A Project Report
on

Hangman Game Using DSA Concepts

Submitted by

Arin kulshreshtha (RA2211026030009)
Aniket Routray(RA2211026030018)
Baladitya(RA2211026030019)

Under the Supervision of
Ms.Nidhi Pandey

(Assistant Professor, Department of Computer Science and Engineering)

SRM
INSTITUTE OF SCIENCE AND TECHNOLOGY
(Deemed to be University u/s 3 of UGC Act, 1956)
DELHI-NCR CAMPUS, GHAZIABAD (U.P)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
FACULTY OF ENGINEERING & TECHNOLOGY
SRM INSTITUTE OF SCIENCE & TECHNOLOGY

SRM
INSTITUTE OF SCIENCE AND TECHNOLOGY
(Deemed to be University u/s 3 of UGC Act, 1956)
DELHI-NCR CAMPUS, GHAZIABAD (U.P.)

BONAFIDE CERTIFICATE

Arin kulshreshtha (RA2211026030009)
Aniket Routray (RA2211026030018)
Baladitya(RA2211026030019)

Certified to be the bonafide record of project Hangman game work done by Arin kulshreshtha(RA2211026030009),Aniket Routray (RA2211026030018),Baladitya(RA22110260300019) Of 3RD Semester,2ND year B.TECH degree course in SRM INSTITUTE OF SCIENCE & TECHNOLOGY, DELHI-NCR Campus for the Department of Computer Science &Engineering, in Programming for Operating System, during the academic year 2023-24.

Faculty-In-Charge

Akash Pun
Head of the department

SRM Institute of Science & Technology
HOD
(CSE)
NCR Campus, Ghaziabad

Submitted for end semester examination held on / / at SRM INSTITUTE OF SCIENCE & TECHNOLOGY, DELHI-NCR Campus.

Topic- Hangman Game Project Report

Utilizing Data Structures And Algorithms

Abstract-

Project goals-

- **The primary goal of this project was to create a Hangman game that utilizes Data Structures and Algorithms (DSA) concepts.**
- **The game aimed to offer an engaging user experience while demonstrating fundamental DSA principles.**

Design-

- **The game was designed to randomly select a secret word from a predefined wordlist.**
- **It employed data structures like arrays to maintain the secret word and guessed word.**
- **The core algorithm allowed players to guess letters and checked whether they were present in the secret word.**

Implementation-

- **The implementation was carried out in C, using random word selection, player input, and game logic.**
- **A wordlist containing words related to programming and computer science was used.**
- **The program tracked the number of attempts, displayed the current state of the word, and provided feedback to the player.**

Key Findings-

- **The game successfully integrated DSA concepts like arrays, loops, and conditional statements.**
- **It provided an interactive and enjoyable gaming experience while reinforcing fundamental DSA knowledge.**
- **The project demonstrated how DSA principles can be applied in a practical context, serving as a learning tool and entertainment medium.**

Introduction-

The purpose and objectives of a project involving a Hangman game can vary depending on the context and goals of the project. Here's a general explanation of the purpose and common objectives.

Purpose:

The purpose of the Hangman game project is to create a fun and educational interactive game while simultaneously demonstrating the practical application of Data Structures and Algorithms (DSA) concepts. The project aims to merge entertainment with learning by implementing a classic word-guessing game while emphasizing the use of DSA principles.

Objectives:

1. **Educational Enhancement: The primary objective is to provide a platform for learning and reinforcing fundamental DSA concepts. By building the game, participants, developers, or learners can gain hands-on experience in implementing DSA-related logic.**
2. **Entertainment: The project's objective is to create an engaging and enjoyable gaming experience for users. The Hangman game is a popular word game that can captivate and amuse players, promoting interaction and entertainment.**
3. **Demonstration of Data Structures: The project aims to demonstrate how data structures like arrays, strings, or linked lists can be effectively used in a real-world application. It showcases how these data structures help manage game data, such as the secret word and player's guessed letters.**
4. **Algorithm Implementation: One of the core objectives is to implement algorithms that govern the game's logic. These algorithms should include word selection, user input processing, and the mechanism for determining wins and losses.**
5. **Problem-Solving Skills: By developing the game, participants can enhance their problem-solving skills as they address challenges related to word selection, word comparison, and user interaction.**

Importance of Hangman Game-

1. **Problem-Solving Skills:**

   • Algorithm Development: Building a Hangman game involves designing and implementing algorithms for word selection, letter checking, and game state management. Developing these algorithms enhances problem-solving skills and algorithmic thinking.

2. **Programming Practice:**

   • Coding Proficiency: Implementing a Hangman game is a practical exercise in coding. It allows individuals to practice programming concepts, improve their coding skills, and learn how to structure and manage code in a real-world project.

3. **Entertainment and Engagement:**

   • Fun Learning: The Hangman game is not only educational but also entertaining. It engages players, making learning a more enjoyable experience. This can be particularly effective in educational settings where students may be motivated to learn while having fun.

Overview-

The importance of implementing a Hangman game lies in its ability to merge education with entertainment, reinforce DSA concepts through practical application, and enhance problem-solving and programming skills. It serves as a valuable tool for learning, practicing, and demonstrating the relevance of DSA in software development.

Problem Statement-

-Defining the problem we aimed to address with Hangman Game

1.Educational Reinforcement: The primary issue is the need to reinforce DSA concepts through practical application and interactive learning. Students and individuals studying DSA need a medium to apply theoretical knowledge to real-world scenarios, enhancing their understanding and retention of these concepts.

2.Interactive Learning: Traditional classroom instruction can be supplemented with interactive and engaging educational tools. The Hangman game aims to provide an environment where students and aspiring programmers can learn while having fun.

3.Coding Practice: Implementing the game involves coding exercises that enhance programming proficiency. Developers practice coding skills, learn how to structure and manage code effectively, and gain experience in creating and managing a software project.

Role of DSA in the Hangman Game-

1.Word Selection: Data Structures are used to store the list of possible words that the player must guess. The algorithm for word selection involves DSA concepts to efficiently choose a word from the list. Data structures like arrays or linked lists are employed to manage and access the wordlist. Algorithms for random word selection use concepts such as random number generation and data structure traversal.

2.Game Logic: The game logic itself relies on algorithms that dictate how the game proceeds. DSA concepts are used to manage game state, check for wins and losses, and control the flow of the game.

3.Efficiency: **DSA principles are employed to ensure the game's efficiency. Algorithms are designed to minimize the time and space complexity of tasks like checking guesses, verifying word completion, and managing data structures.**

Uses of Arrays, Lists and Trees in Hangman Game-

1. Arrays:

- **Wordlist: An array is a fundamental data structure for storing the wordlist from which the secret word is selected. Each element of the array represents a possible word that the player must guess. Arrays provide easy access to individual words and facilitate random word selection. They are efficient for this purpose, as you can directly access words by their index, and word selection is as simple as generating a random index.**
- **Secret Word: Another array is used to store the secret word that the player is trying to guess. The secret word is represented as an array of characters or letters, where each letter is stored at a specific index in the array. As the player guesses letters, this array is updated to reveal correctly guessed letters while keeping unknown letters as placeholders (e.g., "_ _ A _ _"). Arrays allow for efficient and straightforward manipulation of the secret word.**
- **Guessed Letters: An array or list can be used to keep track of the letters that the player has guessed. This data structure allows the game to efficiently check whether a guessed letter is correct, display guessed letters to the player, and prevent duplicate guesses.**

2. Lists:

- **Guessed Letters: While arrays can be used to store guessed letters efficiently, lists, such as linked lists or dynamic arrays, offer flexibility. Lists can dynamically grow as the player makes guesses, making them suitable for scenarios where the number of guesses is not predetermined. They provide ease in adding and removing elements, which can be useful for managing the list of guessed letters.**
- **Previous Guesses: Lists can also be used to maintain a history of previous guesses. This is not essential for the core gameplay but can be a useful feature for player feedback and tracking.**

3. Trees (less common):

• Trees are generally not a common data structure choice for Hangman game implementations because they are not the most efficient choice for managing wordlists, word representations, or guessed letters. However, they could be used in certain scenarios, such as advanced game variants or educational exercises to showcase tree data structures.

- Most Hangman game implementations, the combination of arrays and lists is sufficient to handle the game's requirements efficiently. Arrays are particularly essential for managing wordlists and secret words, while lists provide flexibility for maintaining guessed letters and game history. The choice of data structures depends on the specific goals and complexity of the Hangman game implementation.

Flow of the Hangman Game-

Overall game design-

Game Elements:

- Secret Word: The secret word, which the player must guess, is initially hidden with blanks or underscores to represent the unknown letters.
- Hangman Figure: A visual representation of the hangman is gradually revealed as the player makes incorrect guesses. It typically includes the gallows, a head, body, arms, and legs.
- Guessed Letters: A section of the screen displays the letters the player has guessed. Correct guesses are displayed in their respective positions in the secret word, and incorrect guesses are marked as incorrect.

Game Logic:

- Word Selection: At the start of each game, a random word is selected from a predefined wordlist. This word is the secret word that the player needs to guess.

- **Player Interaction: The player interacts with the game by entering letter guesses. The game logic checks if the guessed letter is in the secret word and updates the game accordingly.**

- **Game State: The game maintains the current state, including the number of incorrect guesses, the status of the hangman figure, the revealed part of the secret word, and the guessed letters.**

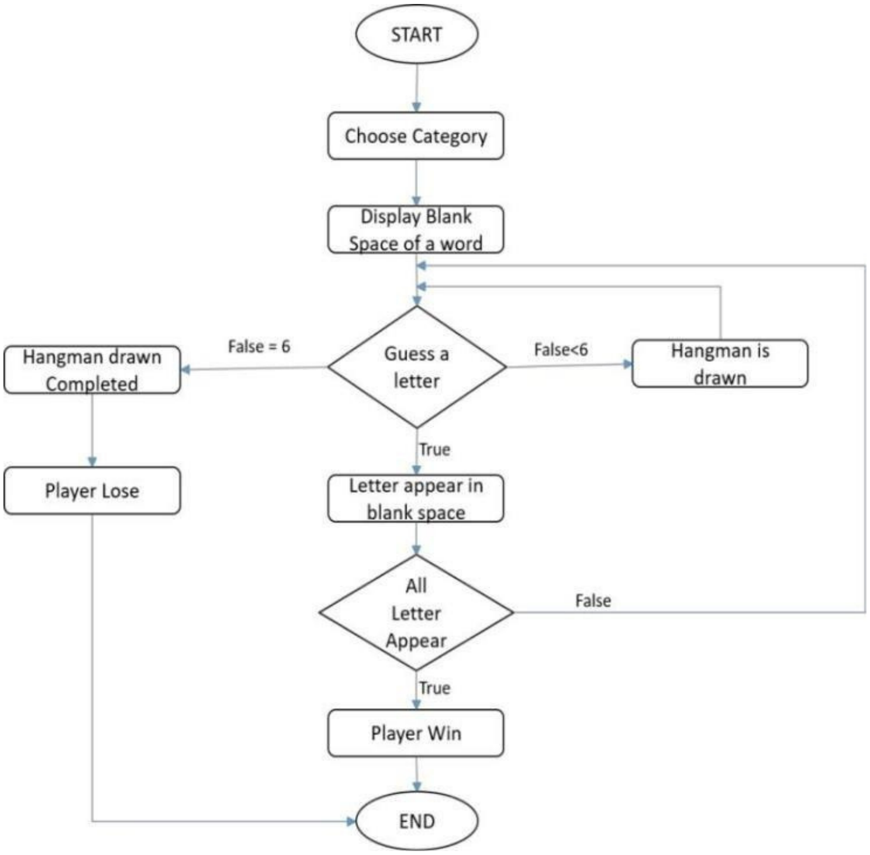
- **Win/Loss Conditions: The game logic determines when the player wins (guesses the word correctly) or loses (makes a certain number of incorrect guesses and the hangman is fully revealed).**

Game Flow:

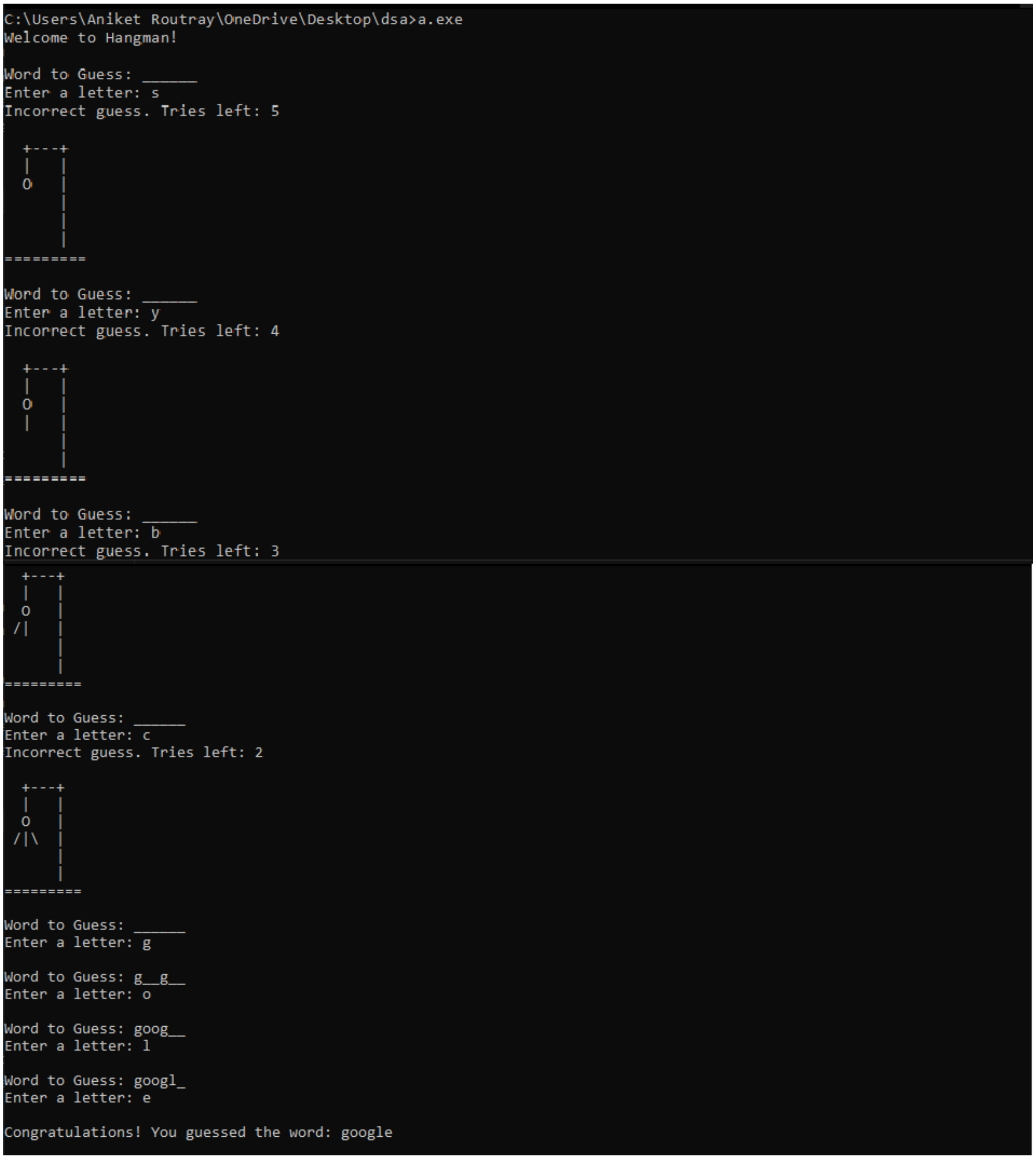
- **The game follows a specific flow, starting with word selection, player interactions, game logic updates, feedback, and eventual determination of a win or loss.**

- **A game menu and replay options are typically included to facilitate multiple rounds.**

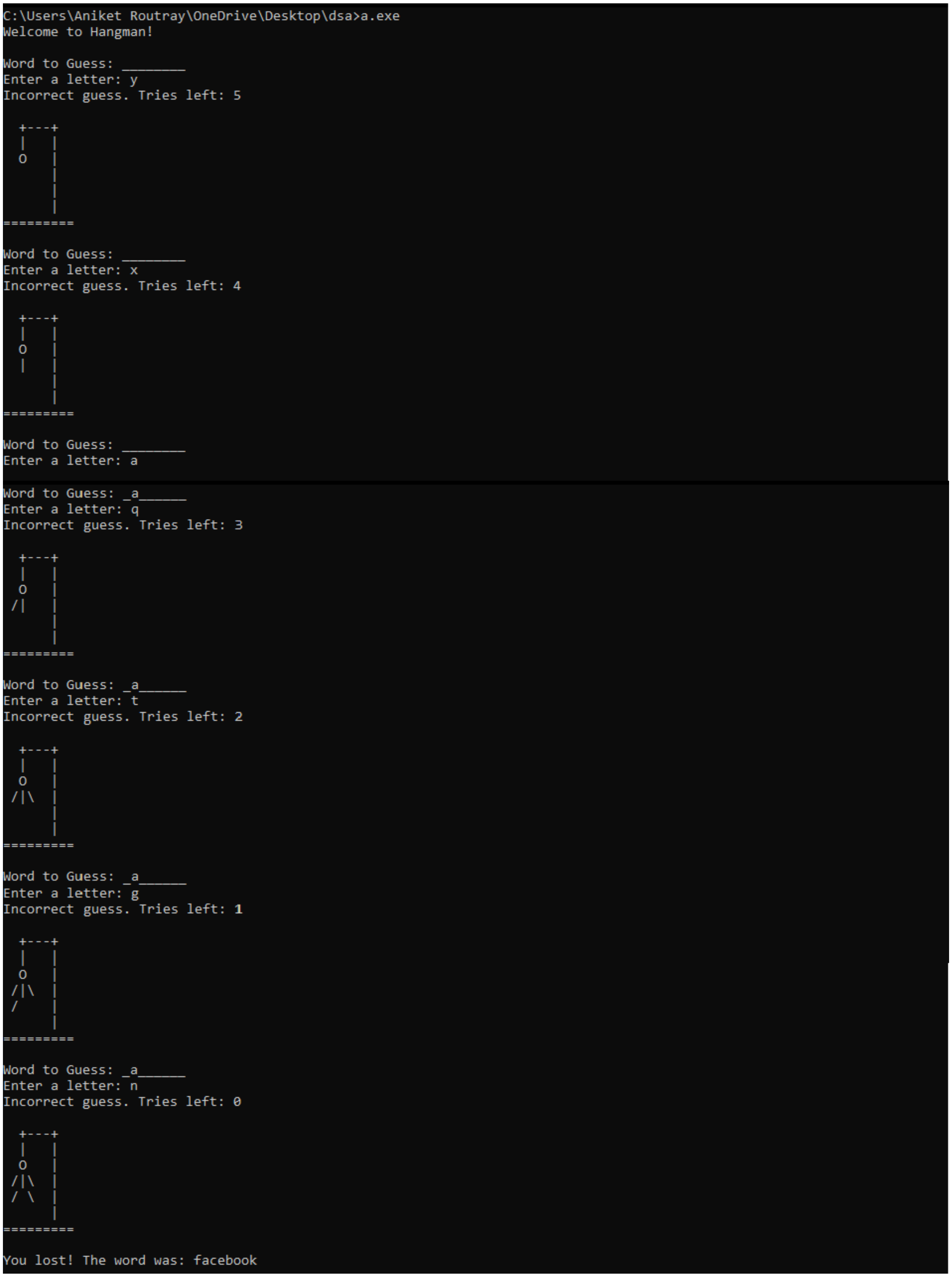
Flow Chart-



Output-



```
C:\Users\Aniket Routray\OneDrive\Desktop\dsa>a.exe
Welcome to Hangman!

Word to Guess: ______
Enter a letter: s
Incorrect guess. Tries left: 5

  +---+
  |   |
  O   |
      |
      |
      |
=========

Word to Guess: ______
Enter a letter: y
Incorrect guess. Tries left: 4

  +---+
  |   |
  O   |
  |   |
      |
      |
=========

Word to Guess: ______
Enter a letter: b
Incorrect guess. Tries left: 3
  +---+
  |   |
  O   |
 /|   |
      |
      |
=========

Word to Guess: ______
Enter a letter: c
Incorrect guess. Tries left: 2

  +---+
  |   |
  O   |
 /|\  |
      |
      |
=========

Word to Guess: ______
Enter a letter: g

Word to Guess: g__g__
Enter a letter: o

Word to Guess: goog__
Enter a letter: l

Word to Guess: googl_
Enter a letter: e

Congratulations! You guessed the word: google
```



```
C:\Users\Aniket Routray\OneDrive\Desktop\dsa>a.exe
Welcome to Hangman!

Word to Guess: ________
Enter a letter: y
Incorrect guess. Tries left: 5

  +---+
  |   |
  O   |
      |
      |
      |
=========

Word to Guess: ________
Enter a letter: x
Incorrect guess. Tries left: 4

  +---+
  |   |
  O   |
  |   |
      |
      |
=========

Word to Guess: ________
Enter a letter: a
Word to Guess: _a______
Enter a letter: q
Incorrect guess. Tries left: 3

  +---+
  |   |
  O   |
 /|   |
      |
      |
=========

Word to Guess: _a______
Enter a letter: t
Incorrect guess. Tries left: 2

  +---+
  |   |
  O   |
 /|\  |
      |
      |
=========

Word to Guess: _a______
Enter a letter: g
Incorrect guess. Tries left: 1

  +---+
  |   |
  O   |
 /|\  |
 /    |
      |
=========

Word to Guess: _a______
Enter a letter: n
Incorrect guess. Tries left: 0

  +---+
  |   |
  O   |
 /|\  |
 / \  |
      |
=========

You lost! The word was: facebook
```

Conclusion-

Key Points:

1. **Educational Integration: The project effectively integrated educational elements, providing a practical platform for reinforcing Data Structures and Algorithms (DSA) concepts. It showcased the application of DSA principles in real-world game development.**
2. **Entertainment and Engagement: The Hangman game design successfully combined education with entertainment. It offered an engaging and enjoyable gaming experience, encouraging players to learn and interact with the game.**
3. **Algorithm Implementation: The project showcased the development of various algorithms for word selection, letter checking, and game logic. These algorithms promoted algorithmic thinking and problem-solving skills.**

Achievements:

1. **Interactive Learning: The Hangman game successfully provided an interactive learning experience, allowing players to apply and reinforce DSA concepts while enjoying the game.**
2. **Algorithmic Thinking: The development of the game promoted algorithmic thinking, improving problem-solving skills. It showcased how algorithms are used to manage word selection, word representation, and letter checking.**
3. **Practical Application of DSA: The project demonstrated the practical application of DSA knowledge in software development. It illustrated the direct relevance of data structures and algorithms in solving real-world problems and enhancing the gameplay experience.**