Below are the basic questions that you must address through the code you are designing. My hope was that each group would generate their own set of minimum (min) requirements from the project description, but it seems to me that some of you could benefit from more definition. I have also added suggestions on how you could go above these min requirements.

For your final report, you will have a section that describes the minimum requirements your group used.

You can simply copy and paste these, if that is what you ended up doing.

1. Mesh Import, Parsing, and Validation

- Implement a module to import, parse and export surface or volume meshes in common formats (e.g., STL, OBJ, MSH). You may used the provided formats, or come-up with one that is your own. Please document carefully.
    - Beyond min: You can go beyond the min import requirement by performing an operation such as: Validate the mesh structure, ensuring closed surfaces, consistent face orientations, etc. absence of self-intersections.
    - Beyond min: Provide error handling and reporting for corrupted or incomplete mesh files.

---

2. Efficient Mesh Storage and Data Structures

- Develop (ideally efficient ones) data structures to store unstructured meshes with minimal memory overhead.
    - Beyond min: Ensure compatibility with large-scale meshes, capable of handling millions of elements while maintaining quick access to vertices, edges, and faces.
- Store mesh metadata, such as boundary conditions, material properties, and element tags.

---

3. Mesh Manipulation and Transformation API

- Implement geometric transformations, such as translation, rotation, and scaling, while maintaining mesh integrity.
- Boolean operations (union, difference, intersection) for multi-object integration.

- Create a visualization tool, or use one that exists for on-demand inspection.
    - Beyond min: Be able to perform vertex smoothing and relocation.
    - Beyond min: Rapid refinement (splitting) and coarsening (merging) of elements.

---

4. BEYOND MIN: Computational Efficiency

- Ensure algorithms operate efficiently for high-resolution meshes, avoiding excessive memory consumption and processing delays.
- Implement lazy evaluation strategies where possible to avoid redundant computations.

---

5. Output, Visualization, and Documentation

- Provide functionality to export the processed mesh in industry-standard formats, or a format of your own choosing that is well-documented, for downstream simulation tools (e.g., CFD, FEA).
- Beyond min: Integrate with visualization tools such as Paraview or MeshLab for quality inspection.