### 1. ****Building a Financial Calculator (1 TO 4)****

* Create a financial calculator that computes mortgage payments, loan interest, and amortization schedules based on user inputs.

### 2. ****Implementing a Simple Voting System (5 TO 8)****

* Design a voting system where users can submit votes for candidates, tally the votes, and declare a winner.

### 3. ****Web Scraping (with Simple HTTP Requests) (9 TO 12)****

* Implement a basic web scraper that can fetch data from a webpage, clean the content, and store relevant data.

### 4. ****Customer Invoice System (13 TO 16)****

* Develop a system to generate customer invoices by applying discounts, calculating taxes, and generating the final payable amount.

### 5. ****Data Encryption and Decryption (Basic Cryptography)(17 TO 20)****

* Write functions to perform basic encryption (e.g., Caesar Cipher) and decryption for securing messages.

### 6. ****Generating Reports for a Warehouse System (21 TO 24)****

* Create a function that reads data about product inventories, sales, and shipments, and generates periodic reports about warehouse stock levels.

### 7. ****Movie Recommendation System (25 TO 28)****

* Implement a simple recommendation system that takes user preferences and suggests movies based on similar ratings or genres.

### 8. ****Real-time Weather Data Processing (29 TO 32)****

* Fetch and process real-time weather data from an API, and generate summaries or alerts for users based on certain weather conditions.

### 9. ****Online Store Cart System (33 TO 36)****

* Build a shopping cart system where users can add, remove, or update items, and calculate the total cost, taxes, and applicable discounts.

### 10. ****Credit Card Validation (37 TO 40)****

* Write a system that validates credit card numbers using the Luhn algorithm to check whether a card number is valid.

### 11. ****Stock Portfolio Analyzer (41 TO 44)****

* Develop a tool to analyze a user’s stock portfolio, calculate returns, evaluate portfolio performance, and suggest rebalancing strategies.

### 12. ****Real-time Currency Converter (45 TO 48)****

* Build a currency converter that fetches real-time exchange rates and allows users to convert between different currencies.

### ****Bank Transaction Reconciliation (49 TO 52)****

* Implement a reconciliation system to match bank transactions with invoices, ensuring that every transaction is accounted for.

### 14. ****Inventory Restocking System (53 TO 56)****

* Create a function that monitors inventory levels in a store or warehouse, triggering restocking orders when the stock falls below a certain threshold.

### 15. ****Travel Route Planner (57 TO 60)****

* Develop a simple route planner for a travel application, calculating the optimal routes between two locations based on distance or travel time.

### 16. ****Ticket Reservation System (61 TO 64)****

* Build a basic reservation system for events like movie tickets or concerts, allowing users to reserve, modify, or cancel bookings.

### 17. ****Email Automation System (65 TO 68)****

* Design an email automation system that can send reminders, confirmations, or newsletters based on user actions or preset schedules.

### 18. ****Real-time Chat Application (69 TO 72)****

* Create the backend logic for a real-time chat system, handling user messages and broadcasting them to all participants.(basic implementation only)

### 19. ****Password Strength Checker (73 TO 76)****

* Write a program that evaluates the strength of user passwords based on criteria such as length, complexity, and inclusion of special characters.

### 20. ****Basic Health Tracker (77 TO 80)****

* Develop a simple health tracker that takes inputs like daily steps, water intake, and calories consumed, and provides suggestions for improvement.