Beginner-Level Python Projects: Features & Fundamentals

# 1. Number Guessing Game

## Features:

* • Generates a random number between a range (e.g., 1–100)
* • User inputs guesses
* • Provides hints: “Too high” or “Too low”
* • Tracks the number of attempts
* • Option to play again

## Fundamentals Used:

* • `input()` function
* • `random.randint()`
* • Loops (`while`)
* • Conditional statements (`if`, `else`)
* • Variables and data types

# 2. Simple Calculator

## Features:

* • Performs basic arithmetic: +, –, ×, ÷
* • Takes two inputs and an operation
* • Option to perform more calculations or exit
* • Handles division by zero errors

## Fundamentals Used:

* • Functions (`def`)
* • User input
* • Operators (`+`, `-`, `\*`, `/`)
* • Conditional statements
* • Exception handling (`try-except`)

# 3. To-Do List (Console)

## Features:

* • Add new tasks
* • View all tasks
* • Mark tasks as done
* • Delete tasks
* • Save/load tasks to a file

## Fundamentals Used:

* • Lists
* • File handling (`open()`, `.read()`, `.write()`)
* • String methods
* • Loops
* • Functions

# 4. Dice Roller Simulator

## Features:

* • Simulates rolling a 6-sided die (or custom number of sides)
* • Randomly displays number after each roll
* • Option to roll again

## Fundamentals Used:

* • `random.randint()`
* • Loops (`while`)
* • User input
* • Functions

# 5. Mad Libs Generator

## Features:

* • Takes various word inputs (noun, verb, adjective)
* • Inserts them into a prewritten story template
* • Displays the completed story

## Fundamentals Used:

* • Strings and concatenation
* • `input()` function
* • Lists
* • Basic formatting

# 6. Password Generator

## Features:

* • Generates secure passwords of desired length
* • Uses uppercase, lowercase, digits, and symbols
* • Option to copy to clipboard (optional with `pyperclip`)

## Fundamentals Used:

* • `random.choice()` or `random.sample()`
* • `string` module (`string.ascii\_letters`, `string.digits`, etc.)
* • Loops
* • Functions

# 7. Simple Contact Book

## Features:

* • Add, search, and delete contacts
* • Store data in a dictionary or text file
* • List all saved contacts

## Fundamentals Used:

* • Dictionaries
* • File handling
* • Functions
* • Conditional statements
* • Loops

# 8. Currency Converter

## Features:

* • Converts currency based on user input
* • Uses static rates or fetches live rates via an API (optional)
* • Display conversion results
* • Menu-driven options

## Fundamentals Used:

* • Arithmetic operations
* • Dictionaries (for storing rates)
* • Optional: API requests (`requests` library)
* • Loops and conditions

# 9. Flashcard Quiz App

## Features:

* • Ask multiple-choice or true/false questions
* • Show correct answers
* • Keep score
* • Option to replay

## Fundamentals Used:

* • Lists/dictionaries
* • Loops
* • Functions
* • Conditional statements

# 10. Alarm Clock

## Features:

* • Set an alarm for a specific time
* • Alert (with sound or print) when time is reached
* • Option to stop alarm

## Fundamentals Used:

* • `datetime` and `time` modules
* • Conditional statements
* • Loops
* • Optional: Play sound using `playsound` or `winsound`

# 11. Unit Converter (with GUI)

## Features:

* • Convert units (e.g., km ↔ miles, kg ↔ pounds)
* • Dropdown menus for units
* • Input field and result display

## Fundamentals Used:

* • Tkinter (GUI library)
* • Functions
* • Conditional logic
* • Basic math

# 12. Basic Notepad App

## Features:

* • Create/edit/delete text files
* • Open/save files
* • Basic UI for text editing

## Fundamentals Used:

* • Tkinter
* • File handling
* • GUI events (buttons, menus)
* • Functions

# 13. Stopwatch/Timer (GUI or Console)

## Features:

* • Start, pause, reset stopwatch
* • Countdown timer option
* • Display time in real-time

## Fundamentals Used:

* • `time.sleep()`
* • Loops
* • Tkinter (for GUI)
* • Functions