

# **PROJECT REPORT**

## **Guess the Number Game – Using Prolog**

---

### **Submitted By**

**Name:** Shivani Chandrakant Bande  
**Roll No.:** 164  
**Branch:** Computer Science & Engineering  
**Subject:** Artificial Intelligence  
**Date:** 26/11/2025

---

### **1. Title of the Project**

#### **Guess the Number Game in Prolog**

---

### **2. Abstract**

This project demonstrates a simple interactive number-guessing game using SWI-Prolog. The system generates a random secret number between 1 and 10 and asks the user to guess it. Based on the user input, the system provides hints such as “Too Low” or “Too High” until the correct number is guessed. This project explains the use of recursion, input/output predicates, and decision rules in Prolog programming.

---

### **3. Objectives**

- To demonstrate interactive programming in Prolog.
  - To understand the use of recursion and logical decision rules.
  - To implement AI-based reasoning using rule-based logic.
  - To learn Prolog user input/output handling.
- 

### **4. Software and Tools Used**

Category	Description
Programming Language	Prolog
IDE / Interpreter	SWI-Prolog 9.2.9
Operating System	Windows / Linux
File Name	guess.pl
Method Used	Rule-based & Recursive Decision Making

---

## 5. Working Principle

1. The program welcomes the user and explains the game rules.
  2. Prolog randomly selects a number between 1 and 10.
  3. User enters a number guess.
  4. Prolog compares input with the secret number.
  5. If guess is wrong, it gives hints (Too High / Too Low).
  6. Loop continues until correct number is guessed.
  7. Program ends with a success message.
- 

## 6. Flow of Execution

```

Start
↓
Generate random number (1-10)
↓
Prompt user for input
↓
User enters a guess
↓
Compare guess with secret number
  ┌─ If correct → Display Congratulations → End
  └─ If too low → Display "Too low" → Ask again
      └─ If too high → Display "Too high" → Ask again
↓
End

```

---

## 7. Code Implementation

```

:- use_module(library(random)).

start :-
    write('-----'), nl,
    write('      GUESS THE NUMBER GAME'), nl,
    write('-----'), nl,
    write('I have selected a number between 1 and 10.'), nl,

```

```

write('Try to guess it!'), nl,
random(1, 11, Number),    % Random number generated
guess(Number).

guess(Number) :-
    write('Enter your guess: '),
    read(Guess),
    check(Guess, Number).

check(Guess, Number) :-
    Guess == Number,
    write('□ Congratulations! You guessed the correct number!'), nl,
    write('Game Over.'), !.

check(Guess, Number) :-
    Guess < Number,
    write('Too low! Try again.'), nl,
    guess(Number).

check(Guess, Number) :-
    Guess > Number,
    write('Too high! Try again.'), nl,
    guess(Number).

```

---

## 8. Execution Commands

Open SWI-Prolog and type:

```

?- ['E:/guess.pl'].
?- start.

```

---

## 9. Sample Output

```

-----
GUESS THE NUMBER GAME
-----
I have selected a number between 1 and 10.
Try to guess it!
Enter your guess: 4.
Too low! Try again.
Enter your guess: 9.
Too high! Try again.
Enter your guess: 7.
□ Congratulations! You guessed the correct number!
Game Over.

```

---

## 10. Conclusion

The Guess the Number game helps understand decision-making logic, recursion, and user interaction in Prolog. It demonstrates how AI rule-based systems work using simple comparisons and looping mechanisms. This is an excellent beginner project to learn Prolog fundamentals.

---

## 11. Future Enhancements

- Add score counter and track number of attempts.
- Difficulty levels (1–50, 1–100 range).
- GUI version using Python or JavaScript.
- Multiplayer mode.