# ****Project Report: AI Code Breaker****

**Project Title:** AI Code Breaker – Number Guessing Game  
**Submitted By:** Ali Raza Shaikh 22k-4313 ( Leader )

Haseeb Mujtaba 22k-4307

Murtaza Johar 22k-4508  
**Course:** AI  
**Instructor:** [Abdullah Yaqoob]  
**Submission Date:** 11-May-2025

## ****1. Executive Summary****

**Project Overview:**  
This project presents a modified version of the classic number guessing game, designed as a competition between a human player and an AI. The primary goal is to test the capabilities of an AI model built to deduce a secret 4-digit number using rule-based logic and eliminate invalid guesses based on feedback. The AI strategy focuses on intelligent filtering and deduction using feedback such as digit correctness and position accuracy.

## ****2. Introduction****

**Background:**  
The number guessing game is a classic logic puzzle involving the identification of a hidden number through trial and error. This project builds on the conventional version by adding a competitive AI opponent that simulates human-like reasoning in narrowing down potential correct answers.

**Objectives of the Project:**

* Develop an interactive, web-based number guessing game
* Implement AI logic that mimics deductive human reasoning
* Evaluate the AI's performance against a human player
* Enhance understanding of feedback-based AI decision-making

## ****3. Game Description****

**Original Game Rules:**  
In the traditional number guessing game, players attempt to guess a secret number by submitting multiple guesses and receiving feedback on correctness.

**Innovations and Modifications:**  
This version introduces a 2-player system (human vs AI), and adds a feedback loop to guide the guessing strategy. The AI continuously learns which digits are correct, incorrect, or in the correct position, refining its guesses each round.

## ****4. AI Approach and Methodology****

**AI Techniques Used:**  
A rule-based logic system was used to identify and eliminate invalid guesses. Unlike search-tree algorithms like Minimax, this AI evaluates past results to prune the guess space.

**Algorithm and Heuristic Design:**  
The AI tracks:

* Digits confirmed to be in the number (good\_digits)
* Digits not present (bad\_digits)
* Known correct positions (current\_positions)

It refines guesses based on feedback like the number of correct digits and correct positions.

**AI Performance Evaluation:**  
Performance was evaluated based on the number of rounds taken to guess correctly. In test runs, the AI guessed the correct number in fewer turns than random guessing, showing logical learning over time.

## ****5. Game Mechanics and Rules****

**Modified Game Rules:**  
The game uses a 4-digit number with all unique digits. Each guess is validated, and feedback is provided on digit and position correctness.

**Turn-based Mechanics:**  
Turns alternate between the human and AI. Both receive feedback, and the first to guess correctly wins.

**Winning Conditions:**  
The first player to correctly guess the number with all digits in the correct position wins the game.

## ****6. Implementation and Development****

**Development Process:**  
The project was built using Flask for the web framework. The backend handles AI logic, game state, and player input. Frontend rendering and game interface were implemented using HTML with Jinja2 templating.

**Programming Languages and Tools:**

* **Programming Language:** Python
* **Libraries:** Flask, random
* **Tools:** Browser for UI, GitHub for version control

**Challenges Encountered:**  
Key challenges included designing efficient AI logic that avoids repeated guesses and correctly infers digit placement based on minimal feedback. Handling game state persistence and session management was also a learning opportunity.

## ****7. Team Contributions****

* **]Ali Raza:** AI logic implementation,
* **Haseeb Mujtaba** :Flask backend development,
* **Murtaza Johar:** overall project integration

## ****8. Results and Discussion****

**AI Performance:**  
The AI consistently improved its guessing strategy with each attempt. Compared to a human player, it showed strong performance by reducing the guess space logically. It demonstrated efficiency in identifying correct digits and optimizing guesses accordingly.

## ****9. References****

* Python random module documentation
* AI logic inspiration from logical deduction puzzle strategies
* Stack Overflow discussions and examples