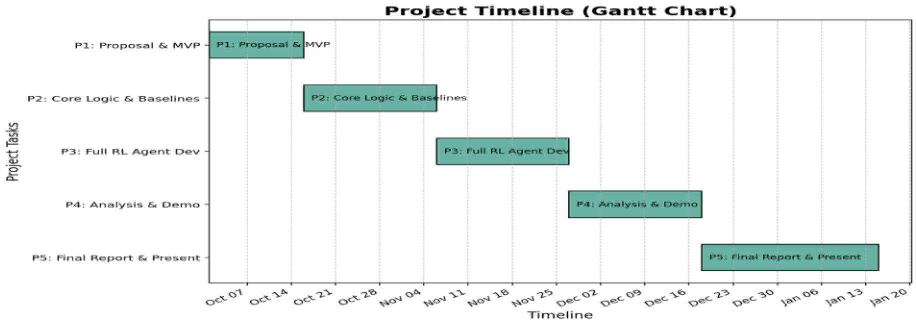
MDM2 – Case Study: Intelligent Systems in Production One-Page Proposal

Team	Group 5
Members	Sanaboyina Satya Narasimha, Rifshu Hussain Shaik, Gnanasudha Patur, Alwin Shaji, Ankith Ramesh Babu.
Project Title	Optimizing Job Shop Scheduling with Deep Reinforcement Learning
GitHub Repository URL	https://github.com/rifshu/JOB_SHOP-CASE-STUDY-Main
Contact Email	shaik.rifshu-hussain@stud.th-deg.de
Industrial Application	Manufacturing, Production Planning, and Logistics
Keywords	Job Shop Scheduling, Deep Reinforcement Learning, Optimization, Simulation, Smart Factory
Submission Date	2025-10-16
Gantt Chart	
1) Problem Statement & Measurable Outcomes	The Job Shop Scheduling Problem (JSSP) is a concrete production logistics challenge where traditional scheduling methods are often inefficient; this project will develop an intelligent agent using Reinforcement Learning (RL) to find near-optimal policies in a simulated environment. Our success will be measured by three key KPIs: reduced makespan, increased machine utilization, and decreased job tardiness.
2) Motivation & Industrial Relevance	Production planners and factory managers are the primary beneficiaries, as an effective Deep RL-based scheduler creates significant value by improving throughput, reducing costs, and increasing adaptability to disruptions. This work is critically important now because the widespread adoption of Industry 4.0 principles in manufacturing demands intelligent, dynamic scheduling systems that can outperform traditional, rigid methods.
3) Related Work Snapshot	Prior work by Belmamoune et al. (2022) successfully used a Q-learning agent to select from dispatching rules but noted this approach was limited by a small action space and underperformed compared to more complex Deep RL models. Our project directly addresses this gap by training an agent to select individual jobs for a more granular policy, which we will benchmark against both simple heuristics and advanced Deep RL techniques to quantify the benefits.
4) Method & Feasibility	Our method involves developing a discrete-event job shop simulation in Python with `simpy` as a training ground for our RL agent. We will implement the agent using `Stable-Baselines3`, train it on benchmark JSSP datasets, and analyze its performance against baseline heuristics (FIFO, SPT). The expected artifacts—simulation code, a trained model, and a final comparative report—confirm that this approach is highly feasible within the project's scope.
5) Milestones & Timeline	**P1:** Proposal, MVP Simulation & Basic Agent (2025-10-16) **P2:** Core Simulation Logic & Baseline Implementation (2025-11-06) **P3:** Full RL Agent Implementation & Training (2025-11-27) **P4:** Preliminary Results Analysis (2025-12-18) **P5:** Final Report & Presentation (2026-01-15)
6) Risks & Ethics (1–2 sentences)	The primary feasibility constraint is the "simulation-to-reality gap," where our simplified model may not capture real-world complexities. Ethically, our focus is on proper citation for all datasets and analyzing the trained agent for scheduling biases; data privacy is not a concern as we are using public, anonymized benchmarks.