Desktop

Chatbot

- **Objective**: Create a basic chatbot that can answer user queries or have a conversation with users.
- **Libraries**: `nltk`, `chatterbot` or ChatGPT similar API.
- **Enhancements**: Integrate with a GUI, or make it a Discord bot, or even implement some machine learning algorithms to make it more interactive.

Personal Diary System

- **Objective**: Create a secure, password-protected personal diary system where users can write, read, and edit their diaries.
- **Libraries**: `tkinter` for GUI, `cryptography` for encryption.
- **Enhancements**: Implement a search feature, back up diaries to the cloud, or send daily email reminders to write a diary.

Personal Finance Tracker

- **Objective**: Create an app to track expenses and income, with categories and budget limits.
- **Libraries**: `tkinter` for GUI, `matplotlib` for visualizing expenses.
- **Enhancements**: Implement features to export data, analyze spending trends, and send budget alerts.

Scraping

Instagram Photo Downloader

- **Objective**: Develop a script that can download Instagram photos of a public account to your computer.
- **Libraries**: `beautifulsoup4`, `requests`.
- **Enhancements**: Develop a GUI, enable downloading of videos, add options to download from hashtags.

Automated Job Application Bot

- **Objective**: Create a bot that automatically fills out job applications or sends emails to job postings found online.
- **Libraries**: `selenium` for browser automation, `beautifulsoup4` for web scraping.
- **Enhancements**: Implement customization options for job search criteria, add a GUI for user input.

E-Commerce Price Tracker

- **Objective**: Create a scraper that tracks the prices of specific products on e-commerce websites and notifies you when the prices drop.
- **Skills Practiced**: BeautifulSoup, requests, sending emails using Python.
- **Enhancements**: Add the ability to track prices on multiple websites or track multiple products.

News Aggregator

- **Objective**: Scrape news articles from various websites and display them in one place, with links to the original articles.
- **Skills Practiced**: BeautifulSoup, requests, Flask for displaying aggregated news.
- **Enhancements**: Categorize news by topics, add a search functionality.

Advanced scraping libraries: scrappy, selenium.

API

Data Visualization: Plotting Weather Data

- **Project**: Plotting Weather Data of a City
- **Objective**: Use a weather API to fetch weather data and plot it using a library like matplotlib.
- **Tools & Libraries**: `matplotlib`, `requests`
- **Enhancements**: Add functionality to compare the weather of two cities and visualize the comparison.

Twitter Bot

- **Objective**: Create a bot that automatically posts tweets or replies to tweets on Twitter.
- **Libraries**: `Tweepy` or Twitter API.
- **Enhancements**: Make the bot respond to mentions, retweet posts based on specific hashtags, or implement sentiment analysis.

Cryptocurrency Tracker

- **Objective**: Develop an application to track cryptocurrency prices in real-time.
- **Libraries**: `tkinter` for GUI, `requests` to fetch data from a cryptocurrency API.
- **Enhancements**: Add features like price alerts, portfolio tracking, and historical price charts.

Games

Snake Game

- **Objective**: Create the classic Snake Game where the player controls a snake to consume food and grow.
- **Libraries**: `pygame`.
- **Enhancements**: Add different levels, customize snake and food appearance, implement a scoring system.

Text-Based Hangman

- **Project**: Create a version of the game Hangman.
- **Objective**: Let the user guess letters to figure out a secret word.
- **Tools & Libraries**: Built-in Python libraries.
- **Enhancements**: Extend the game by adding different categories of words or implementing a scoring system.

Interactive Quiz with GUI

- **Objective**: Create an interactive multiple-choice quiz game with a graphical user interface.
- **Libraries**: `tkinter` for GUI.
- **Enhancements**: Implement a scoring system, add a timer, and provide immediate feedback on correct/incorrect answers.

Space Invaders

- **Objective**: Remake the classic Space Invaders game where the player controls a spaceship to defend against waves of aliens.
- **Libraries**: `pygame`.
- **Enhancements**: Add different levels, customize ship and alien appearance, implement power-ups and scoring system.

Online Multiplayer Game

- **Objective**: Develop a simple online multiplayer game like Tic Tac Toe.
- **Libraries**: `pygame` for game development, `socket` for networking.
- **Enhancements**: Implement chat functionality, allow more than two players, add different game modes.

Web

Flask Blog

- **Project**: Simple Blog
- **Objective**: Create, Read, Update, and Delete blog posts.

- **Tools & Libraries**: `Flask`, `SQLite`
- **Enhancements**: Implement user authentication and styling the blog using HTML/CSS.

To-Do List with Flask

- **Objective**: Develop a to-do list where users can add and delete tasks.
- **Frontend**: HTML, CSS, JavaScript.
- **Backend**: `Flask`, `SQLite`.
- **Enhancements**: Add user authentication and the ability for users to have their individual to-do lists.

Weather Dashboard with Flask

- **Objective**: A simple dashboard to show the weather using a public API.
- **Frontend**: HTML, CSS.
- **Backend**: `Flask`, `requests` to handle API requests.
- **Enhancements**: Allow users to search for the weather in different locations.

Polling App

- **Objective**: Create a simple app where users can create polls and vote.
- **Frontend**: HTML, CSS.
- **Backend**: `Flask`, `SQLite`.
- **Enhancements**: Add user authentication, allow users to view poll results in different formats (bar chart, pie chart, etc.).

Simple E-Commerce Platform

- **Objective**: A platform where users can view products and add them to their cart.
- **Frontend**: HTML, CSS, Bootstrap.
- **Backend**: `Flask`, `SQLite`.
- **Enhancements**: Add user authentication, allow users to checkout and place orders.

Advanced web libraries: django.