacies of the project and the development process helps us to identify the critical role of the project effectively. The large functions are broken down into a valid set of small activities which would help us to organize our plans in a more effective way.

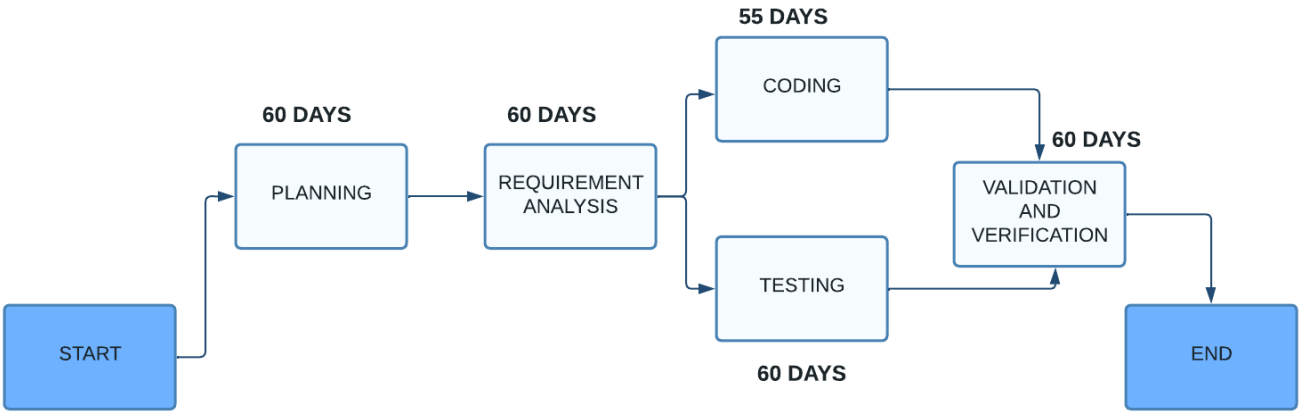
What is scheduling?

Scheduling is a collection of techniques used to develop and present schedules that show when work will be performed. The work breakdown structure formalism supports us to break down the function systematically after all tasks have broken down , the purpose and constructs the work breakdown structure; we can find the dependency among the activities.

a. Gantt Chart



b. Pert Chart



5. Preliminary Product Description

Preliminary product description helps in identifying the requirements and the objectives of the new proposed product/project/system. It helps in defining the functions and associated activities or operations of the proposed product/project/system.

The prime focus of this project is to create a user-friendly, all age inclusive and brain tickling game application called Bears & Bulls. The objective of the game is to iteratively improve ability of the user to guess the words of variable difficulty in minimum number of turns

The requirements for this application is through understanding of Python concepts in which the game will be developed, good hold on development of front end and database connectivity, in our case we will be using SQLite server.

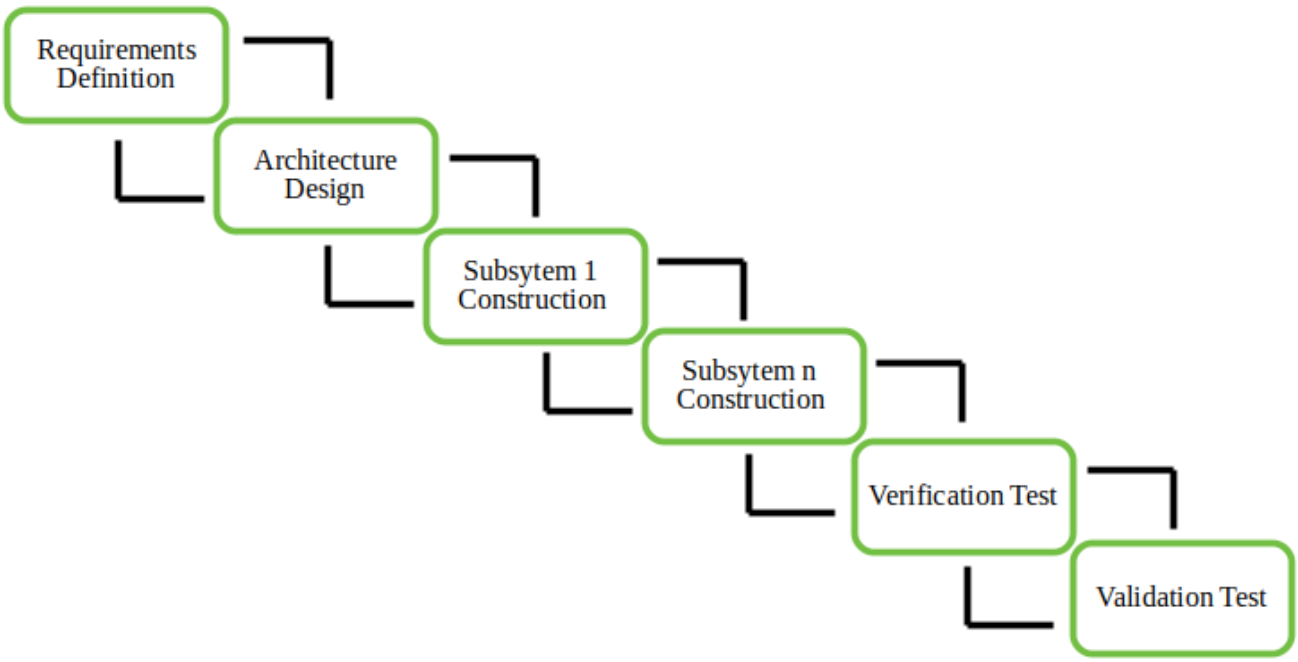
6. Conceptual Model

a. Process Model

Process models are processes of the same nature that are classified together into a model. Thus, a process model is a description of a process at the type level. One possible use of a process model is to prescribe how things must/should/could be done in contrast to the process itself which is really what happens

Proposed Process Model

- **Incremental Development Model**
- The core features are developed first and the whole software is developed by adding new features in successive versions.The requirements of the software are divided into several modules that can be incrementally developed and delivered.
- Fig 3.6.1 Incremental Development Model Diagram.



- Since this is a small scale project, the incremental model allows us to accommodate the entire process in an incremental fashion and it is a flexible model which helps us to satisfy our latest changing requirements.
- We first segregate our project according to our requirements and priority and then start dividing our project into smaller segments. With this model being flexible we can alter our project stages according to our needs,
- In the Incremental development model, there is no fixed time to complete the next iteration. It is easy to break down tasks because of the divide and conquer approach used. It is good to use when projects use new Technology and it is more flexible and less costly to change scope and requirements.
- There are also some disadvantages for this model, as it requires a good planning design and needs a clear, complete definition of the whole system before it can be broken down and built incrementally.

b. <u>The goals of a process model are to be:</u>

a. Descriptive

   Track what actually happens during a process.

b. Prescriptive

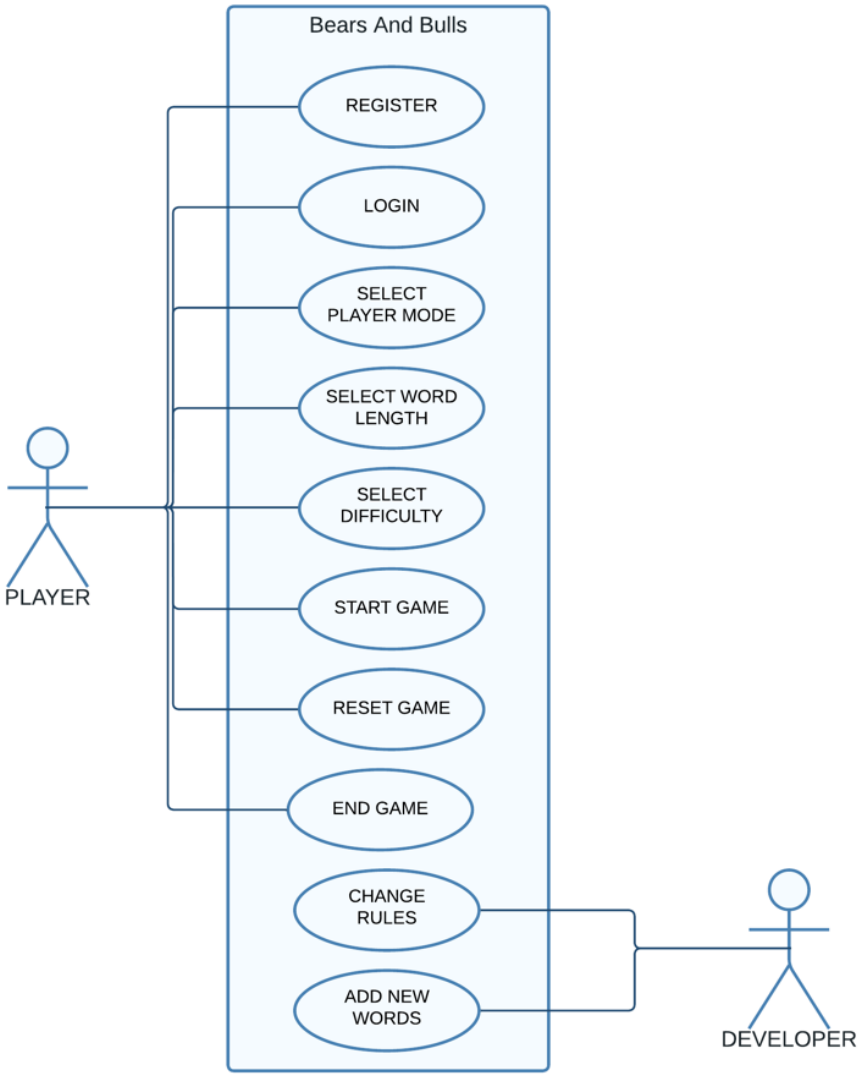
   Define the desired processes and how they should/could/might be performed.

c. Explanatory

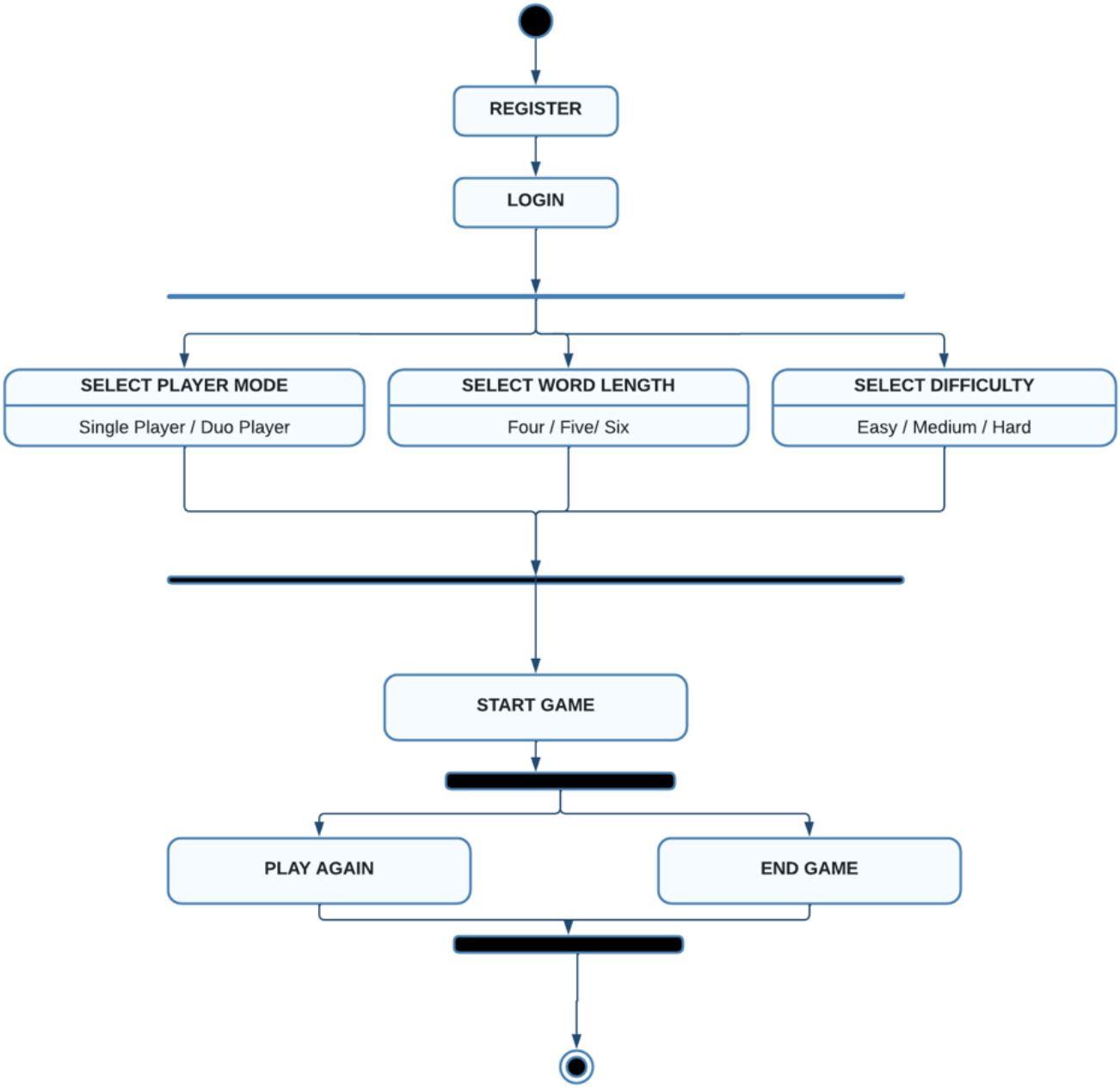
   Provide explanations about the rationale of processes.

c. Diagrams to be included in the design phase are as follows:

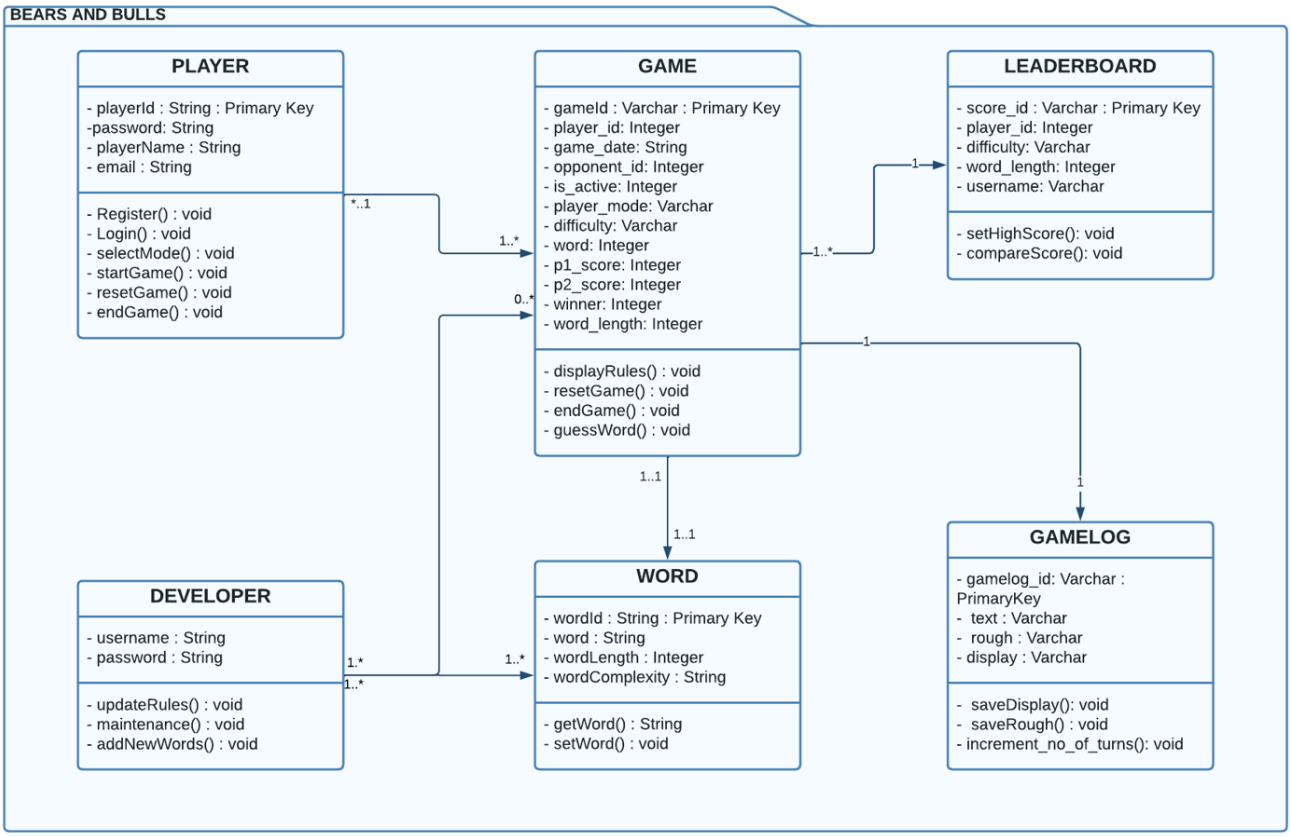
1. Use case diagram



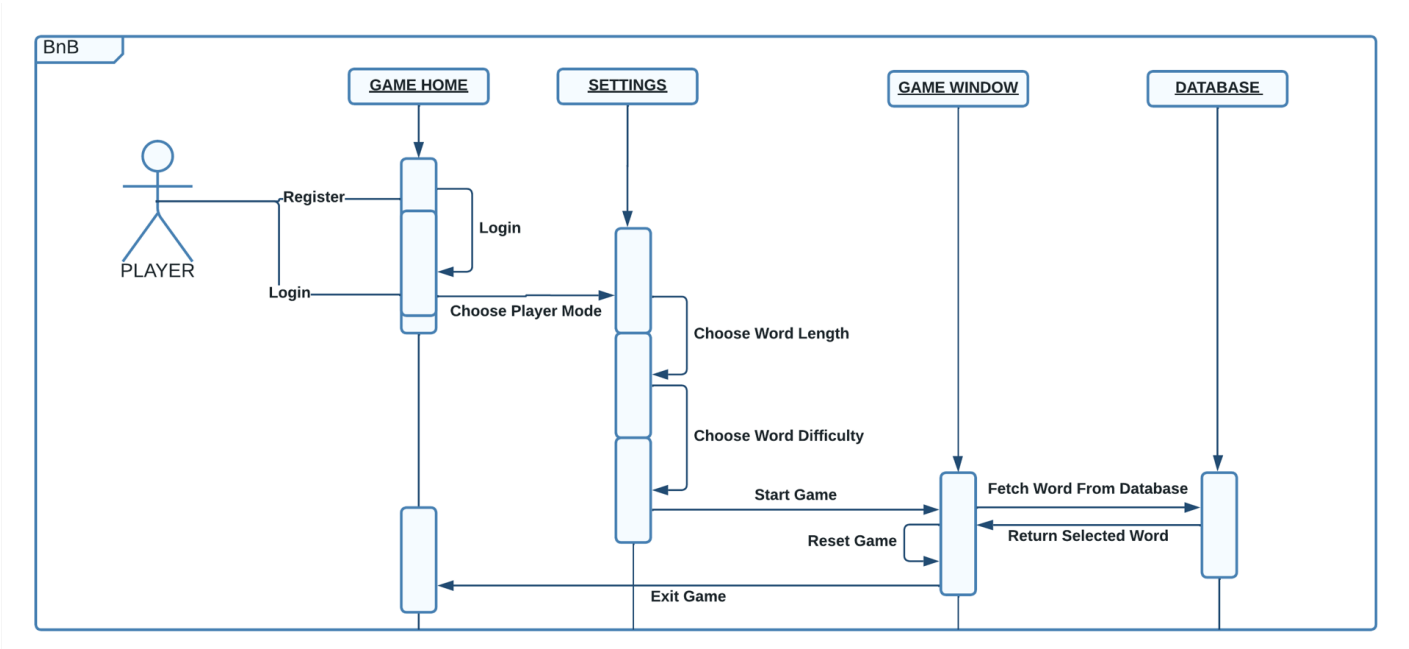
2. Activity diagram



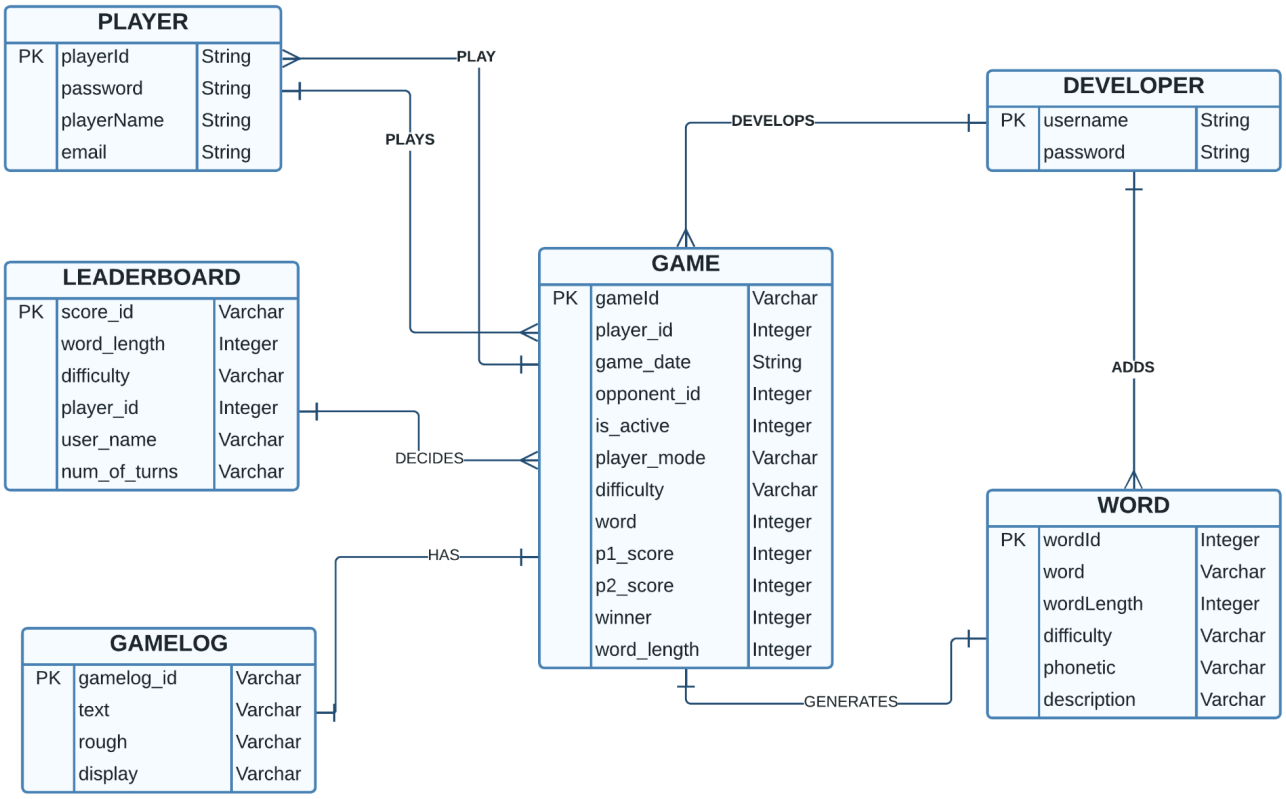
3. Class diagram



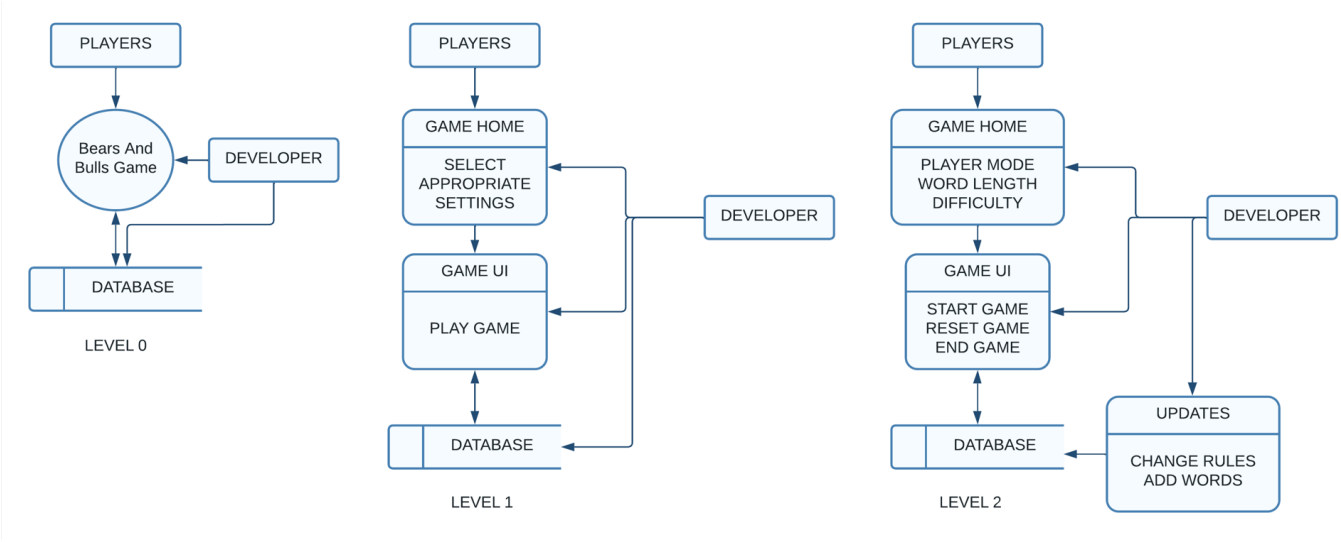
4. Sequence diagram



5. E-R model



6. Data Flow Diagram



CHAPTER 4 : SYSTEM DESIGN

Basic Modules

1.1 Register

1.2 Login

1.3 Select Player Mode

1.4 Select Word Length

1.5 Select Difficulty

1.6 Start Game

1.7 Quit Game

1.8 End Game

1.9 LeaderBoard

a. Description of Desired Modules

The following is the description of the above discussed modules:

1. Register

   It allows new players to provide their details and register as a new player.

2. Login

   After registration, the player can log into their account to play the game.

3. Start Game

   Once all the settings are validated we can start the game.

4. Quit Game

   If any player wishes to exit abruptly we can quit the game using this module.

5. End Game

   Once we have successfully guessed the word we have a choice to log out or play again.

6. LeaderBoard

   The player who guesses the word with the minimum number tries gets to be on the leaderboard.

b. Description of Desired Features

- Select Player Mode: We can play a single player game with the system or a two player game against each other.
- Select Word Length: We can set the word length as three, four and five lettered word that is to be guessed according to our wish
- Select Difficulty: We can choose between the difficulty level of a word as easy, medium and hard.

Data Design

In the design phase, the requirements will be broken down further to be able to forecast the project's timeline and estimate the level of effort and amount of resources needed. Design is a very important phase and is a multi-step process which represents structure, program, interface characteristics and procedural details. The proposed system is designed using the design models such as functional decomposition diagrams, data flow diagrams, entity relationship diagrams or any unified modeling language diagrams. The design phase includes all the diagrams which provide an outline of how the application would look.

I. Schema Design

No.	Trigger	Source	Activity	Response	Destination
1	New Register	New User	Create Account	Registration	Admin
2	Login	Admin	Login	Login successful	Start Game
3	New Game	Start Game	Generate word	Database generates word	Database
4	End Game	Quit/ Finish Game	End Game	Log out	Log out
5	Leaderboard	Endgame	Displays leaderboard	Shows leaderboard	Leaderboard page

II. Data Integrity and Constraints

1. Player

Column Name	Data Type	Constraints
playerId	String	Primary Key
password	String	Not Null
player Name	String	Not Null
email	String	Not Null

2. Game

Column Name	Data Type	Constraints
gameId	Varchar	Primary Key
player_id	Integer	Not Null
game_date	String	Not Null
opponent_id	Integer	Not Null
is_active	Integer	Not Null
player_mode	Varchar	Not Null
difficulty	Varchar	Not Null
word	Integer	Not Null
p1_score	Integer	Not Null
p2_score	Integer	Not Null
winner	Integer	Not Null
word_length	Integer	Not Null

3. Word

Column Name	Data Type	Constraints
wordId	String	Primary key
word	String	Not Null
wordLength	Integer	Not Null
complexity	String	Not Null
description	String	Not Null
phonetic	String	Not Null

4. GameLog

Column Name	Data Type	Constraints
gamelog_id	Varchar	Primary key
text	Varchar	Not Null
rough	Varchar	Not Null
display	Varchar	Not Null

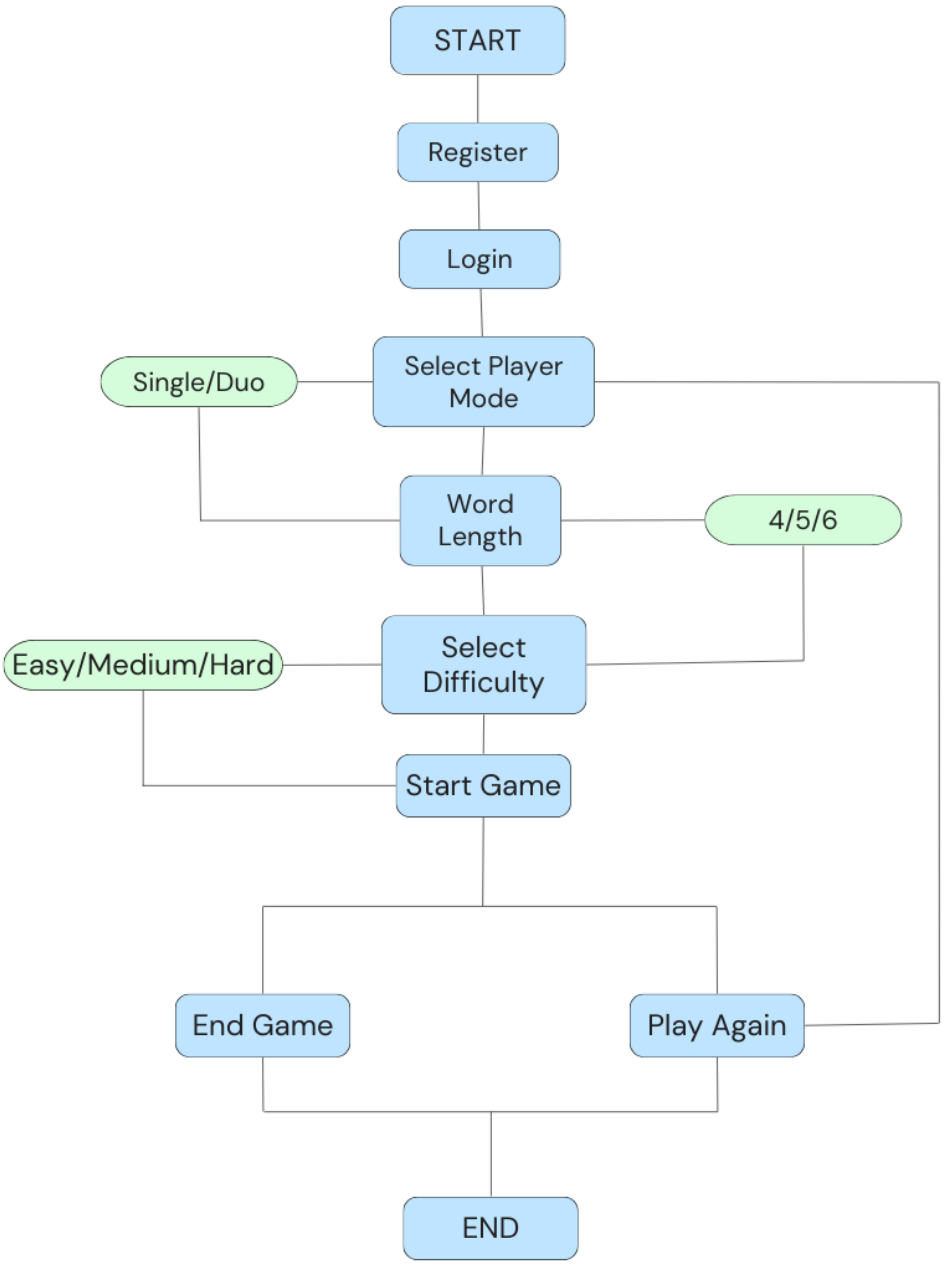
5. LeaderBoard

Column Name	Data Type	Constraints
score_id	Varchar	Primary key
player_id	Integer	Not Null
difficulty	Varchar	Not Null
word_length	Integer	Not Null
username	Varchar	

Procedural Design

I. Logic Diagram

Fig 4.3 Logic Diagram



II. Data Structures

- Array

  An array is a collection of items stored at contiguous memory locations.

  It helps us for letter manipulation in our game and hence it is used in our project.

III. Algorithm design

- Step 1:

  Fetching a superword from the database.

- Step 2:

  Take input from the user as a word without having repeated letters.

- Step 3:

  Validation of the word basis of a given set of rules.

- Step 4:

  If the word satisfies the given condition, compare the word with superword.

- Step 5:

  If the word has the same letters as the superword then increment the count of bears.

- Step 6:

  If the word has letters in the same position as the superword then increment the count of bulls.

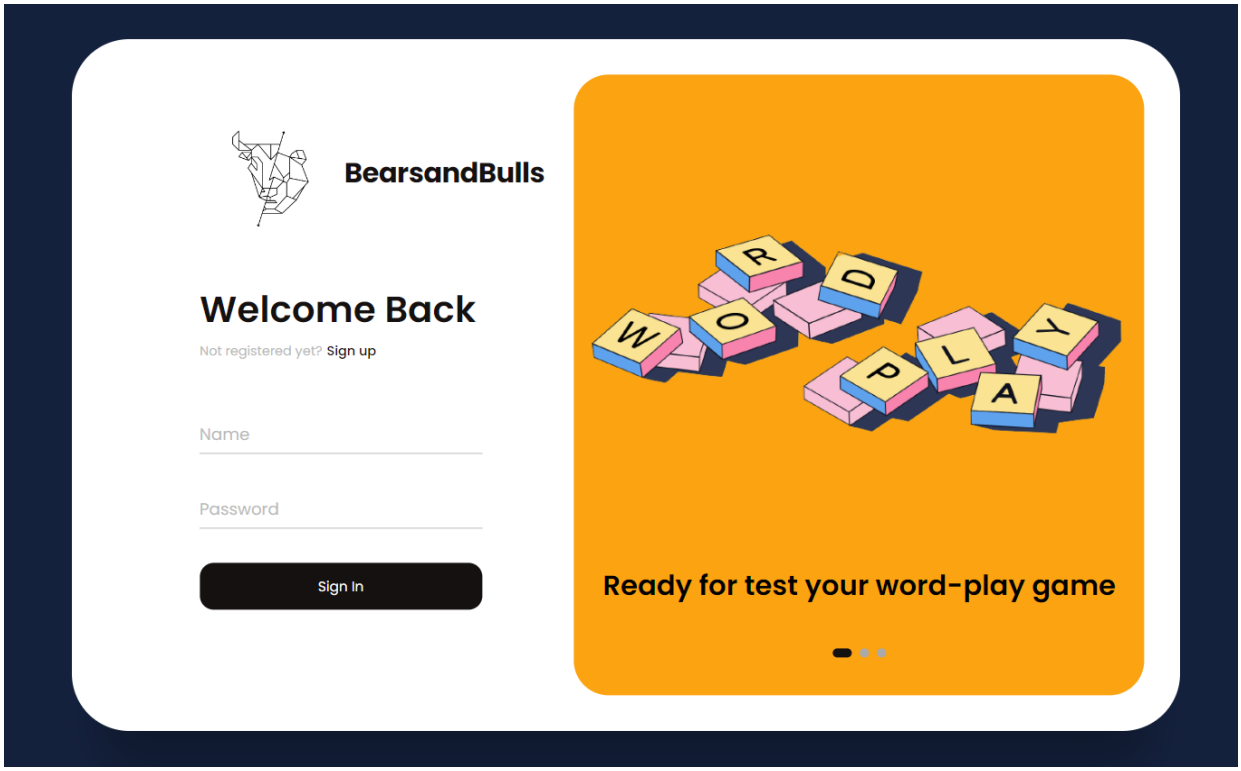
- Step 7:

  If all the words match the superword you have won the game

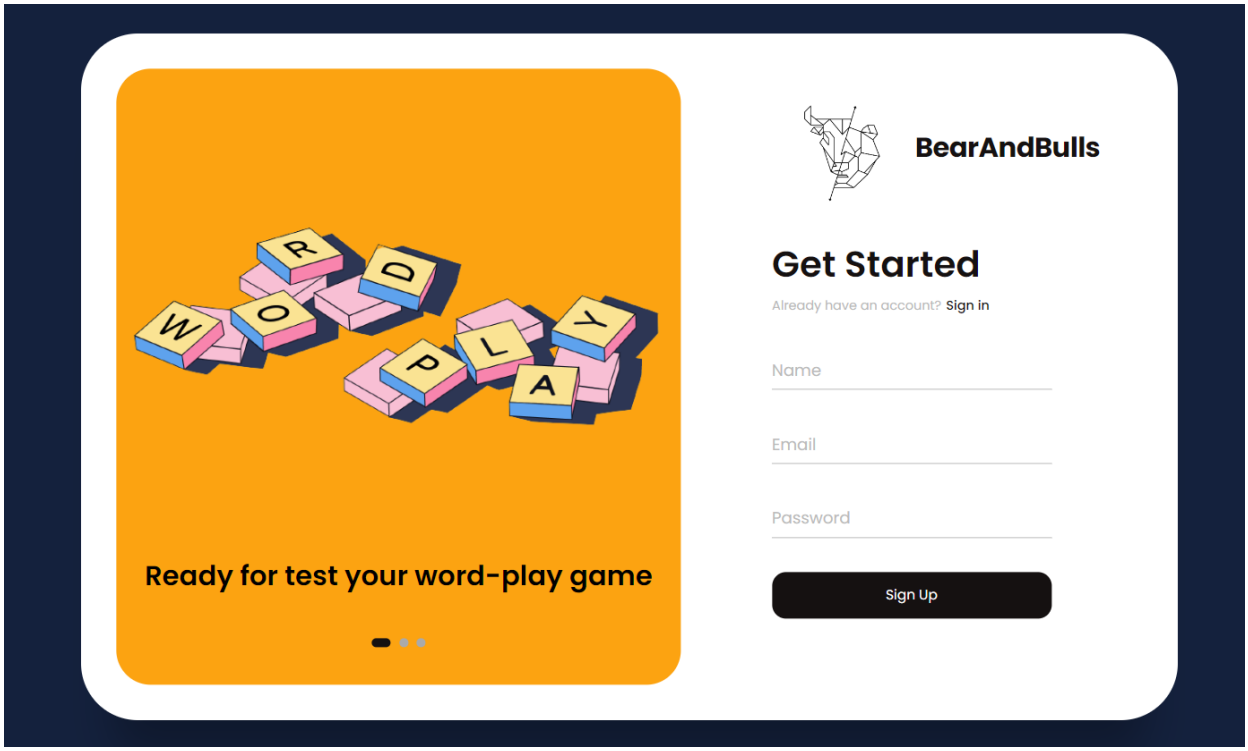
  or you can also quit the game

User Interface Design

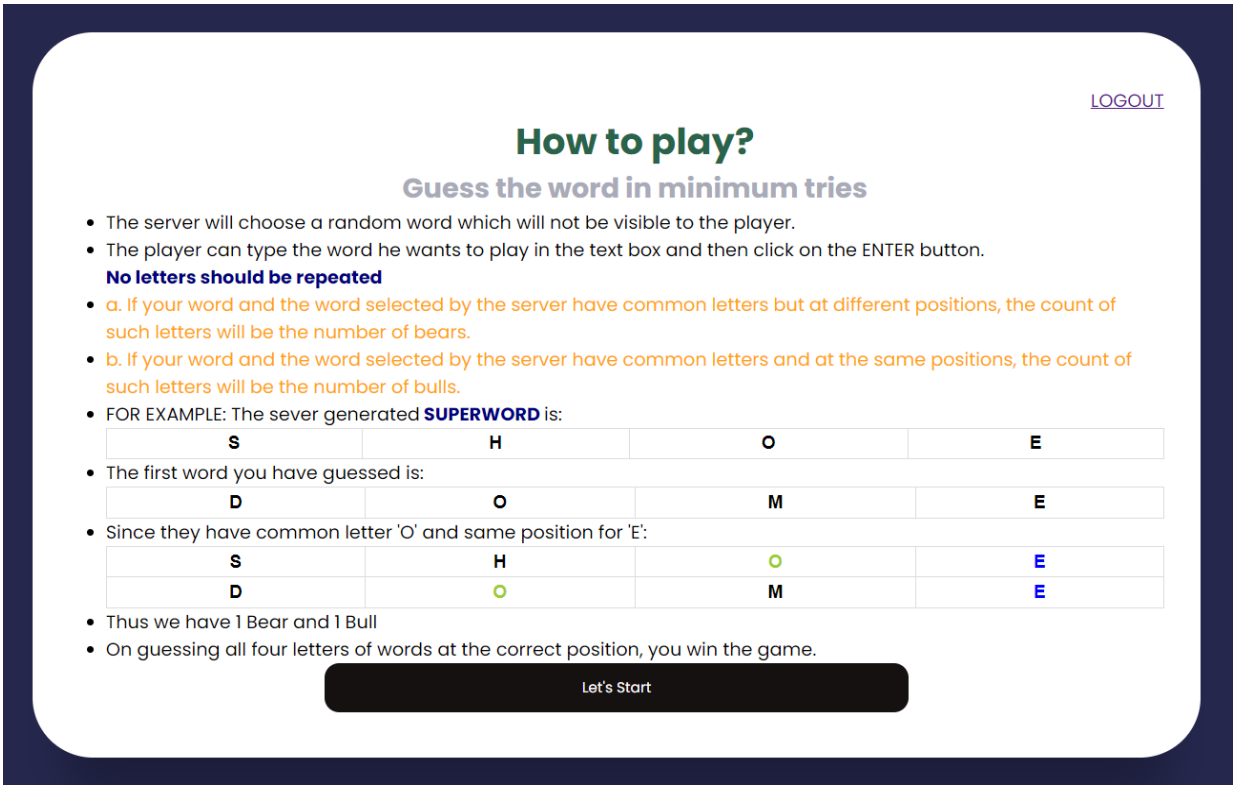
I. Login



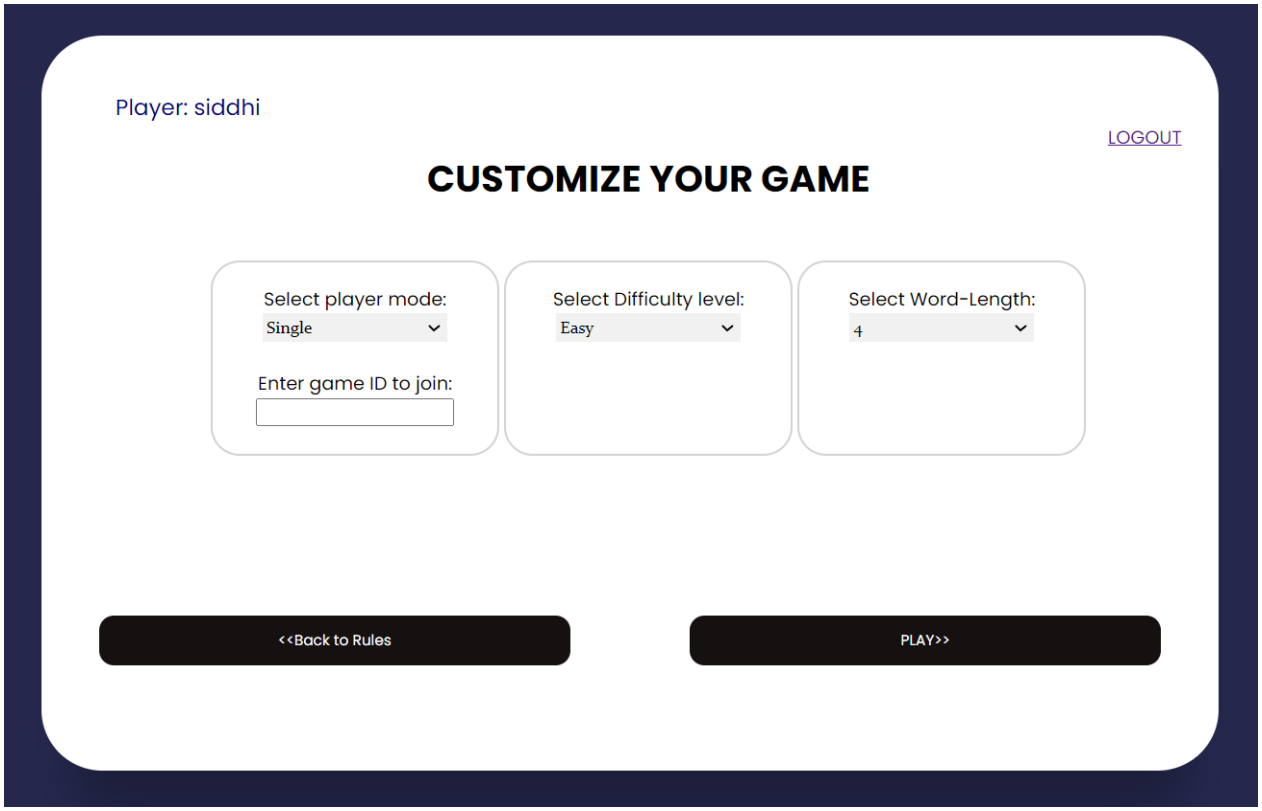
II. Register



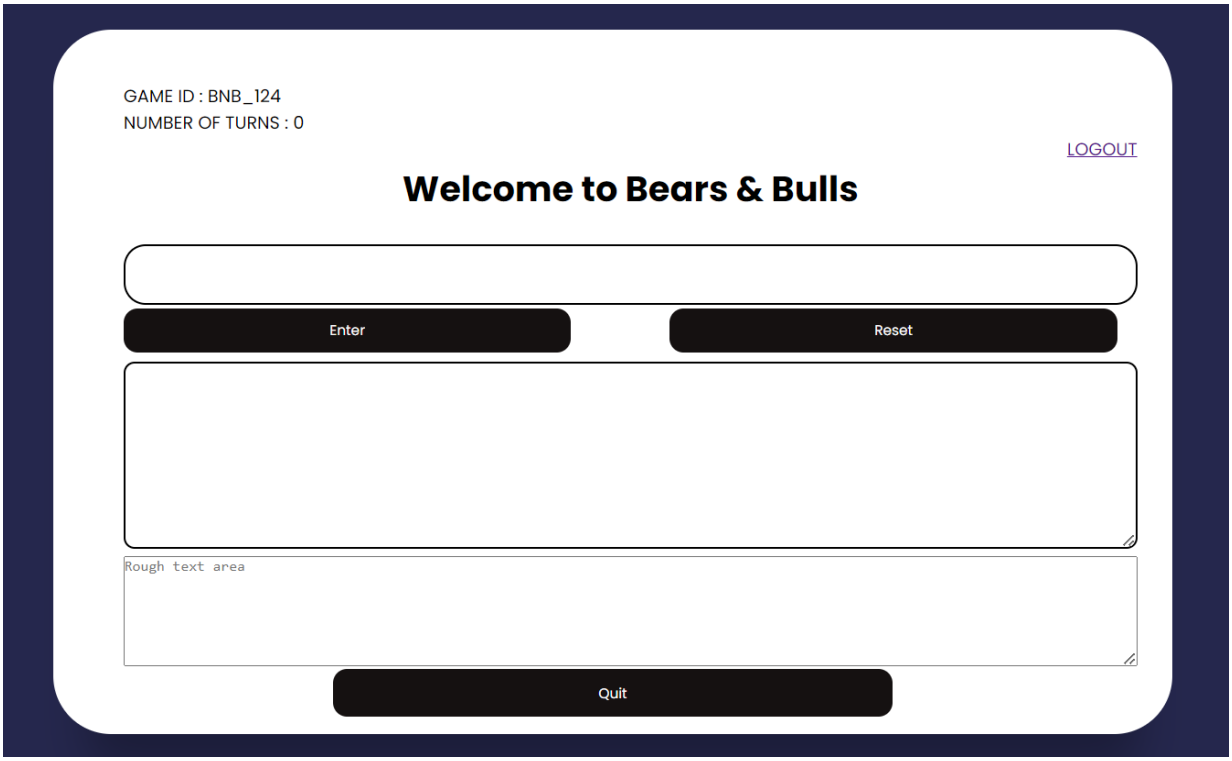
III. Rules



IV. Customize your Game



V. Game Interface



Security Issues

I. There must not be duplicate registrations so that the user cannot manipulate the other user's account and thus cause invasion of one's data integrity.

II. Since this is a wordplay game people often tend to cheat by using unethical means.

III. With proper planning and eternal vigilance, we can avoid such casualties and provide a secure environment.

CHAPTER 5 : IMPLEMENTATION AND TESTING

1. Implementation Approaches

The Incremental Development Model is a software development approach that breaks down a project into smaller, manageable chunks called increments. Each increment is developed and tested independently before being integrated into the final product. This approach is particularly useful for large and complex projects, such as a word play game, as it allows for a gradual, step-by-step development process and the ability to make changes and adjustments as needed.

a. Define the implementation plan

Plan for how Incremental Development Model could be applied to a word play game is:
Initial increment: Develop basic game features such as the user interface and word database.
Second increment: Add game mechanics such as scoring and level progression.
Third increment: Implement multiplayer functionality
Fourth increment: Add additional game modes and challenges.
Fifth increment: Finalize the game, conduct testing and debugging, and prepare for release.
Each increment is developed, tested, and integrated with the previous increment, allowing for a gradual build-up of the game's functionality. This approach allows for feedback to be incorporated and changes to be made as needed, resulting in a polished, fully functional final product.

b. State the standards and protocols used in implementation

Django uses basic server side protocol HTTP to communicate between the client and server. Django's ORM (Object-Relational Mapping) allows you to define your data models as Python classes, and the database is automatically created based on these classes following the DRY (Don't Repeat Yourself) principle, and keep your models simple and modular.

It also has in-built security standards such as user authentication and permissions, to protect sensitive information and prevent attacks such as SQL injection and cross-site scripting (XSS). For commenting I have used Contextual commenting to describe any purpose, intend or feature of the code and also logical comments.

2. Coding Details and Code Efficiency

a. Code of the main logic (must be with comments)

views.py

```
"""=========================#
Author : Siddhi Santosh Kotre
Project : Bears And Bulls
TYBSc IT Roll Number : 730
#========================="""
# Importing packages
import random
from django.contrib import messages
from django.contrib.auth import authenticate, login, logout
from django.contrib.auth.decorators import login_required
from django.contrib.auth.models import User
from django.core import serializers
from django.shortcuts import render
from .models import game, word, gamelog, leaderboard

def index(request):
  return render(request, 'index.html')

# Module for registering a new user
def register(request):
  if request.method == 'POST':
    username = request.POST['username']
    email = request.POST['email']
    password = request.POST['password']
    user = User.objects.create_user(username=username, email=email, password=password, )
    user.save();
    messages.success(request, "Your account has been successfully created")
  return render(request, 'register.html')

# Module for user login authentication
```

```
def login_view(request):
  # Retrieving user inputs for logging in
  if request.method == 'POST':
    username1 = request.POST['username1']
    password1 = request.POST['password1']
    user1 = authenticate(username=username1, password=password1)
    # Checking if user is valid and logging him into the game
    if user1 is not None:
      login(request, user1)
      return render(request, "rules.html")
    # if user is invalid displaying appropriate error message
    else:
      messages.error(request, "Invalid username or password.")
      return render(request, "register.html")
  return render(request, 'register.html')

# Rendering rules webpage
@login_required
def rules(request):
  player_id = request.user
  print(f"player is {player_id},{player_id.id}")
  return render(request, 'rules.html')

# Module for pre game setting selections
@login_required
def pregame(request):
  if request.method == 'POST':
    # Resetting the sessions for new game
    request.session['display'] = ""
    request.session['rough'] = ""
    request.session['turn'] = 0
    player = request.user
    playerch = request.POST['player']
    opp_game_id = str(request.POST['txtvalue']).strip()
```

```
    # If trying to join a duo game setting it's id as opponent player id
    if opp_game_id != "":
        game_obj2 = game.objects.filter(game_id=opp_game_id).first()
        if game_obj2 is None or game_obj2.is_active!=0:
            messages.error(request,"Sorry, No such game exists! Please try again")
            return render(request, 'pregame.html')
        game_obj2.opponent_id = request.user.id
        game_obj2.is_active = 1
        game_obj2.save()
        game_id = game_obj2.game_id
    # Else, saving player's choice of length and difficulty of word
    else:
        difficulty = request.POST['difficulty']
        length = request.POST['length']
        # Generating a new Game ID using Increment Game Number method
        game_id = increment_game_number()

        if player is not None:
            game_obj = game()
            game_obj.game_id = game_id
            game_obj.player_id = player.id
            # If it is single player choice set game as active
            if playerch == 'single':
                game_obj.is_active = 1
            game_obj.player_mode = playerch
            game_obj.difficulty = difficulty
            game_obj.word_length = length
            # Selecting a random super word based on player's choices
            swfilter = word.objects.filter(length=length).filter(difficulty=difficulty)
            game_obj.word = random.choice(list(swfilter.values()))['word_id']
            game_obj.save()
    # Setting game id as necessary sessions to be used during games
    request.session['game_id'] = game_id
    request.session['gamelog_id'] = game_id
```

```
        gamelogobj = gamelog()
        gamelogobj.gamelog_id = game_id
        gamelogobj.save()
        return render(request, 'game.html')
    return render(request, 'pregame.html')

# Core logic for calculating bears and bulls
def bears_and_bulls(superword, curword, game_id, request, description, phonetic):
    # Initializing variables and converting them into upper case for handling case sensitivity
    message = ""
    input_word = curword.upper()
    super_word = superword.upper()
    print(input_word + " " + super_word)
    invalid = False
    # Input word validation block
    # Rule : The input word length should be same as the length selected in pre game settings.
    if len(input_word) != len(super_word):
        message = f"Please enter a {len(super_word)} lettered word..."
        invalid = True
    else:
        # Rule : Letters in the word should not be repeated.
        for i in range(len(super_word)):
            for j in range(i + 1, len(super_word)):
                if input_word[i] == input_word[j]:
                    invalid = True
        # If any of the above rule is not satisfied set appropriate error message
        if invalid:
            message = "Cannot accept word with duplicate letters, please enter new word..."
    # If the entered word is valid, calculating bears and bulls
    if not invalid:
        bears = 0
        bulls = 0
        for i in range(len(super_word)):
            for j in range(len(super_word)):
```

```
            if (input_word[i] == super_word[j]) and (i == j):
                bulls += 1
            elif (input_word[i] == super_word[j]) and (i != j):
                bears += 1
        # If number of bulls is equal to length of superword you have won the game
        if bulls == len(super_word):
            message = f"Congratulations you have guessed the word correct ! The word was {super_word}.
Description: {description} Phonetic:{phonetic}"
            gameobj = game.objects.filter(game_id=game_id).first()
            gameobj.is_active = 2
            # Setting respective player scores
            if request.user.id == gameobj.player_id:
                gameobj.p1_score = int(request.session['turn']) + 1
            elif request.user.id == gameobj.opponent_id:
                gameobj.p2_score = int(request.session['turn']) + 1
            gameobj.winner = request.user.id
            gameobj.save()
        # If the game is not over displaying number of bears and bulls
        else:
            message = f"YOUR WORD - {input_word} - {str(bears)} BEARS AND {str(bulls)}  BULLS."
    return message


@login_required
def game_view(request):
    # Retrieving game session details
    gamelog_id = request.session['gamelog_id']
    print(f"game log id {gamelog_id}")
    gamelogobj = gamelog.objects.all().filter(gamelog_id=gamelog_id).first()
    gameid = request.session['game_id']
    gameobj = game.objects.filter(game_id=gameid).first()
    # If the player has selected duo mode, game does not start if opponent has not joined
    if gameobj.is_active == 0:
        messages.info(request, "Wait for your opponent to join...")
    gameid = request.session['game_id']
```

```
    gameobj = game.objects.filter(game_id=gameid).first()
    wrd = word.objects.filter(word_id=gameobj.word).first()
    # Terminating the game if the opponent has already won
    if gameobj.is_active == 2:
        messages.info(request,
                      f"Your opponent won the game, The word was {wrd.word} which means {wrd.description}
and it is pronounced as {wrd.phonetic}. Try again next time...")
        # Saving the current scores of the player
        if request.user.id == gameobj.player_id:
            gameobj.p1_score = int(request.session['turn']) + 1
        elif request.user.id == gameobj.opponent_id:
            gameobj.p2_score = int(request.session['turn']) + 1
        gameobj.save()
        return render(request, 'endgame.html')
    # Terminating if your opponent quits the game
    if gameobj.is_active == 3:
        messages.info(request,
                      f"Unfortunately your opponent quit the game. The word was {wrd.word} which means
{wrd.description} and it is pronounced as {wrd.phonetic}. Please join a new game.")
        return render(request, 'endgame.html')
    # Main active game logic
    if gameobj.is_active == 1:
        if request.method == 'POST':
            current_word = request.POST['text']
            gamelogobj.inputtext = current_word
            wrd = word.objects.filter(word_id=gameobj.word).first()
            # Passing the input value from players to Bears and Bulls Method
            message = bears_and_bulls(wrd.word, current_word, gameid,
                                      request, wrd.description, wrd.phonetic)
            if "Congratulations" in message:
                messages.success(request,
                                 f"You have guessed the word correct ! The word was {wrd.word.upper()}."
                                 f" Description: {wrd.description} "
                                 f"Phonetic: {wrd.phonetic}")
```

```
            # Setting leaderboard values
            leader_row = leaderboard.objects.filter(word_length=gameobj.word_length).filter(
                difficulty=gameobj.difficulty).first()
            # If this is the first entry in that leaderboard category
            if not leader_row.num_of_turns.isnumeric():
                leader_row.player_id = request.user.id
                leader_row.user_name = request.user.username
                leader_row.num_of_turns = int(request.session['turn']) + 1
                leader_row.save()
            # If it is not the first entry comparing the existing high scores
            elif int(request.session['turn']) < int(leader_row.num_of_turns):
                leader_row.player_id = request.user.id
                leader_row.user_name = request.user.username
                leader_row.num_of_turns = int(request.session['turn']) + 1
                leader_row.save()
            return render(request, 'endgame.html')
        # Display area updated with the latest message
        display = request.user.username + ": " + message + "\n" + gamelogobj.display
        request.session['turn'] = display.count(request.user.username)
        gamelogobj.rough = request.POST['rough']
        gamelogobj.display = display
        request.session['display'] = display
        request.session['rough'] = request.POST['rough']
        gamelogobj.save()
    return render(request, 'game.html')

# Rendering end game page
def endgame(request):
    return render(request, 'endgame.html')

# Module for quit game event
def quit_event(request):
    gameid = request.session['game_id']
    gameobj = game.objects.filter(game_id=gameid).first()
```

```
    # Setting game active id as 3, to notify opponent that other player has quit the game.
    gameobj.is_active = 3
    gameobj.save()
    wrd = word.objects.filter(word_id=gameobj.word).first()
    # Displaying the superword to the player
    messages.info(request,
                  f"Better Luck Next Time! The word was {wrd.word} which means {wrd.description} and it is
pronounced as {wrd.phonetic}.")
    return render(request, 'endgame.html')

# Module for generating new game ID
def increment_game_number():
    gids = game.objects.all().values_list('game_id', flat=True)
    last_game = []
    if not gids:
        return 'BNB_' + '0'
    for i in gids:
        last_game.append(int(i.split('_')[1]))
    last_game.sort()
    booking_int = last_game[len(last_game) - 1] + 1
    return 'BNB_' + str(booking_int)
# Displaying leaderboard on the page
def leaderboard_view(request):
    data = serializers.serialize("python", leaderboard.objects.all())
    context = {
        'data': data,
    }
    return render(request, 'leaderboard.html', context)
# Module for a player to log out of current session
def logout_view(request):
    logout(request)
    return render(request, 'register.html')
```

Models.py

```
from django.db import models

class word(models.Model):
  word_id = models.IntegerField(primary_key=True)
  word = models.CharField(max_length=50)
  length = models.IntegerField(default=4)
  difficulty = models.CharField(max_length=20, default="easy")
  description = models.CharField(max_length=200, default="Description not available")
  phonetic = models.CharField(max_length=200, default="phonetic not available")

  def __str__(self):
     return self.name

'''==========================================================================
==========
game.is_active options:
game.is_active = 0 -> The game is inactive and waiting for your opponent to join
game.is_active = 1 -> The game is active and in play
game.is_active = 2 -> The game is inactive since your opponent has won the game
game.is_active = 3 -> The game is inactive since your opponent has quit the game
==========================================================================
=========='''

class game(models.Model):
  player_id = models.IntegerField()
  game_id = models.CharField(primary_key=True, max_length=50)
  opponent_id = models.IntegerField(default=0)
  is_active = models.IntegerField(default=0)
  game_date = models.DateField(auto_now=True)
  player_mode = models.CharField(max_length=20, default='single')
  word_length = models.IntegerField(default=4)
  difficulty = models.CharField(max_length=20, default='easy')
```

```
    word = models.IntegerField()
    p1_score = models.IntegerField(default=0)
    p2_score = models.IntegerField(default=0)
    winner = models.IntegerField(default=0)

    def __str__(self):
        return self.name

class gamelog(models.Model):
    gamelog_id = models.CharField(max_length=50, primary_key=True)
    text = models.CharField(max_length=50)
    display = models.CharField(max_length=5000, default="")
    rough = models.CharField(max_length=5000, default="")

class leaderboard(models.Model):
    score_id = models.CharField(max_length=50, primary_key=True)
    word_length = models.IntegerField(default=4)
    difficulty = models.CharField(max_length=20)
    player_id = models.IntegerField(default=0)
    user_name = models.CharField(max_length=100, default='<Imagine your name here, now go
play>')
    num_of_turns = models.CharField(max_length=100, default='--')

    def __str__(self):
        return self.name
```

url.py

```
from django.contrib.staticfiles.urls import staticfiles_urlpatterns
from django.urls import path

from . import views

urlpatterns = [
  path('', views.index, name="index"),
  path('register/', views.register, name="register"),
  path('login/', views.login_view, name="login"),
  path('logout/', views.logout_view, name="logout"),
  path('rules/', views.rules, name="rules"),
  path('pregame/', views.pregame, name="pregame"),
  path('game/', views.game_view, name="game"),
  path('endgame/', views.endgame, name="endgame"),
  path('quit/', views.quit_event, name="quit"),
  path('leaderboard/', views.leaderboard_view, name="leaderboard"),

]

urlpatterns += staticfiles_urlpatterns()
```

b. Code of the algorithm, if any

Step 1: Fetching superword from database

Step 2: Take input from user as word without having repeated letters

Step 3: Validation of the word of basis of given set of rules

Step 4: If the word satisfies the given condition, compare the word with superword.

Step 5: If the word has the same letters as the superword then increment the count of bears.

Step 6: If the word has letters in the same position as the superword then increment the count of bulls.

Step 7: If all the words match the superword you have won the game or you can also quit the game

c. Code Efficiency

There is still room for improvement in my code. The code is running smoothly on various platforms but the code can be further optimized by using more efficient algorithms and data structure. Avoiding cache can also avoid redundant work. Using tools like static analysis and linting to identify potential performance issues or coding mistakes.

Optimization of the code should be done in such a way that it will strike a balance between performance and maintainability and to achieve this we have to use benchmarking tools to attain optimality

3. Testing Approach

Rule-based system State machine used for testing the system where rules are used to determine the behavior of game entities based on their current state and the events that occur in the game.

The rules can be written in a scripting language, such as Python, or represented as data structures in code. The rules are processed by the game engine to determine the behavior of the entities. Rule-based systems and state machines can be used in a game to manage the behavior of game entities, but the choice between them will depend on the requirements of the project and the design of the game.

a. Functional Testing

This type of testing is used to ensure that the game functions as intended and that all the features work as expected. This includes testing the game's mechanics, user interface, and the overall gameplay experience.

1. User Acceptance Testing or Beta Testing

User Acceptance Testing and Beta Testing are important steps in the development of a wordplay game as they help to ensure that the game meets the needs and expectations of the target users, and that it is functional and of high quality before it is released to the public.

2. Unit Testing

In the context of a wordplay game, unit tests can be used to verify the behavior of individual game mechanics, such as word validation, scoring, and word completion, as well as smaller units of code, such as utility functions or data structures.

3. Integration Testing

Integration testing is a critical step in the software development process as it helps to ensure that different components or systems of a wordplay game work together seamlessly, improving the overall quality of the game.

b. Non-Functional Testing

. The testing process should be iterative, and the game should be tested, modified and re-tested multiple times until it meets the quality standards. It's important to test the game with a diverse group of users to ensure the game is suitable for a wide range of players.

1. Performance Testing

This type of testing is used to ensure that the game runs smoothly and performs well on different devices and platforms. It is also used to test the game's load and stress limits.

2. Scalability Testing

Scalability testing is the process of testing a software application's ability to handle increasing workloads, such as more users or larger data sets. Scalability testing can be used to verify that the game can handle increasing numbers of players, larger dictionaries, and more complex game mechanics without affecting performance or functionality.

3. Portability Testing

Portability testing is the process of testing a software application's ability to run on different platforms, such as different operating systems, devices, or browsers. For a wordplay game, portability testing can be used to verify that the game can run on different devices, such as smartphones, tablets, and laptops, and different operating systems, such as Windows, MacOS, and Linux, without any issues.

c. Black Box Testing

Blackbox testing for a word play game refers to testing the game without having access to its internal code or structure. The focus is on testing the game's external behavior and functionality. Black box testing involves testing the user interface, checking if the game functions as expected, and ensuring that the game mechanics work correctly.

d. White Box Testing

Whitebox testing for a wordplay game refers to testing the game by having access to its internal code and structure. The focus is on testing the game's internal logic and functionality. white box testing involves testing the algorithms used to generate valid words, checking if the data structures used to store the game state are efficient, or ensuring that the code follows best practices and coding standards.

4. Test Cases

TEST CASE			
System Name:	Bears & Bulls		
Module Code:	SK001 - Login And Registration		
Pass	Number of test cases passed	Pending	Number of test cases pending
Fail	Number of test cases failed	Number of test cases:	10

ID	Test Case Description	Test Case Procedure	Expected Output	Actual Output	Test date	Result	Note
Test Case 1	Register Name, Email and Password	1. Enter your name, email, use proper annotations and set a password 2. The length of password should be more than 4 characters	Takes input and creates a account for that user in database as a new entry	All inputs taken successfully and entry added in database	04th January 2023	Pass	
Test Case 2	Register using same Credentials	1. Enter same name and email address and password	Should give warning about entering same credentials	Gives error message of please use other credentials	04th January 2023	Pass	
Test Case 3	Sign Up Button	1. Clicking on Sign up button	The slider will move and the sign in section will be displayed	The page gets refreshed and the sign in section is displayed	04th January 2023	Pass	
Test Case 4	Sign in	1. Click on sign in to go to sign in section	The slider will move and the sign in section will be displayed	The slider moves and displays the sign in section	04th January 2023	Pass	
Test Case 5	Sign in using incorrect credentials	1. Enter the name and password used while registration 2. Click on sign in	Warning box which states that your username and password is incorrect	Warning display for having invalid username and credentials	05th January 2023	Pass	
Test Case 6	Sign in using the same credentials you registered	1. Enter Name and password 2. Click on Sign in to enter the next page	On entering the credentials you should be redirected to next page	The credentials are correct but does not move to next page	05th January 2023	Fail	
Test Case 7	Sign in using the same credentials you registered	1. Enter Name and password 2. Click on Sign in to enter the next page	On entering the credentials you should be redirected to next page	The credentials are correct and we have signed in successfully	05th January 2023	Pass	

Test Case 8	Sign in using incorrect credentials	1. Enter the name and password used while registration 2. Click on sign in	Warning box which states that your username and password is incorrect	Warning display for having invalid username and credentials	05th January 2023	Pass	
Test Case 9	Rules	1. After reading the rules click on next button	The page will be redirected to the next page	The page gets redirected	06th January 2023	Pass	
Test Case 10	Player Name	The player name that you have entered while login	The player name gets displayed on the page	The player name gets displayed on the page at the top left corner	06th January 2023	Pass	

TEST CASE			
System Name:	Bears & Bulls		
Module Code:	SK002 - Customization of Game		
Pass	Number of test cases passed	Pending	Number of test cases pending
Fail	Number of test cases failed	Number of test cases:	10

ID	Test Case Description	Test Case Procedure	Expected Output	Actual Output	Test date	Result	Note
Test Case 1	DropDown box	1. Checking whether all the dropdown boxes have proper options	Each option selected must be successfully fetched from the database	All options were successfully fetched from the database	08th January 2023	Pass	
Test Case 2	Select the mode of single, easy and four lettered	1. From dropdown select the mode:single difficulty:easy letters:4 2. Click on let's play button	A game should be generated with the selected preferences	Redirected to next page with the selected preferences single, easy and four letters	08th January 2023	Pass	
Test Case 3	Select the mode single, easy and five lettered	1. From dropdown select the mode:single difficulty:medium letters:5 2. Click on let's play button	A game should be generated with the selected preferences	Redirected to next page with the selected preferences single, medium and five letters	08th January 2023	Pass	
Test Case 4	Select the mode single, easy and 6 lettered	1. From dropdown select the mode:single difficulty:hard letters:6 2. Click on let's play button	A game should be generated with the selected preferences	Redirected to next page with the selected preferences single, medium and six letters	08th January 2023	Pass	
Test Case 5	Select the mode single, medium and 4 lettered	1. From dropdown select the mode:single difficulty:medium	A game should be generated with the selected	Redirected to next page with the selected	08th January 2023	Pass	

		letters:4 2. Click on let's play button	preferences	preferences single, medium and four letters			
Test Case 6	Select the mode single, medium and 5 lettered	1. From dropdown select the mode:single difficulty:medium letters:5 2. Click on let's play button	A game should be generated with the selected preferences	Redirected to next page with the selected preferences single, medium and five letters	08th January 2023	Pass	
Test Case 7	Select the mode single, medium and 6 lettered	1. From dropdown select the mode:single difficulty:medium letters:6 2. Click on let's play button	A game should be generated with the selected preferences	Redirected to next page with the selected preferences single, hard and four letters	08th January 2023	Pass	
Test Case 8	Select the mode single, hard and 4 lettered	1. From dropdown select the mode:single difficulty:hard letters:4 2. Click on let's play button	A game should be generated with the selected preferences	Redirected to next page with the selected preferences single, hard and four letters	08th January 2023	Pass	
Test Case 9	Select the mode single, hard and 5 lettered	1. From dropdown select the mode:single difficulty:hard letters:5 2. Click on let's play button	A game should be generated with the selected preferences	Redirected to next page with the selected preferences single, hard and five letters	08th January 2023	Pass	
Test Case 10	Select the mode single, hard and 6 lettered	1. From dropdown select the mode:single difficulty:hard letters:6 2. Click on let's play button	A game should be generated with the selected preferences	Redirected to next page with the selected preferences single, hard and six letters	08th January 2023	Pass	

TEST CASE			
System Name:	Bears & Bulls		
Module Code:	SK003 - Working Of Game		
Pass	Number of test cases passed	Pending	Number of test cases pending
Fail	Number of test cases failed	Number of test cases:	14

ID	Test Case Description	Test Case Procedure	Expected Output	Actual Output	Test Date	Result	Note
Test Case 1	You have selected a 4 letter word but you are entering a different lettered	1. Enter the word in the textbox 2. Click on Enter	Should give a error message in textbox indicating " Please enter a 4	Gives the expected error message	09th January 2023	Pass	

	word		lettered word"				
Test Case 2	You have selected a 5 letter word but you are entering a different lettered word	1. Enter the word in the textbox 2. Click on Enter	Should give a error message in textbox indicating " Please enter a 5 lettered word"	Gives the expected error message	09th January 2023	Pass	
Test Case 3	You have selected a 6 letter word but you are entering a different lettered word	1. Enter the word in the textbox 2. Click on Enter	Should give a error message in textbox indicating " Please enter a 6 lettered word"	Gives the expected error message	09th January 2023	Pass	
Test Case 4	You have entered a 4 lettered word	1. Enter the word in the textbox 2. Click on Enter	Takes the input word and displays the number of bear and bulls occuring in them	Successfully displays the number of bears and bulls in the word	09th January 2023	Pass	
Test Case 5	You have entered a 5 lettered word	1. Enter the word in the textbox 2. Click on Enter	Takes the input word and displays the number of bear and bulls occuring in them	Successfully displays the number of bears and bulls in the word	09th January 2023	Pass	
Test Case 6	You have entered a 6 lettered word	1. Enter the word in the textbox 2. Click on Enter	Takes the input word and displays the number of bear and bulls occuring in them	Successfully displays the number of bears and bulls in the word	09th January 2023	Pass	
Test Case 7	Checking duplicate values for 4 lettered word	1. Enter the word in the textbox 2. Click on Enter	Displays an message in the textbox indicating please enter 4 lettered word	The error gets display as please enter a 4 lettered word	10th January 2023	Pass	
Test Case 8	Checking duplicate values for 5 lettered word	1. Enter the word in the textbox 2. Click on Enter	Displays an message in the textbox indicating please enter 5 lettered word	The error gets display as please enter a 5 lettered word	10th January 2023	Pass	
Test Case 9	Checking duplicate values for 6 lettered word	1. Enter the word in the textbox 2. Click on Enter	Displays an message in the textbox indicating please enter 6 lettered word	The error gets display as please enter a 6 lettered word	10th January 2023	Pass	
Test Case 10	Game id generation	1. On creating a new game the game id should be updated	The gameID should be assigned	Displays the gameID	10th January 2023	Pass	

Test Case 11	Number of turns	1. Enter the word in the textbox 2. Click on Enter	On pressing Enter the number of turns should be incremented by one each time	The number of turns gets updated successfully	10th January 2023	Pass	
Test Case 12	Reset	1. Enter the word in the textbox 2. Click on Reset	On pressing the reset button the value entered in the textbox should be reset	The string entered gets reset	11th January 2023	Pass	
Test Case 13	Quit	1. Click on Quit	The page should be redirected to nextendgame page	The page gets successfully redirected on Endgame	11th January 2023	Pass	
Test Case 14	Quit	1. Click on Quit	The page should be redirected to nextendgame page	Successfully displays the number of bears and bulls in the word	11th January 2023	Pass	

TEST CASE			
System Name:	Bears & Bulls		
Module Code:	SK04 - Working Of Game(Duo)		
Pass	Number of test cases passed	Pending	Number of test cases pending
Fail	Number of test cases failed	Number of test cases:	14

ID	Test Case Description	Test Case Procedure	Expected Output	Actual Output	Test Date	Result	Note
Test Case 1	Checking for generation of game ID	1. Start a new game	The gameID should be generated	The gameID is generated	13th January 2023	Pass	
Test Case 2	Entering the gameID that your opponent provided to start the game	1. Enter the gameID in Textbox in the pregame page 2.Click on lets start	A new game should be generated where both player can guess the same word	A new game gets generated with same word to guess	13th January 2023	Pass	
Test Case 3	Entries from both players will be displayed in the textbox	1. Enter the word in the textbox 2. Click on Enter	The inputs from the player gets displayed in the provided textbox	The guessed word are successfully displayed form both sides	13th January 2023	Pass	
Test Case 4	Number of turns	1. Enter the word in the textbox 2. Click on Enter	The number of turns gets updated of the individual player	Increment in number of turns of each player	13th January 2023	Pass	

Test Case 5	You have entered a 4 lettered word	1. Enter the word in the textbox 2. Click on Enter	Takes the input word and displays the number of bear and bulls occuring in them	Successfully displays the number of bears and bulls in the word	13th January 2023	Pass	
Test Case 6	You have entered a 5 lettered word	1. Enter the word in the textbox 2. Click on Enter	Takes the input word and displays the number of bear and bulls occuring in them	Successfully displays the number of bears and bulls in the word	13th January 2023	Pass	
Test Case 7	You have entered a 6 lettered word	1. Enter the word in the textbox 2. Click on Enter	Takes the input word and displays the number of bear and bulls occuring in them	Successfully displays the number of bears and bulls in the word	13th January 2023	Pass	
Test Case 8	Checking duplicate values for 4 lettered word	1. Enter the word in the textbox 2. Click on Enter	Displays an message in the textbox indicating please enter 4 lettered word	The error gets display as please enter a 4 lettered word	14th January 2023	Pass	
Test Case 9	Checking duplicate values for 5 lettered word	1. Enter the word in the textbox 2. Click on Enter	Displays an message in the textbox indicating please enter 5 lettered word	The error gets display as please enter a 5 lettered word	14th January 2023	Pass	
Test Case 10	Checking duplicate values for 6 lettered word	1. Enter the word in the textbox 2. Click on Enter	Displays an message in the textbox indicating please enter 6 lettered word	The error gets display as please enter a 6 lettered word	14th January 2023	Pass	
Test Case 11	RoughArea	1. Enter the inputs for work out of word	On entering the rough area stays constant and not gets erased on page refresh	The rough area works fine and the inputs are not erased	14th January 2023	Pass	
Test Case 12	Winner	1. Enter the word in the textbox 2. Click on Enter	On entering the correct word the page will get redirected to the endgame page	The page gets redirected to the endgame page and a pop up appears with word, description and its phonetic	14th January 2023	Pass	

Test Case 13	Loser	1. Enter the word in the textbox 2. Click on Enter	If the opponent has completed the game prior to the player then message pop up	When the Opponent player wins the game before the player pop up message appears with Better luck next time and the word, its description and phonetic	14th January 2023	Pass	
Test Case 14	One player quits or leaves	1. Enter the word in the textbox 2. Click on Enter	The game will end and the other player will get message that the opponent has left the game	The pop up appears with better luck next time and message that the opponent has left the game	14th January 2023	Pass	

TEST CASE			
System Name:	Bears & Bulls		
Module Code:	SK05 - Leaderboard		
Pass	Number of test cases passed	Pending	Number of test cases pending
Fail	Number of test cases failed	Number of test cases:	10

ID	Test Case Description	Test Case Procedure	Expected Output	Actual Output	Test Date	Result	Note
Test Case 1	LeaderBoard for Easy and 4 lettered	Go to view leaderBoard	The leader gets displayed on the leaderboard with its username and number of turns	The username and number of turns gets updated	15th January 2023	Pass	
Test Case 2	LeaderBoard for Easy and 5 lettered	Go to view leaderBoard	The leader gets displayed on the leaderboard with its username and number of turns	The username and number of turns gets updated	15th January 2023	Pass	
Test Case 3	LeaderBoard for Easy and 6 lettered	Go to view leaderBoard	The leader gets displayed on the leaderboard with its username and number of turns	The username and number of turns gets updated	15th January 2023	Pass	
Test Case 4	LeaderBoard for Medium and 4 lettered	Go to view leaderBoard	The leader gets displayed on the leaderboard with its username and number of turns	The username and number of turns gets updated	15th January 2023	Pass	

Test Case 5	LeaderBoard for Medium and 5 lettered	Go to view leaderBoard	The leader gets displayed on the leaderboard with its username and number of turns	The username and number of turns gets updated	15th January 2023	Pass	
Test Case 6	LeaderBoard for Medium and 6 lettered	Go to view leaderBoard	The leader gets displayed on the leaderboard with its username and number of turns	The username and number of turns gets updated	15th January 2023	Pass	
Test Case 7	LeaderBoard for Hard and 4 lettered	Go to view leaderBoard	The leader gets displayed on the leaderboard with its username and number of turns	The username and number of turns gets updated	15th January 2023	Pass	
Test Case 8	LeaderBoard for Hard and 5 lettered	Go to view leaderBoard	The leader gets displayed on the leaderboard with its username and number of turns	The username and number of turns gets updated	15th January 2023	Pass	
Test Case 9	LeaderBoard for Hard and 6 lettered	Go to view leaderBoard	The leader gets displayed on the leaderboard with its username and number of turns	The username and number of turns gets updated	15th January 2023	Pass	
Test Case 10	Go back	Click go back button	On clicking Go back the page gets redirected to Endgame page	The page gets redirected to endgame	15th January 2023	Pass	

5. Modification and Expected Improvements

Future modifications of the project may include a module wherein the players themselves can choose the superword for the opponent to guess. Integrate social media sharing options to encourage players to invite their friends to play. Incorporate a timer to add a sense of urgency and make the game more exciting. Include hints or clues to assist players who may be stuck on a particular word.

Offer a variety of word categories, such as pop culture, history, and science, to keep the game fresh and interesting. There can be a model for pronunciation of the guessed word.

CHAPTER 6 : RESULTS AND DISCUSSIONS

1. Test Reports

TEST REPORT						
	Date:	10 February 2023				
	No	**Module code**	**Pass**	**Fail**	**Pending**	**Total Number of test cases**
	1	SK001 - Login And Registration	9	1	0	10
	2	SK002 - Customization of Game	10	0	0	10
	3	SK003 - Working Of Game	14	0	0	14
	4	SK004 - Working Of Game(Duo)	14	0	0	14
	5	SK005 - LeaderBoard	10	0	0	10
		Sub total	57	1	0	58
		Test coverage		**100.00** %		
		Test successful coverage		**98.28** %		

For the whole project prepare one test report which covers the following points

- Bears & Bulls
- Test Objective
    - Registration, Login and Rules Module: The working of the actual modules depend on whether the values are properly inserted and fetched from the database giving us an account for each player and also checking the rules of the game are defined accurately.
    - Customization of Game Module: This module depends on the preferences of each player and they can choose their mode according to their comfort.

    - Working of Game: The word generated on the server side is available for guessing and the overall mechanics of the game are running smoothly with maximum accuracy depends on this module of the game
    - Working of Game(Duo): For playing with other players this module of the game ensures that the words generated for both the players are the same and expectations and other warnings are carried out efficiently.
    - LeaderBoard: Ensuring that the player who guessed the word in minimum tries gets to be on the leaderboard and for each category there is a top player.
- Test Summary
    - The total modules introduced in the project were: 5
    - Test cases overall: 58
    - Out of which :

        Test Cases Passed are: 57

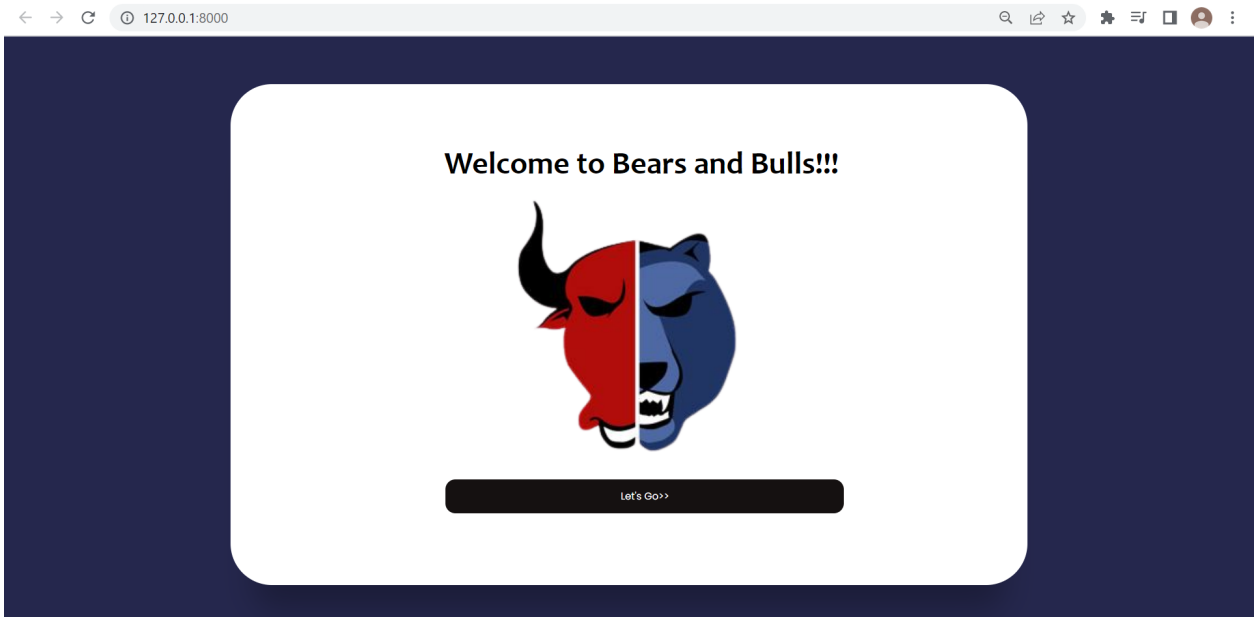
        Test Cases Failed are: 1

        Test Coverage: 100%

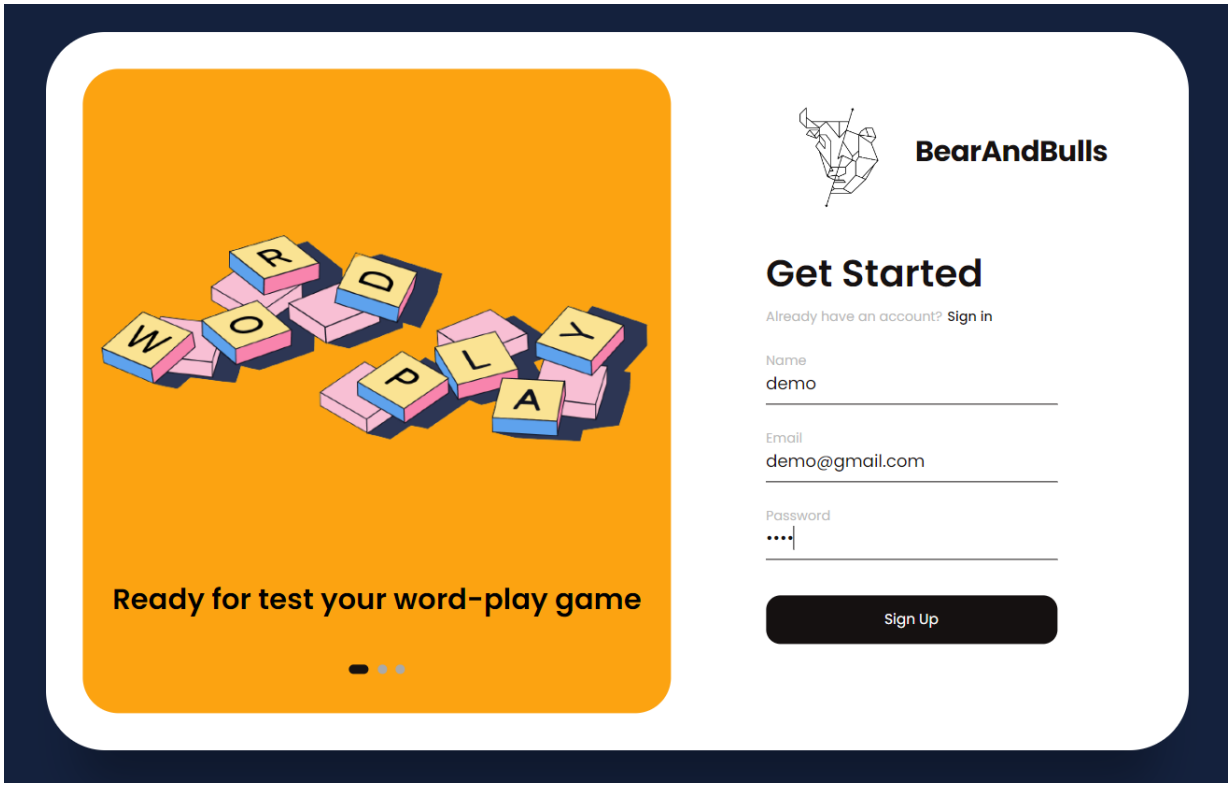
        Test successful coverage: 98.28%

2. User Documentation

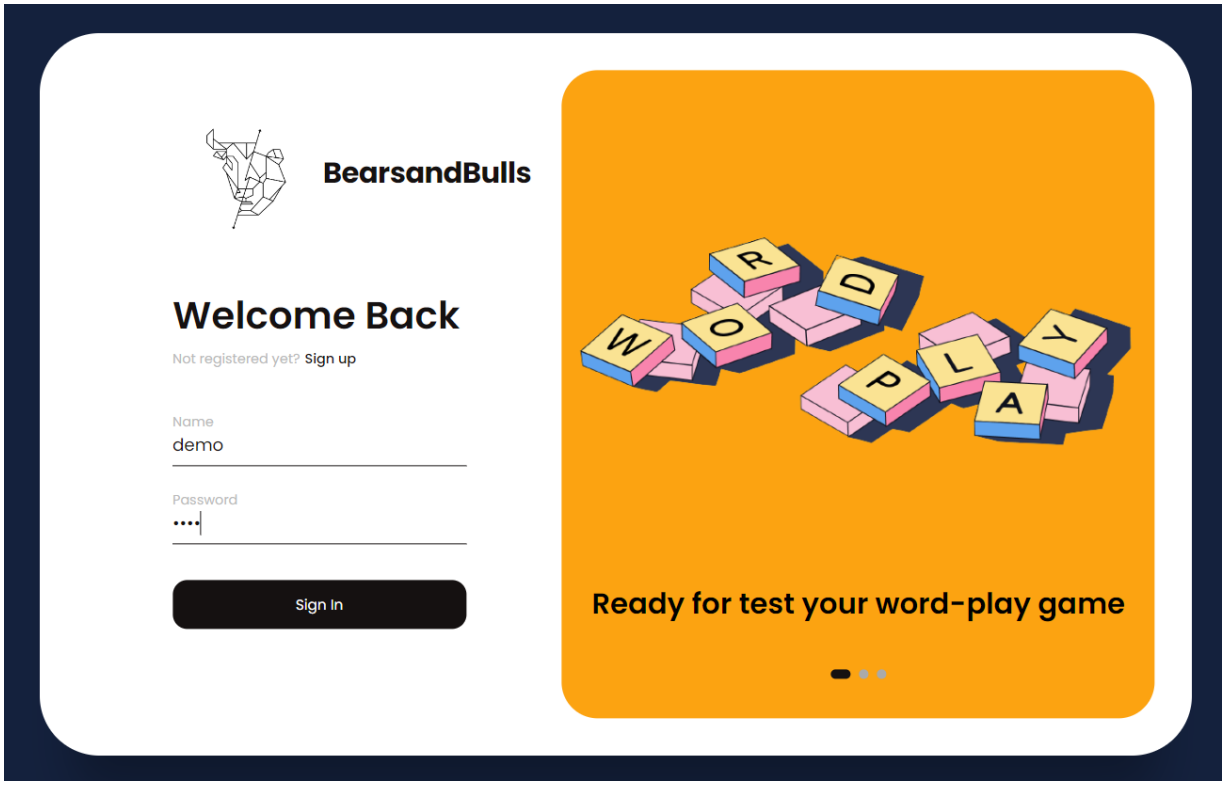
a. We will start the game by going to the url of the game.



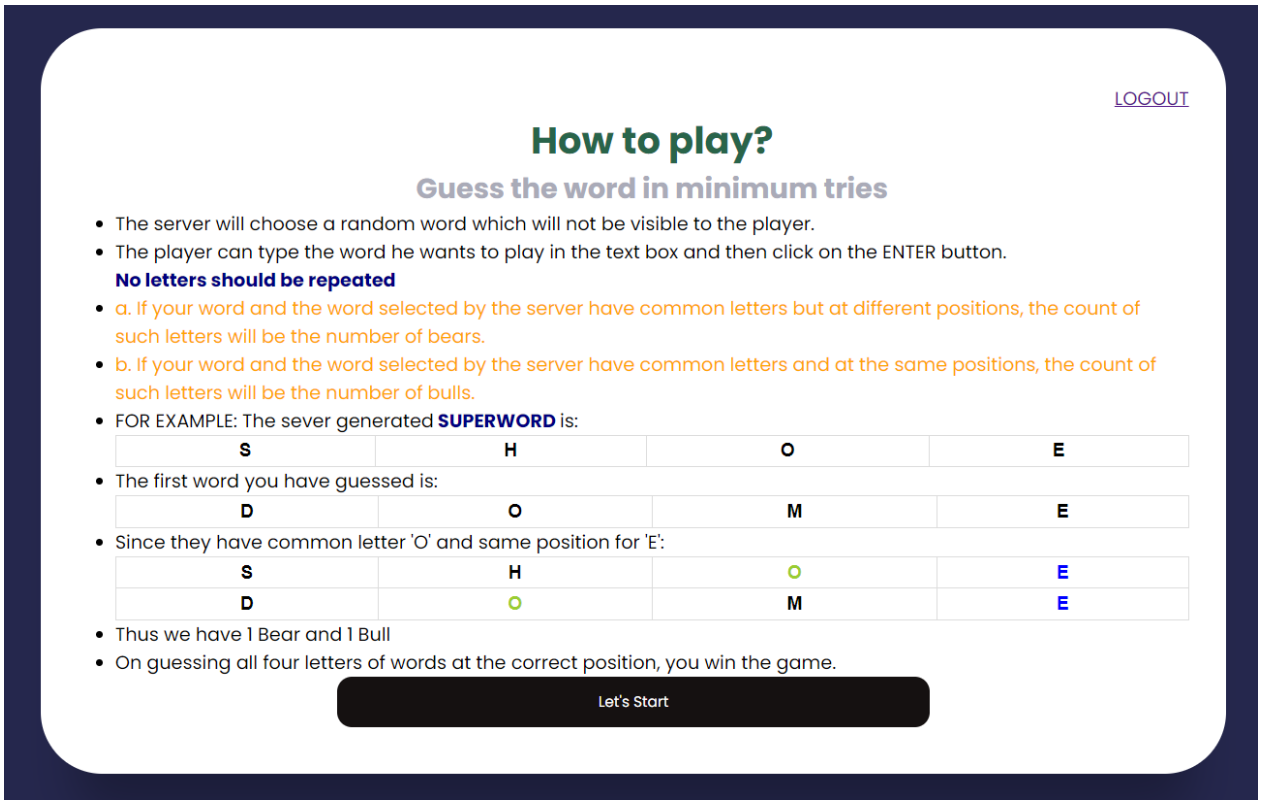
b. We click on the Let's Go button and Go to the User Registration and Login Page. Players can create their account and register themselves.



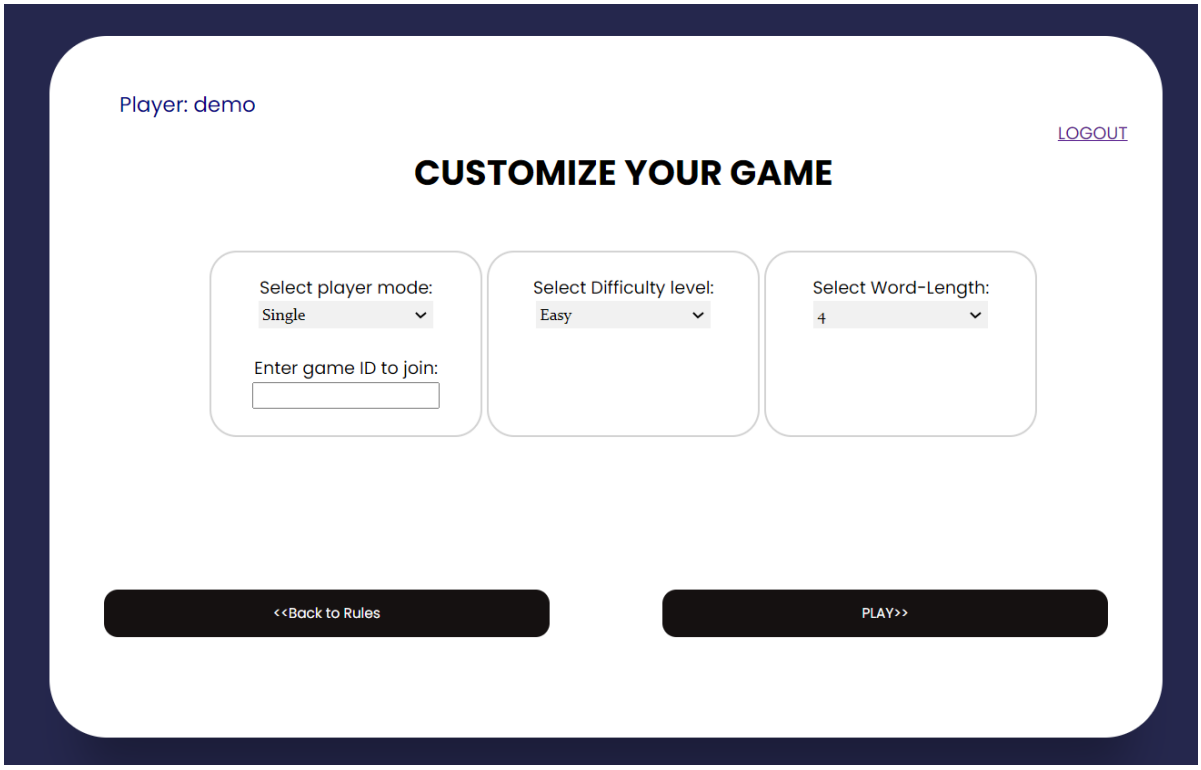
c. Add your credentials for Login and click on Sign in



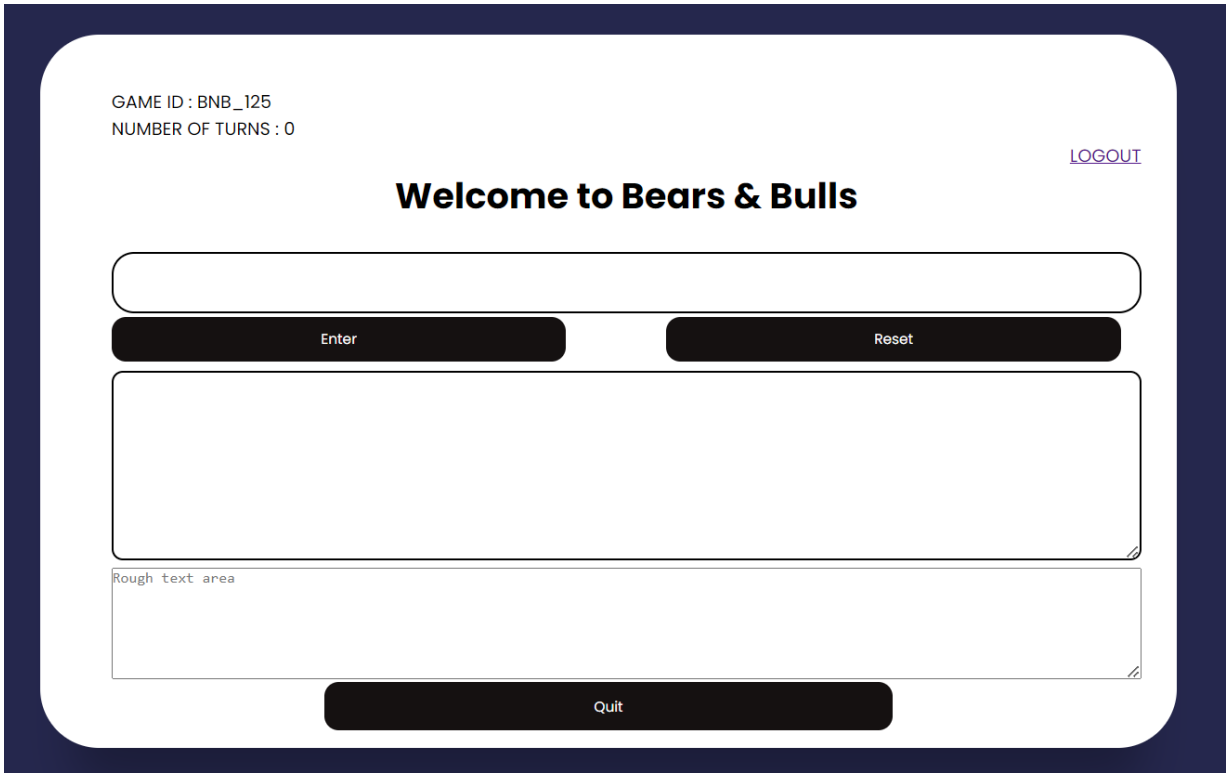
d. The player can read the rules to get familiar with the protocols of the game and click on Let's Start.



e. The player can customize their game according to their preferences. They can choose the Player mode: Single or Duo, Difficulty level: Easy, Medium or Hard, Word-Length: 4,5,6.

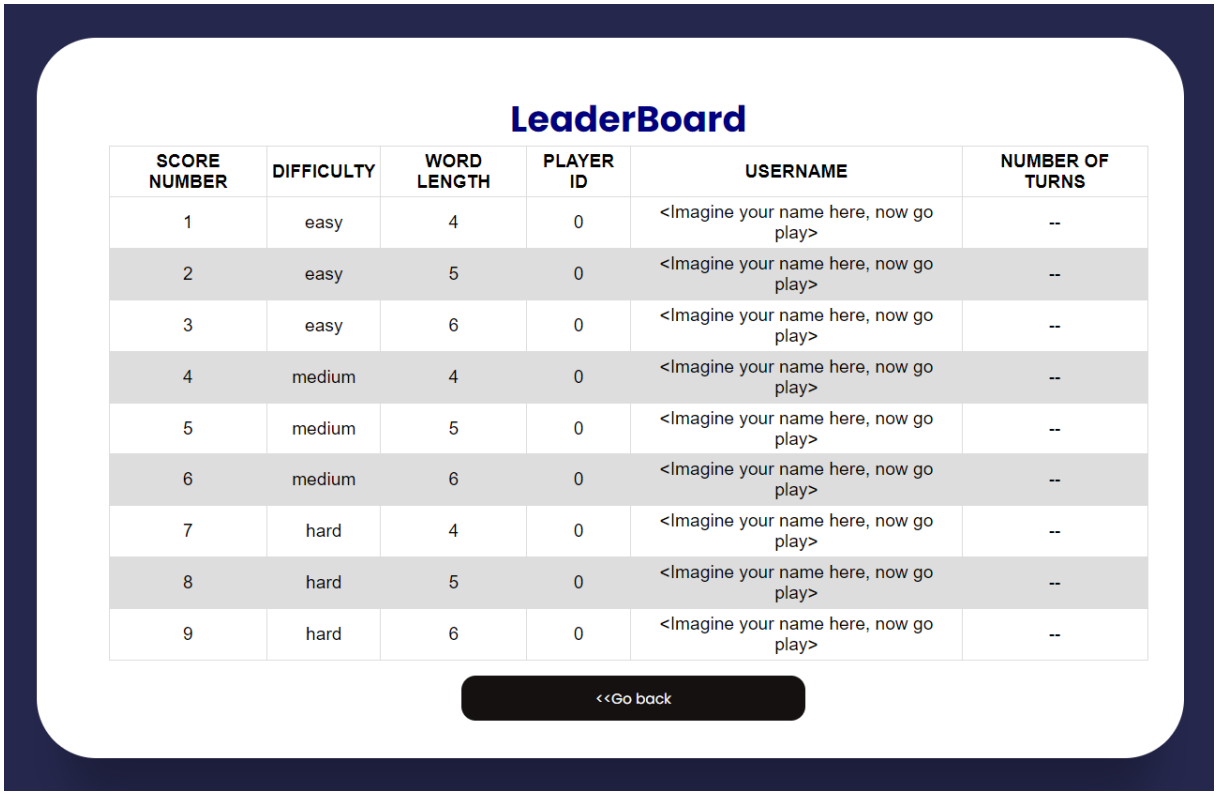


f. The player can start guessing the word and click on Enter to submit their guess. If the player wants to give up the player can press the quit button.



g. Once we have guessed the word correctly or quit the game we will be redirected to this page where we can either play the game again or view the leaderboard.

h. The leaderboard will appear like this with your name on the leaderboard.



SCORE NUMBER	DIFFICULTY	WORD LENGTH	PLAYER ID	USERNAME	NUMBER OF TURNS
1	easy	4	0	<Imagine your name here, now go play>	--
2	easy	5	0	<Imagine your name here, now go play>	--
3	easy	6	0	<Imagine your name here, now go play>	--
4	medium	4	0	<Imagine your name here, now go play>	--
5	medium	5	0	<Imagine your name here, now go play>	--
6	medium	6	0	<Imagine your name here, now go play>	--
7	hard	4	0	<Imagine your name here, now go play>	--
8	hard	5	0	<Imagine your name here, now go play>	--
9	hard	6	0	<Imagine your name here, now go play>	--

3. Cost Estimation

Cost estimation models are mathematical algorithms or parametric equations used to estimate the costs of a product or project. The results of the models are typically necessary to obtain approval to proceed, and are factored into business plans, budgets, and other financial planning and tracking mechanisms.

- **The Development Model**
  COCOMO (Constructive Cost Model) is a regression model based on LOC viz. number of Lines of Code. It is a procedural cost estimate model for software projects and often used as a process of reliably predicting the various parameters associated with making a project such as size, effort, cost, time and quality.

- **Key Parameter**
  a. Efforts - measured in person months units
  b. Schedule - measured in span of months or weeks

To estimate the effort and development time, COCOMO uses the same equations but with different coefficients (a, b, c, d in the effort and schedule equations) for each development mode. Types are as follows :

- Organic System
- Semi - detached System
- Embedded System

The basic COCOMO equations take the form

- Effort Applied (E) = ab (KLOC) bb [person-months]
  = 2.4*3 KLOC ^1.05
  =7.94 person-months
- Development Time (D) = cc (Effort Applied) dd [months]
  = 2.5 * 7.94 * 0.38
  = 7.5 months
- People Required (P) = Effort Applied / Development time [count]
  =7.94/7.5 =1.05 count

Where, KLOC is the estimated number of delivered lines (expressed in thousands) of code for a project.

The coefficient ab, bb, cb and db are given in the following table:

Software Project	a_b	b_b	c_c	d_d
Organic	2.4	1.05	2.5	0.38
Semi-detached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

COCOMO Model for "Bears&Bulls"

a. Effort : 7.94 person-months
b. Time for development : 7.5 months

Where,

Effort = Number of staff months (SM)

Size = Number of source lines of code

Time = Total number of months required to complete the project

The Project Code for MyProjectName application contains 0000 Lines of code

Since, we know that 1000 Lines of Code = 1 KLOC (K - Kilo - 10^3)

Therefore, the project consists of 3 KLOC.

Effort = = 7.94 SM

Time for development = = 7.5 Months

Cost per Month = Rs.1000/-

Total Cost of the Project = Cost per Month * Time required for the development project

= 1000 * 7

= Rs. 7,000

CHAPTER 7 : CONCLUSIONS

1. Conclusion

In conclusion, the development of this word play game has been a challenging and rewarding experience. The game is designed to improve vocabulary and language skills through an interactive and engaging gameplay experience. The game's algorithm ensures that the players are challenged and the level of difficulty increases as they progress through the game.

The game can be used as a tool for educational purposes and can be easily integrated into language learning programs. The game has been tested and validated to ensure that it is suitable for a wide range of users, from children to adults.

The game has some limitations as discussed in the previous sections, such as vocabulary limitations, complexity limitations and technical limitations. However, the future scope of the game is promising as technology continues to advance and more people are looking for ways to improve their language skills. Word play games have the potential to become an even more effective and engaging tool for improving vocabulary and language skills in the future.
Overall, this project has been a valuable learning experience and the final product is a fun and effective tool for improving vocabulary and language skills.

2. Limitations

Each project has its own limitations which we can overcome in future. Some of such limitations related to my game are:

- **Vocabulary limitations:**
  The game may have a limited number of words or may be restricted to a specific language, which could limit the game's appeal to players who are not fluent in that language.
- **Complexity limitations:**
  The game's mechanics and difficulty level may not be suitable for all players, which could limit the game's appeal to a specific group of players.
- **Platform limitations:**
  The game may be developed for a specific platform, such as mobile devices or personal computers, which could limit the game's accessibility to players who do not have access to that platform.
- **Learning limitations:**
  The game may not be effective in teaching certain aspects of language, such as grammar or sentence structure, which could limit its usefulness as an educational tool.

- **Technical limitations:**

  The game may have technical issues such as bugs or errors that could hinder the player's experience.

It is important to keep these limitations in mind during the development process, and take steps to mitigate or address them in order to create a successful and appealing game for players.

3. Future Scope of the Project

There are several areas where a Bears&Bulls could be expanded or improved in the future:

Vocabulary expansion: The game could include more words and languages to appeal to a wider audience.

Gameplay enhancement: The game could include additional mechanics or features to make it more challenging and engaging for players.

Platform integration: The game could be made available on a wider range of platforms, such as gaming consoles or virtual reality devices.

Analytics: The game could incorporate analytics to track player's progress and provide feedback.

Social integration: The game could be integrated with social media platforms to allow players to share their progress and challenge friends.

All this could be done to make the game more accessible, engaging, and useful for players, and will increase the popularity and success of the game.The future of Bears&Bulls games is promising as technology continues to advance and more people are looking for ways to improve their language skills.

REFERENCES

[1] https://www.researchgate.net/publication/337252791_THE_IMPLEMENTATION_OF_GUESSNG_GAME_IN_TEACHING_SPEAKING_AT_TENTH_GRADE_OF_SENIOR_HIGH_SCHOOL
[2] https://blog.filmsupply.com/articles/wordle-the-ultimate-case-study-for-a-passion-project/
[3]Non-functional requirements that influence gaming experience: A survey on gamers satisfaction factors[**Apostolos Ampatzoglou**]
[4]User requirements analysis for educational games in manufacturing
[5]Online Games and Security Issues
[6] Django Software Foundation. (2019). *Django*. Retrieved from https://djangoproject.com
[7] Hipp, R. D. (2020). *SQLite*. Retrieved from https://www.sqlite.org/index.html
[8] Van Rossum, G., & Drake, F. L. (2009). *Python 3 Reference Manual*. Scotts Valley.
[9] Ireland, Christopher Bowers, David Newton, Mike Waugh, Kevin. (2009). Understanding object-relational mapping: A framework based approach. Int J Adv Softw. 2.