

## DATA STRUCTURES SEMESTER PROJECT RUBRIC

GROUP MEMBERS:

EVALUATOR:

Criteria	Weightage(Marks)	Description
<b>Code Quality &amp; Standards</b>	10	<ul style="list-style-type: none"><li>- Follows coding standards and best practices.</li><li>- Well-documented code with appropriate comments and clear variable names.</li><li>- Efficient use of data structures and algorithms.</li><li>- Avoid committing incomplete or non-functional code</li><li>- Only the standard C++ library is allowed. External libraries, such as STL-based or non-standard libraries, are prohibited.</li></ul>
<b>Functional Implementation</b>	50	<ul style="list-style-type: none"><li>- Successful implementation of core system modules (traffic network, vehicle routing, signal management, etc.).</li><li>- Accuracy of algorithms like Dijkstra's, BFS/DFS, priority queues and all data structures</li><li>- Handling of emergency scenarios.</li></ul>
<b>Simulation Features</b>	15	<ul style="list-style-type: none"><li>- Completeness and interactivity of the simulation dashboard.</li><li>- Proper visualization of traffic flow, congestion levels, and signal statuses.</li><li>- Effective use of sample inputs.</li></ul>
<b>Performance Metrics</b>	10	<ul style="list-style-type: none"><li>- Demonstrates key metrics like average vehicle travel time, signal adjustment efficiency, and resolved congestion events.</li><li>- Accurate and clear metric tracking.</li></ul>
<b>Repository Management</b>	10	<ul style="list-style-type: none"><li>- Regular and meaningful GitHub commits.</li><li>- Well-structured repository with a detailed README</li><li>- Comprehensive project documentation.</li></ul>
<b>Error Handling</b>	5	<ul style="list-style-type: none"><li>- Graceful handling of potential errors like invalid inputs, file I/O issues, and system disruptions.</li><li>- Code robustness under edge cases and failure scenarios.</li></ul>

<b>Creativity &amp; Bonus Tasks</b>	10	- Implementation of the bonus task (e.g dynamic routing with changing traffic conditions). - Innovative additions beyond the stated requirements.
-------------------------------------	----	--

- Teams must showcase their repository structure, commit history, and functionality during the presentation/demo.
- Ensure all features are demonstrable. Failure to do so will result in deduction of marks.
- Strict adherence to the honor code; plagiarized submissions will be penalized as outlined.
- Code with syntax errors will receive zero marks for the affected sections.
- Proper commenting and indentation are mandatory for readability and organization.