Sr No.	List of Python Practicals
1.	Write a Python Program for Electricity Bill Calculation where take input as Total Units Consumed and Apply following Conditions: • 0-100 units: ₹5/unit • 101-300 units: ₹7/unit • >300 units: ₹10/unit And Display Total Bill to the user.
2.	Write a Python Program for Password Validation System with following conditions: • Allow the users 3 attempts to enter the correct password. • If the password matches -> print "Login Successful" and break the loop. • If all attempts fail -> print "Account Locked".
3.	Write a Python Program for Shopping Cart System with following conditions: • Given a list of products with prices. • Skip products with price 0 using continue. • If product is "Exit" → stop scanning using break. • If all items are scanned successfully → print a message from the else block. • Use pass for future discount implementation.
4.	Write a Python Program Library Book Management with following conditions: • Use range to generate unique book IDs (101–110). • Traverse a string to check if the book title contains vowels. • Traverse a list of book titles to display available books. • Traverse a dictionary with {Book: Author} to display book-author pairs.
5.	Write a Python Program for ATM Cash Withdrawal Simulation with following conditions: • Input the withdrawal amount. • Check if the balance is sufficient. • If amount is not multiple of 100 → show an error. • Use loops to allow multiple transactions until the user exits.
6.	Write a Python Program for Student Marks Processing with following conditions: • Given a list of marks of students. • Use list comprehension to find marks > 40 (pass students). • Use dictionary comprehension to create {student_name: grade}. • Use set comprehension to find unique grades. • Use tuple comprehension (generator expression) to store squares of marks.
7.	Write a Python Program for Employee Salary Slip with following conditions: • Define a normal function to calculate HRA, DA, PF. • Define a lambda function to calculate net salary. • Show difference between user-defined and lambda function usage.

8.	Write a Python Program for Online Banking System with following conditions: • Create a decorator @authenticate_user to check if a user is logged in before performing a transaction. • Implement a generator function transaction_history() that yields last 5 transactions of a user.
9.	Write a Python Program for Scientific Calculator with following conditions: • Create a module calculator.py with functions add, subtract, multiply, divide. • Import the module in another file main.py and perform user-chosen operations.
10.	Write a Python Program for Travel Expense Calculator with following conditions: • Create two modules: ➢ transport.py → contains bus/train/flight fare functions. ➢ stay.py → contains hotel/food expense functions. • Import specific functions into main.py to calculate total trip cost.
11.	Write a Python Program for Lucky Draw Contest with following conditions: • Use random to pick 5 winners from a list of participants. • Use datetime to print the date & time of draw. • Use math to calculate prize money distribution with percentage-based bonus.
12.	Write a Python Program for Library Management System with following conditions: ➢ Create a package library with modules: ➢ books.py → add, remove, search books. ➢ members.py → register and update members. ➢ Import these modules inside main.py to simulate library operations.
13.	Write a Python Program for University Department System with following conditions: ➢ Create a namespace package university with sub-packages: ➢ science → subjects list. ➢ arts → subjects list. ➢ commerce → subjects list. ➢ Import and display subjects from different namespaces.
14.	Write a Python Program for Hospital Management System with following conditions: • Create a package hospital with modules: ○ patient.py → store patient details. ○ doctor.py → store doctor details. ○ appointment.py → book appointments. • Import and combine all to simulate hospital operations.
15.	Write a Python Program for Student Record File System with following conditions: • Try to open a file students.txt. • If the file is missing, handle exception and create a new file.

16.	Write a Python Program for Web Application Debugging with following conditions: • Create a function that logs exceptions into an error file error_log.txt. • Along with the exception, store an error code and timestamp.

17.	Write a Python Program for Railway Ticket Reservation with following conditions:
	• Create a class Railway Reservation with :
	  o Total Seats
	  o Method book_seat(user)
	• Use Multithreading to simulate multiple users trying to book at the same time.
	• Use locking to avoid
	  o OverBooking
	  o Inconsistent seat count.
	• Print success/failure for each booking attempt.

18.	Write a Python Program for Image Processing Pipeline with following conditions:
	1. Create a abstract class image task with a method process.
	2. Implement subclasses
	  I. Resize task
	  II. Filter Task
	  III. Save Task
	3. Use Thread to execute each Step concurrently.
	4. Use Thread Synchronization (Event or Queue) to ensure task running order.

19.	Write a Python Program for Stock Market Price Tracker with following conditions:
	1. Create a base class stock.
	2. Create thread for each Stock that:
	  I. Generate random Price Update.
	  II. Print Alerts.
	3. Create Stock Manager Class:
	  I. Start and Stop monitoring threads,
	4. Use a flag to gracefully stop the threads.

20.	Write a Python Program for Smart Home System with following conditions:
	1. Create Classes:
	  II. Light
	  III. AC
	  IV. Washer
	2. Each must have a run() method that prints status updates intervals.
	3. Run each appliance in a separate thread.
	4. Implement smart home hub class to:
	  I. Start/Stop all appliances.
	  II. Gather logs.

21.	Write a Python Program for University Exam Result Processing with following conditions: 1. Create answerSheet class: 2. Thread class evaluator thread:

	I. Sleeps randomly II. Assign random marks