**PYTHON COMPETATIVE CODING**

**Python internship Report is submitted**

**In accordance with requirement of degree of**

**BACHELOR OF TECHNOLOGY**

**IN**

**ELECTRICAL AND ELECTRONICS ENGINEERING**

Submitted by

D.V.ANJANA DEVI

21KQ1A0204

Under the Mentorship of

**SHRAVAN MANTHRI**

** Technical Trainer**

**DEPARTMENT OF ELECTRICAL AND EL ECTRONICS ENGINEERING**

PACE INSTITUTE OF TECNOLOGY AND SCIENCES (AUTONOMOUS)

(Affiliated to Jawaharlal Nehru Technological University Kakinada, Kakinada &

Accredited by NAAC ‘A’ GRADE,An ISO 9001-2015 Certified Institution)

NH-16, Valluru Post , Prakasam District, A.P-523272.

(2024-2025)

**Scholastic Achievement Dashboard**

Date:- 10th June 2024

Submitted by:- Dande Venkata Anjana Devi (21kq1a0204)

Details of Project:- I’m implementing this project by using Python Programming Language.

DESCRIPTION:-

The purpose of the Scholastic Achievements Dashboard project is to provide an efficient and effective tool for managing and analyzing academic data within educational institution. The Scholastic Achievement Dashboard project is designed to provide a comprehensive overview of students' academic performance and progress. It typically includes features such as tracking grades, attendance records, test scores, and other relevant academic data. It can be a valuable tool for enhancing communication between teachers, students, and parents regarding educational outcomes and areas for improvement.

REQUIREMENTS:-

Database Design: Create a database to store student information. You can use a table with

columns for student name, roll number, and marks.

User Interface: Develop a user-friendly interface for data entry and display. Include

options to add new student records, update records, and view student

information.

Functionality: Implement functions to add student details, update marks, and retrieve

student information based on roll number or name.

Data Validation: Validate input data to ensure correctness. Check for valid roll numbers,

marks within a specified range, and proper formatting of names.

Dashboard Display: Design a dashboard to show student achievements, such as top

scorers, average marks, and performance.

Security: Implement security measures to protect student data, such as user

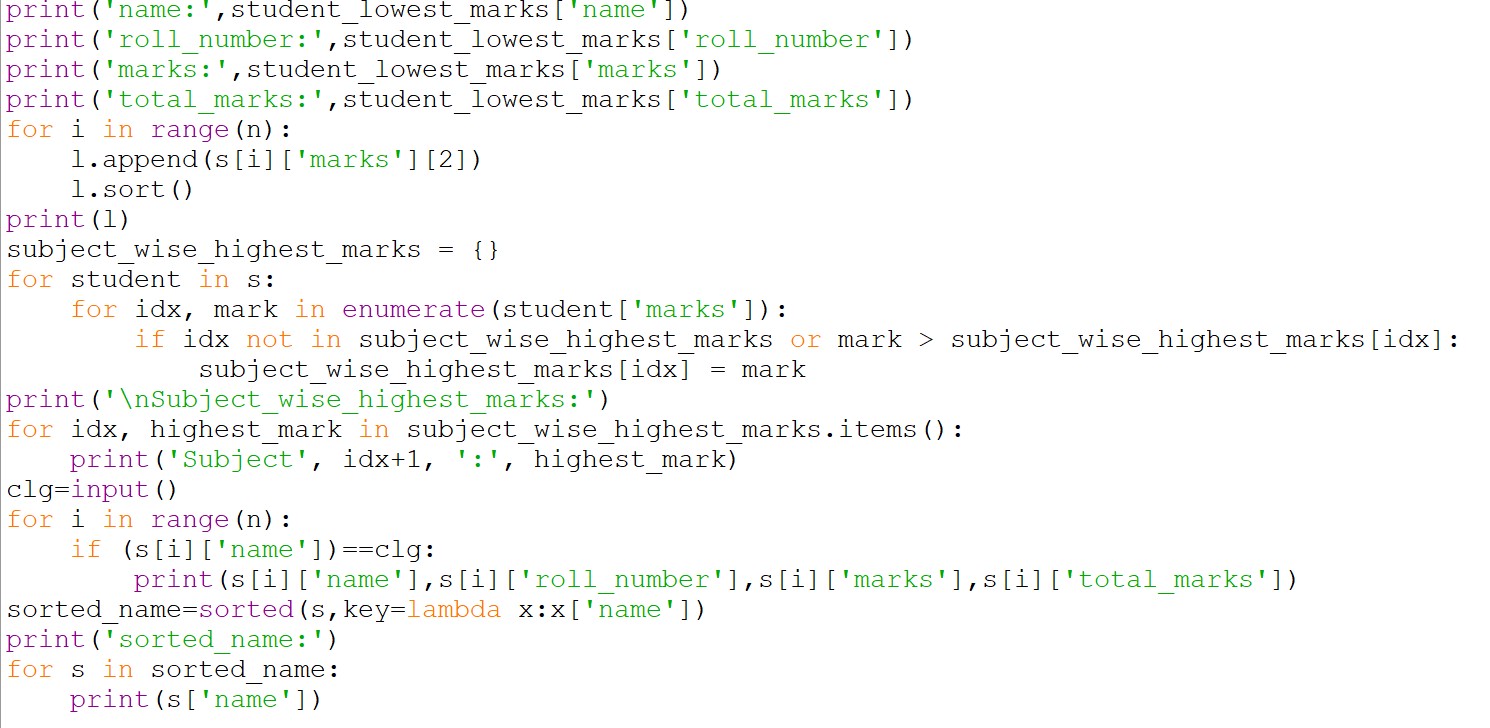
authentication, data encryption.

**Project approach:-**

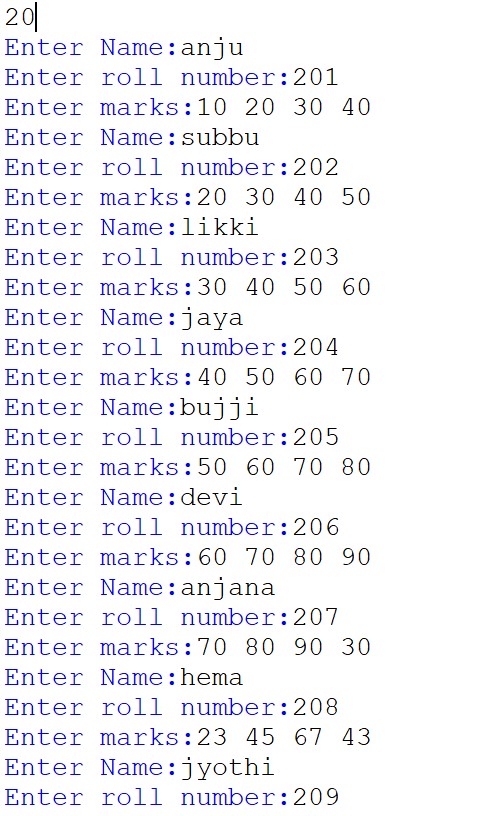
1. **Understanding Requirements:**
   * Review the project requirements to ensure alignment with the implemented code.
   * Identify additional features needed to meet the specified requirements such as user interface, data validation, security, and dashboard display.
2. **Refactor Code into Modular Components:**
   * Break down the code into modular functions or classes to improve readability, reusability, and maintainability.
   * Separate concerns such as data processing, user input handling, and output presentation.
3. **Database Integration:**
   * Integrate a database (e.g., SQLite, PostgreSQL) to store student information instead of using lists.
   * Design database tables to represent student details, marks, and other relevant data fields.
   * Refactor the add\_student function to insert data into the database instead of appending to a list.
4. **User Interface Development:**
   * Develop a user-friendly interface for data entry and display using frameworks like Tkinter for desktop applications or Flask/Django for web applications.
   * Implement forms for adding new student records, updating existing records, and viewing student information.
   * Ensure the interface follows UX/UI design principles for intuitive navigation and usability.
5. **Data Validation:**
   * Enhance data validation to ensure correctness and consistency of input data.
   * Validate roll numbers, marks, and other fields according to specified ranges and formats.
   * Implement error handling to provide informative feedback to users in case of invalid inputs.
6. **Security Implementation:**
   * Implement security measures to protect student data and application integrity.
   * Incorporate user authentication mechanisms to control access to sensitive operations and data.
   * Apply encryption techniques to secure data transmission and storage, especially for sensitive information.
7. **Dashboard Development:**
   * Design and develop a dashboard to display student achievements and performance metrics.
   * Utilize visualization libraries (e.g., Matplotlib, Plotly) to create interactive charts and graphs.
   * Display metrics such as top scorers, average marks, and subject-wise performance to provide insights into student performance.
8. **Testing and Debugging:**
   * Conduct thorough testing of each component to identify and fix bugs and errors.
   * Perform unit tests, integration tests, and system tests to ensure the application functions as expected.
   * Test edge cases and invalid inputs to verify the robustness of data validation and error handling mechanisms.
9. **Documentation and Deployment:**
   * Document the project including design decisions, implementation details, and usage instructions.
   * Prepare deployment scripts or instructions to deploy the application to a production environment.
   * Deploy the application on a suitable hosting platform with appropriate security configurations.
10. **User Training and Support:**
    * Provide user training sessions to familiarize users with the application and its features.
    * Offer ongoing support and maintenance to address user inquiries, issues, and feature requests.

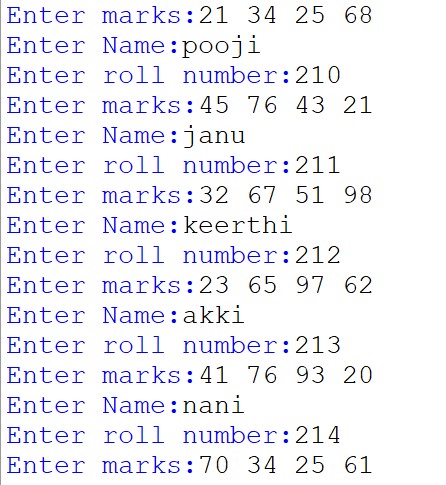
Project code:-

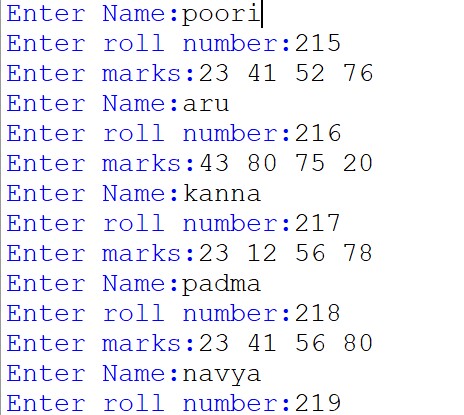


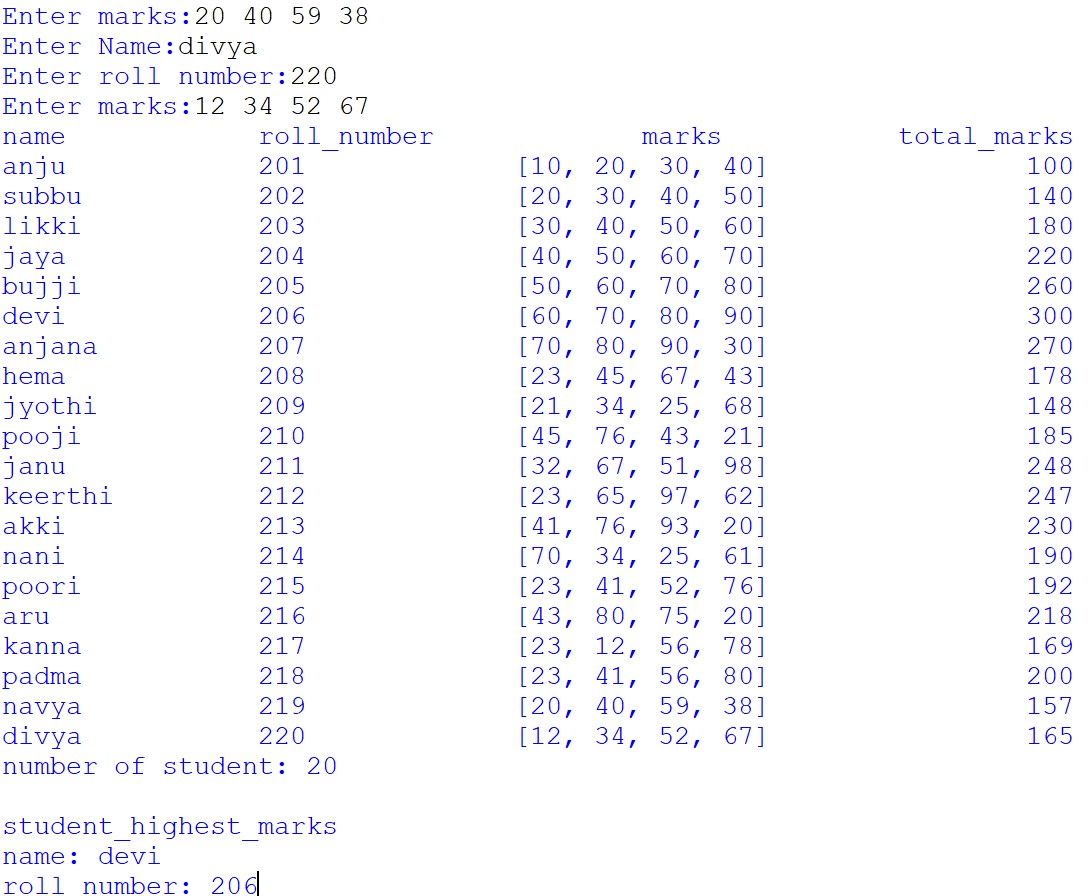


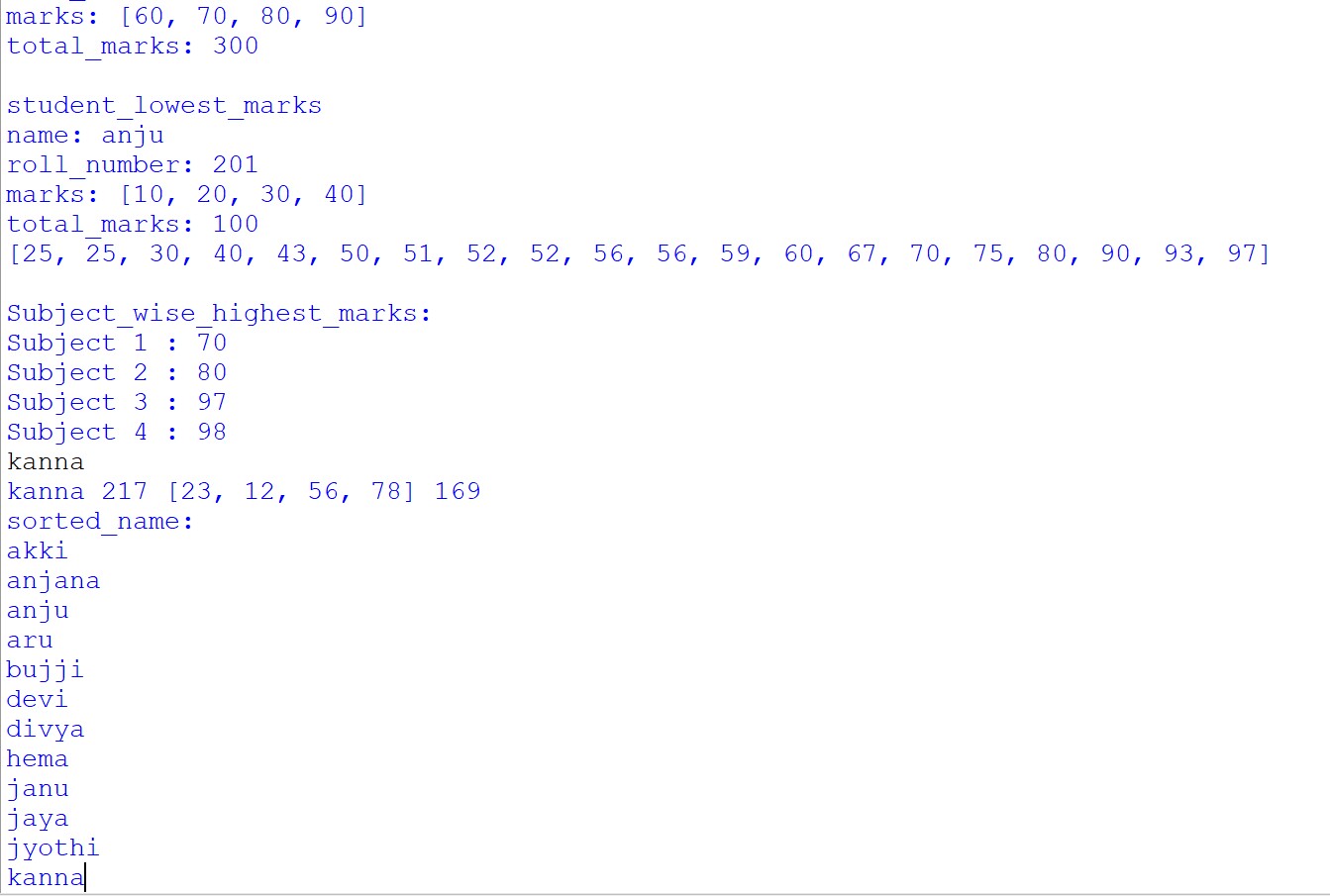
Output:-

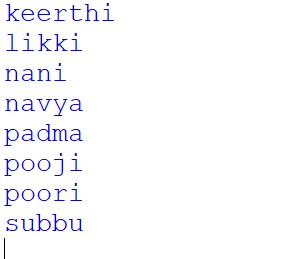












Explanation:-

 **Importing Libraries**: It imports the pandas library as pd.

 **Initializing Lists**: Two lists are initialized, s for storing student data dictionaries and l for storing marks of third subjects across students.

 **add\_student Function**: This function takes three parameters - name, roll number, and marks. It creates a dictionary representing a student's data and appends it to the s list. It also calculates the total marks for each student.

 **Input Section**: It takes input for the number of students (n) and then iterates n times to take input for each student's name, roll number, and marks for different subjects.

 **Printing Student Data**: It prints the student data in a tabular format, including name, roll number, marks, and total marks.

 **Calculating Statistics**: It calculates and prints the number of students, student with the highest marks, student with the lowest marks, marks of the third subject sorted in ascending order, and subject-wise highest marks.

 **College Search**: It takes input for a college name and prints the details of students belonging to that college.

 **Sorting by Name**: It sorts the student data by name and prints the sorted names.

Conclusion:-

The provided Python code manages student data by storing information such as name, roll number, and marks for multiple students. It calculates various statistics like total marks, highest and lowest marks, subject-wise highest marks, and sorts student data by name. Additionally, it allows users to search for students based on a specified college name. However, the code could be further improved with error handling, such as input validation, and more modularization for better readability and maintainability.

Top of Form

Bottom of Form

THANK YOU