### **Python Mastery Challenge**

#### **Objective:**

The goal is to create a series of increasingly complex projects that incorporate key Python concepts and skills. These projects will span from beginner to advanced levels, ensuring a strong foundational understanding while pushing the limits of what you can build with Python.

#### **Phase 1: Fundamentals (2 Weeks)**

1. **Project 1: Number Guessing Game**
   * Develop a simple terminal-based game where the user guesses a random number generated by the computer. Include hints such as "higher" or "lower".
   * Skills: Variables, conditionals, loops, user input.
2. **Project 2: Personal Finance Tracker**
   * Create a program that allows users to input their daily expenses and savings. Calculate the monthly total and display a breakdown of categories.
   * Skills: Lists, dictionaries, functions, basic file handling.
3. **Project 3: To-Do List Application**
   * Build a command-line to-do list application with features like adding tasks, marking them as complete, and viewing pending tasks.
   * Skills: CRUD operations (Create, Read, Update, Delete), file handling, classes.

#### **Phase 2: Intermediate (4 Weeks)**

1. **Project 4: Web Scraper**
   * Develop a script to scrape data from a website (e.g., job postings, news headlines) and store the results in a CSV file. Implement error handling for failed requests.
   * Skills: requests library, BeautifulSoup, CSV handling, exception handling.
2. **Project 5: Data Visualization**
   * Load a dataset from a CSV file (e.g., weather data, sales data) and create various charts (bar, line, scatter) to visualize trends and insights using Matplotlib or Plotly.
   * Skills: Pandas, Matplotlib/Plotly, data cleaning.
3. **Project 6: RESTful API**
   * Build a RESTful API using Flask that performs CRUD operations on a dataset (e.g., book collection). Include endpoints for adding, retrieving, updating, and deleting records.
   * Skills: Flask, HTTP methods, JSON, Postman.
4. **Project 7: Chatbot**
   * Create a simple rule-based chatbot that responds to basic questions and stores the chat history in a text file.
   * Skills: Conditionals, file handling, string manipulation, functions.

#### **Phase 3: Advanced (6 Weeks)**

1. **Project 8: Machine Learning Model**
   * Train a machine learning model on a dataset (e.g., Iris dataset for classification or Boston housing for regression) and evaluate its performance. Create visualizations to explain model predictions.
   * Skills: Scikit-Learn, data preprocessing, train/test split, model evaluation, visualization.
2. **Project 9: Web Application**
   * Develop a web application using Flask and integrate it with a front-end framework like Bootstrap. Allow users to interact with a database (e.g., user registration and login system).
   * Skills: Flask, Jinja templating, SQLAlchemy, Bootstrap, authentication.
3. **Project 10: Automation Script**
   * Automate a daily task, such as downloading files, sending emails, or interacting with APIs (e.g., fetching daily weather or stock data).
   * Skills: smtplib, schedule or APScheduler, requests.
4. **Project 11: Multi-threaded Program**
   * Build a multi-threaded file downloader that can handle multiple files simultaneously, displaying download progress.
   * Skills: threading or concurrent.futures, file I/O, status handling.

#### **Phase 4: Specialization (Optional, 4 Weeks)**

1. **Specialization Project: Data Pipeline**
   * Build a complete data pipeline that extracts data from an API or web source, processes and transforms it, loads it into a database, and visualizes the results in a dashboard.
   * Skills: ETL (Extract, Transform, Load), SQLAlchemy, Airflow, Matplotlib/Seaborn, Flask/Django.
2. **Specialization Project: AI-powered Application**
   * Develop an AI-powered application, such as a sentiment analysis tool, image classifier, or recommendation system. Integrate the model into a web or desktop application.
   * Skills: TensorFlow/PyTorch, Flask/Streamlit, front-end integration, model deployment.

### **Resources:**

* **Books**: *Python Crash Course* by Eric Matthes, *Automate the Boring Stuff with Python* by Al Sweigart.
* **Online Platforms**: Codecademy, LeetCode, HackerRank, Kaggle.
* **Documentation**: [Python.org](https://docs.python.org/3/), [Real Python](https://realpython.com/), GeeksforGeeks.

### **Evaluation & Learning Path:**

Document each project, including a brief summary, challenges faced, solutions implemented, and what you learned. Push all projects to a GitHub repository and create a README file for each project detailing its functionality and key concepts.

**Goal**: By the end of this challenge, you should have a solid understanding of Python, hands-on experience with various libraries and frameworks, and a portfolio that showcases your progression from beginner to advanced Python projects.