

# **Dr. Harisingh Gour Vishwavidyalaya**

**(A Central University)**



## **Department of Computer Science and Application**

BCA 4<sup>th</sup> Semester

### **Lab Based on Python Programming**

CSA-DSM-414

Submitted to

**Dr. Pangambam Sandesh  
Singh**

Submitted By

**PARAS TAMRAKAR  
ROLL NUMBER**

**Y23170552**

# Project 1

## Student Information System Code

This Python program is a command-line based **Student Information System** that uses **MySQL** for storing and managing student academic records.

---

### Core Functionalities:

#### 1. Database Setup

- Creates two tables:
  - `students (id, name)`
  - `marks (student_id, subject, score)` with foreign key relation.

#### 2. Student Management

- Add, update, and delete student records.

#### 3. Marks Management

- Add marks for each student per subject.
- Calculates:
  - **Average score**
  - **Letter Grade** (A to F)
  - **GPA** (on a 4.0 scale)

#### 4. Student Summary

- Shows detailed performance (subjects, scores, average, grade, GPA).

#### 5. Topper Identification

- Finds the student with the highest average marks.

#### 6. CSV Operations

- **Import** students and marks from a CSV.
- **Export** all records to a CSV.

#### 7. User Interface

- Menu-driven interface for easy navigation.

---

## **Grading & GPA Logic:**

**Grade** A: 90+, B: 80–89, C: 70–79, D: 60–69, F: <60

**GPA:** A=4.0, B=3.0, C=2.0, D=1.0, F=0.0

---

# Project 2.

This Python script is an **Expense Tracker System** designed to help users log expenses, manage budgets, analyze spending habits, and visualize expense data. Here's a concise summary:

## Features:

1. **Expense Logging:** Allows users to enter expenses with details like category, amount, date, and notes.
2. **Budget Management:** Enables setting spending limits for categories and alerts when budgets are exceeded.
3. **Summarization:** Generates reports for expenses over the last 7 or 30 days, categorized by spending types.
4. **Data Export:** Exports expense records to a CSV file for external use.
5. **Visualization:** Creates a pie chart representation of expenses over the past 30 days.
6. **Interactive Menu:** Provides an intuitive console-based menu for managing expenses efficiently.

## How It Works:

- Uses `expenses.json` to store and retrieve financial data.
- Calculates and warns users when their spending exceeds a set budget.
- Offers a well-structured interface to navigate different functions.

# Project 3

## COVID Data Tracker:

### Key Features




#### 1. Data Management:

- Stores COVID-related data (cases, recoveries, deaths) in a CSV file.
- Loads existing data automatically.

#### 2. Daily Data Entry:

- Allows users to input new COVID statistics for different cities.
- Saves records for tracking pandemic history.

#### 3. Risk Zone Analysis:

- Categorizes cities based on active cases:
  -  High Risk (Active cases > 1000)
  -  Medium Risk (Active cases > 100)
  -  Low Risk (Active cases ≤ 100)

#### 4. Summary Report:

- Displays total cases, recoveries, and deaths across all cities.

#### 5. Trend Visualization:

- Generates **line graphs** showing COVID trends for a selected city.

#### 6. Hotspot Prediction:

- Identifies cities with **rapid case increases** in the last 7 days.
- Flags potential outbreak locations.

#### 7. Data Importing:

- Allows importing records from external CSV files.

#### 8. Interactive Menu:

- Provides an intuitive console-based interface for managing COVID data.

## How It Works

- Uses **Pandas** for efficient data handling.
- **Matplotlib** generates graphical representations.
- **Datetime operations** process historical data to identify trends.

This system offers a simple way to track and visualize COVID trends in different cities.

---

# Project 4.

## Library Management System – Summary

This is a Python-based console application that manages a library's book collection and borrowing records using a JSON file for data storage.

### Main Features:

- **Add Book** – Adds new books with title, author, quantity.
- **Remove Book** – Deletes a book by its ID.
- **Issue Book** – Issues a book to a borrower, updates availability and borrow records.
- **Return Book** – Returns a borrowed book and updates inventory.
- **View Inventory** – Lists all books with their available copies.
- **View Borrowers** – Shows which users have borrowed which books.
- **Usage Statistics** – Displays most borrowed books.

**Data is saved in `library_data.json`**, and all interactions are handled through a simple text-based menu.

# Project 5.

health and fitness tracker script:

## 1. Purpose:

- Tracks and logs daily health data, including steps, sleep, calories, and water intake.
- Calculates BMI and estimates daily calorie needs using user-provided details.

## 2. Features:

- **BMI Calculator:** Computes BMI based on the user's weight and height.
- **Calorie Needs Calculator:** Uses the Mifflin-St Jeor formula to estimate daily calorie requirements based on weight, age, and gender.
- **Hydration Reminder:** Reminds users to stay hydrated periodically.
- **Daily Logs:** Prompts the user to input daily health metrics and stores them in a weekly dataset.
- **Weekly Report & Graphs:** Generates a summary of the week's data and visualizes it using graphs (steps, sleep, water intake) with Matplotlib.

## 3. Interaction:

- Users can log daily health data, view weekly progress, or exit the program through a menu-driven interface.

## 4. Visualization:

- Graphs display trends for steps, sleep hours, and water intake across the week.

## 5. Implementation:

- Uses Python modules like datetime, time, and matplotlib.pyplot.
- Relies on simple data structures (e.g., dictionaries, lists) for storing and displaying data.



The program is user-friendly and methodical, allowing for both data input and analysis. It's a great way to track and visualize personal health metrics.

---