**ALGOSCOPE: Intelligent Algorithm Analyzer – PBL Prep Sheet**

**1. Project Summary**

ALGOSCOPE is an Intelligent Algorithm Analyzer aimed at evaluating the performance of various algorithms under different computational conditions. The project enables comparative visualization and analysis, making it suitable for academic and research-based insights.

**2. Objective**

To build an intelligent tool that visualizes and analyzes algorithm performance based on key metrics like time complexity, space complexity, input size, and execution time.

**3. Technology Stack**

* **Frontend:** HTML, CSS, JavaScript
* **Backend:** Python (Flask)
* **Visualization:** Matplotlib, Plotly
* **Data Handling:** Pandas
* **IDE:** Visual Studio Code
* **Deployment:** Local Server / Streamlit (Optional)

**4. Workflow Architecture**

The workflow begins with algorithm selection by the user, followed by input configuration. The backend executes the algorithm, records performance metrics (like time and space), and sends results to the frontend for visualization.

**5. Team Roles & Contributions**

* **Shorya Tripathi:** Frontend development and visualization logic
* **Krishna Bhatt:** Backend algorithm implementation
* **Aditya Kumar Sharma:** Performance metrics integration and testing
* **Devansh Bisht:** Documentation, presentation, and final integration

**6. Algorithms Covered & Implementation Plan**

We analyze the following algorithms:

* Bubble Sort, Merge Sort, Quick Sort
* Linear Search, Binary Search
* Dijkstra's Algorithm (for path finding)

Each algorithm is implemented in Python and instrumented to track execution time and memory usage, with dynamic visual output to help understand behavior.

**7. Expected Results & Evaluation Metrics**

* Execution time graphs for various inputs
* Space usage visualization
* Comparative analysis: Best, Worst, and Average cases
* An interactive dashboard for users to compare results across algorithms

**8. Sample Viva Questions & Answers**

**Q1: What is the objective of ALGOSCOPE?**  
A1: To analyze and visualize the performance of algorithms in an interactive and educational manner.

**Q2: Why did you choose Python for the backend?**  
A2: Python supports fast prototyping, is rich in data and visualization libraries, and is beginner-friendly.

**Q3: How do you measure algorithm performance?**  
A3: We use Python’s time module for execution time and tools like sys.getsizeof or memory\_profiler for space analysis.

**Q4: What role does visualization play in this project?**  
A4: It makes comparisons intuitive and helps learners grasp algorithm efficiency without needing to interpret raw data.

**Q5: How did your team distribute the work?**  
A5: Each member was assigned a domain—frontend, backend, metrics tracking, documentation—ensuring full project coverage and accountability.

**9. Presentation & Report Tips**

* Keep slides clean, aligned, and justified
* Use bullet points for clarity
* Use diagrams for workflows and architecture
* Practice explaining your part clearly
* Check for grammar and consistency
* Include **team photos in college uniform** in the final report