**Project proposal**

**Project Proposal: Hangman Game using Python GUI**

**Student Details:**

**Hammad khan**

**Roll No:237714**

**1. Introduction**

The Hangman Game is a classic word-guessing game where the player attempts to deduce a hidden word by guessing one letter at a time. This project aims to develop a GUI-based Hangman game using Python and the Tkinter library. The game will provide an interactive and educational experience, enhancing vocabulary and spelling skills while demonstrating core programming concepts such as object-oriented programming (OOP) and graphical user interfaces (GUI).

**2. Background**

Hangman has been a popular educational tool and leisure activity for decades. Traditionally played with pen and paper, it involves guessing letters to reveal a word. Incorrect guesses result in parts of a stick figure being drawn, with the goal of guessing the word before the figure is completed. This project translates the traditional game into a digital format, leveraging Python's Tkinter library to create a user-friendly and engaging application.

**3. Problem Statement**

The traditional Hangman game, while simple and fun, lacks the convenience and accessibility of a digital version. There is a need for an easily accessible, interactive, and user-friendly Hangman game that can be played on personal computers. The game should include features such as:

* A graphical interface for ease of use.
* Dynamic word selection from a predefined list.
* Real-time feedback on guesses.
* Persistent game state and score tracking.

**4. Libraries and Technologies**

To develop this project, the following libraries and technologies will be used:

* **Python**: The core programming language for the project.
* **Tkinter**: A built-in Python library for creating graphical user interfaces.
* **Random**: A standard Python library module for generating random choices (used for selecting words).

**5. Solution**

The proposed solution is a Python-based Hangman game implemented with Tkinter for the GUI. The key features and components include:

* **Word Selection**: A predefined list of words from which a random word is selected at the start of each game.
* **Game Interface**: A user-friendly interface displaying the hidden word, input for guessing letters, and feedback on remaining guesses.
* **Game Logic**: Handles user inputs, updates the display based on correct or incorrect guesses, and manages the game state.
* **Feedback Mechanism**: Provides warnings for invalid inputs (non-alphabetical characters or multiple characters), duplicate guesses, and displays win/loss messages.

The game will have the following classes and methods:

* **HangmanGame Class**:
  + \_\_init\_\_: Initializes the game state and sets up the GUI.
  + create\_widgets: Creates and places the GUI components.
  + make\_guess: Handles the logic for processing a player's guess and updates the game state accordingly.

**6. Source of Idea**

The idea for this project was inspired by a search for popular Python GUI projects on Google. Various online resources and tutorials were referenced to understand the implementation of the Hangman game in a GUI environment using the Tkinter library.