📝 Assessment Task Overview

You will be provided with five real-world problem statements designed to evaluate your proficiency in computer vision, 3D modeling, UI-to-code automation, and practical ML deployment.
Please feel free to use any stack or tools you're comfortable with. Casual, exploratory development is accepted.

🚩 Problem 1: Tower Reconstruction from 2D Images

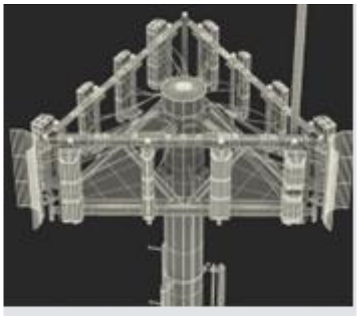
Objective:
Develop a system to reconstruct a cellular tower in 3D using 2D input images (e.g., labeled diagrams or snapshots), along with tabular physical specifications.

Key Tasks:

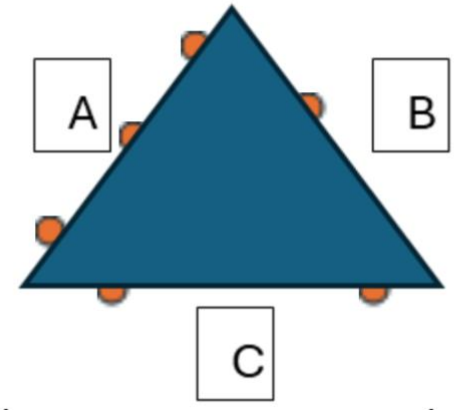
- Parse and apply configuration data (e.g., TOWER_CONFIG) to build structural elements.
- Use a 3D engine or graphics framework to render and manipulate components.
- Support scale, position, and rotation for each component.
- Export formats: .STP, .GLB, or .OBJ.

Example Input:

Tower Reconstruction



Assume that we have 2D images of such towers. For example below one shows there are 3 antennas on side A, only one on B and on C.



Based on this and other physical dimensions available, can you reconstruct the Tower in 3D format (STP/GLB format)
Problem Statement: to develop a **3D reconstruction and simulation system** for **radio communication towers.**
The system should allow dynamic modification of tower components—such as the **base**, **poles**, and **antenna arrays**—and simulate their placement in a 3D environment.

Expected Outcome:
A structured, modifiable 3D model file representing the cellular tower's geometry and antenna placements.

Testing:
You may use any publicly available image (e.g., hand-drawn sketches or photos of towers) and mock configuration specs to develop and demonstrate your solution. Evaluation will be based on how accurately your 3D output reflects image features and configuration details.

🚗 Problem 2: Driving Footage to 3D Road Scene Simulation

Objective:
Build a pipeline that processes publicly available traffic footage and reconstructs a simplified 3D road scene.

Key Tasks:

- Work with fixed-angle dashcam or traffic camera videos.
- Detect and track general vehicle categories.
- Simulate scene with inferred road lanes and vehicle motion.

Expected Outcome:
A working 3D simulation in Unity or WebGL (Three.js) showing accurate vehicle positioning and movement.

Testing:
Use any publicly available traffic or dashcam video to demonstrate your pipeline. We will evaluate how well your simulation captures road geometry and vehicle behavior relative to the source footage.

🧑💻 Submission & Evaluation

- **Code:** Share complete, runnable scripts or projects(Zip folder)
- **Assets:** Use and include any publicly available or generated inputs (images, video, Figma, etc.).
- **Instructions:** Include a clear README.md for setup and execution.
- **Evaluation Criteria:** Accuracy, clarity, modularity, and practical usability of the output.