CEN 352 Term Project Guide:

Design and Implementation of an Intelligent Agent

I. Project Overview and Objectives

The goal of the Term Project is to apply the theoretical knowledge and practical techniques learned throughout the course—covering search, logic, planning, and machine learning paradigms—to design and implement a non-trivial **intelligent agent**.

Successful completion of this project should demonstrate the ability to:

1. Propose and develop an AI system using at least **two different AI techniques** taught in the course (e.g., a neural-network-based classifier plus a search or reinforcement component).
2. The system can be an original idea or an extension of an existing open-source project, but students must **significantly enhance** it (new features, improved performance, or new domain).
3. Include an ethical reflection: identify one potential societal impact of your project.
4. **Deliverables:** Working code, 5–7 minute presentation, and a 5~8-page report explaining techniques used, evaluation results, and ethical considerations.

Examples:
– A chatbot with both intent classification (NN) and planning (search).
– A reinforcement-learning agent for a custom-designed game.
– Extending an open-source AI project to a new application domain.

II. Project Requirements and Constraints

Component	Requirement Details
Group Structure	Students must work in **groups of 2 or alone**.
Project Topic	The project must focus on a core AI problem covered in the course. It should implement and demonstrate the functionality of an **intelligent agent** designed using principles such as **rationality**.
Technical Scope	The project must incorporate and leverage at least one major paradigm studied, such as: **Search** (A*, BFS, DFS), **Logical Reasoning** (Rule-Based Systems/Forward Chaining), **Planning** (PDDL, HTN), **Statistical Learning** (SVM, Decision Tree, RF, XGBoost, etc.), or **Reinforcement Learning** (Q-learning).
Implementation Language	All code must be implemented in **Python**. The use of libraries like **scikit-learn** (used for SVM/Decision Trees) or deep learning frameworks like **PyTorch/Keras/TensorFlow** is encouraged.
Evaluation	The agent's performance must be quantitatively evaluated. This must include a definition of the **Performance Measure** (P in the PEAS framework) and use appropriate metrics such as **Accuracy** (if classification) or the **F1 score**.

III. Deliverables and Submission

The project is due during **week 13 (Jan 11, 2026)**. All materials must be submitted via **GitHub**, following the practice established in the assignments.

A. GitHub Repository Submission

The group must create a private GitHub repository containing the following files:

1. **Source Code:** All Python files required to run the agent. All the code files should be placed under /src/ folder.

2. `README.md`: A comprehensive file detailing:
    - Group members and roles.
    - Instructions for running the project (including dependencies and installation).
    - A brief overview of the implemented AI approach.
3. **Report (PDF):** The final written project guide. Max. 8 pages.

B. Project Presentations

Presentations are mandatory during **Week 14 and 15**. The presentation should cover:

1. **Problem Formulation:** Define the task environment using the **PEAS framework** (Performance Measure, Environment, Actuators, Sensors).
2. **AI Methodology:** Explain the chosen algorithms (e.g., why SVM over a Decision Tree, or why A* over BFS).
3. **Implementation:** Demonstrate the agent's functionality.
4. **Results and Analysis:** Discuss the performance using the chosen metrics (e.g., Accuracy, F1 score).
5. **Future work and Improvements:** Discuss what can be done more to improve the system/solution.

IV. Project Timeline and Grading

Activity	Week	Weight	Notes
Project Proposal	**10 - (Dec 17)**	10%	Briefly define the problem and the proposed AI technique. **Max. 2 pages**
Project Presentations	**Week 14 - 15**	20%	Mandatory public presentation of results and implementation.
Final Project (GitHub)	**13 - (Jan 11)**	**30%**	Comprehensive submission of code and report.

V. Conceptual Evaluation Focus

When preparing the project, ensure the report thoroughly addresses the following aspects of your agent design, aligning with course concepts:

1. **Agent Design:** Justify the classification of your agent (e.g., simple reflex, model-based, learning agent) and analyze its task environment (e.g., fully/partially observable, static/dynamic, discrete/continuous).
2. **Model Selection:** If using statistical learning (Week 6-9), discuss why the chosen model (e.g., SVM, DT, RF, XGBoost) is appropriate for the problem and how its complexity or function compares to alternative models.
3. **Correctness and Efficiency:** If using search or planning, discuss the algorithm's properties concerning **completeness, optimality, time complexity, and space complexity**.

Rubrics

Integrative AI System (100 pts)

Criterion	Points	Description
Proposal & Originality	10	Project idea clear; either original or demonstrably enhanced existing project.
Integration of Multiple AI Techniques	20	At least two different course techniques implemented; clear interaction between them.
Technical Implementation Quality	20	Code runs reliably; non-trivial features implemented; includes training/evaluation.
Evaluation & Results	15	Quantitative or qualitative evaluation; performance metrics or user testing.
Ethical & Societal Reflection	10	Identifies one potential societal/ethical issue; discussion shows awareness.
Presentation Quality (5–7 min)	10	Clear, structured presentation with demo.
Written Report (≤8 pages)	7	Concise, professional; methods, evaluation, and reflection all summarized.
Documentation & Reproducibility	8	Code well-organized, instructions included, can be rerun by others.

Originality Enforcement:
– Students must list any external codebases used.
– A project that is merely a trivial modification of an open-source example can earn at most 50% of points in the Integration + Implementation categories.