# Simulated Robotic Workstation for Tool Manipulation

This step-by-step guide walks you through building a simulation project that showcases tool detection, motion planning, and infrastructure capabilities—perfectly aligned with Altitude AI's mission.

## Stage 1: Project Setup & Folder Structure

Goal: Initialize the environment and project structure.

* Create the folder structure:
* mkdir robotic\_sim && cd robotic\_sim
* mkdir perception motion\_planning infrastructure tools
* touch main.py README.md
* Set up a Python virtual environment:
* python3 -m venv venv
* source venv/bin/activate # or venv\Scripts\activate on Windows
* Install dependencies:
* pip install opencv-python matplotlib pygame flask ikpy

## Stage 2: Perception Module

Goal: Detect tools on a 2D workspace image.

* Collect or create simple tool images (PNG or JPG) and save them in /tools.
* Write detect\_tools.py in /perception/:
* - Load the workspace image.
* - Use OpenCV to convert to grayscale.
* - Apply contour detection or template matching.
* - Return tool names and coordinates.
* Test tool detection by calling the function from main.py and printing results.

## Stage 3: Motion Planning Module

Goal: Simulate a 2D robotic arm reaching for the detected tool.

* In plan\_motion.py (in /motion\_planning/):
* - Simulate a 2-link arm with forward/inverse kinematics.
* - Define tool positions from the perception step.
* - Calculate joint angles to reach the tool’s location.
* Visualize the motion using matplotlib or pygame.
* Integrate with perception by using detected coordinates as targets.

## Stage 4: Infrastructure & Logging

Goal: Add observability and user interaction features.

* Create dashboard.py in /infrastructure/:
* - Use Flask to serve a simple dashboard.
* - Display logs (tool detected, coordinates, motion actions).
* Create logger.py in /infrastructure/:
* - Write a function to log events to a .log file or SQLite database.
* Create a simple CLI in main.py to trigger tool detection and movement.

## Stage 5: Polish & Documentation

Goal: Finalize and document your project for presentation.

* Document each function with docstrings and inline comments.
* Update README.md with an overview, usage instructions, and screenshots.
* Optionally create a short demo video or .gif using screen capture.
* Create requirements.txt: pip freeze > requirements.txt

## Suggested Timeline

|  |  |
| --- | --- |
| Day(s) | Task |
| 1 | Setup + Folder + Tool Images |
| 2–3 | Perception Module |
| 4–5 | Motion Planning Module |
| 6 | Infrastructure + CLI |
| 7 | Documentation + Polish |