

# Client Precondition Report Template

## 1. Company Information

- **Company Name:** CogniCo AI Consulting
- **Primary Contact Person:** Saif Ahmed
- **Email Address:** saif.ahmed@cognico.ca
- **Website:** www.cognico.ca
- **Location / Time Zone:** EST

## 2. Team Structure & Required Skills

Describe your expectations for how the student team will operate.

### a. Team Composition

**Team Composition** We envision a team of 8-10 students operating as a functional agile software team.

- **Agile Technical Project Lead (1):** Manages the technical features/bugs sprint dashboard, timeline, and client communication.
- **AI & Backend Squad (3-4 students):** Responsible for the Recommendation Engine, Python API, and Data Pipeline (ETL).
- **Frontend & Integration Squad (2-3 students):** Responsible for the Student Widget (Learnworlds injection) and the **New Admin Analytics Dashboard**.
- **DevOps & QA Squad (2 students):** Responsible for CI/CD pipelines, automated testing frameworks, and cloud infrastructure management to ensure high code quality and smooth PR reviews.
- Rotations: We encourage students to rotate roles once during the semester (e.g., at the halfway point) to gain exposure to both data engineering and integration logic.

### b. Required Baseline Skills

List the *foundational* skills students must already have to participate successfully.

CS 472 students are **senior CS undergraduates** with general programming experience. They are **not industry experts**, but they are capable of learning new tools and technologies used in your project.

Baseline skills may include:

- General programming experience
- Basic understanding of software development practices
- Familiarity with Git/GitHub

- Willingness and ability to learn new frameworks or tools

Please avoid listing advanced or specialized skills as mandatory prerequisites.

- **General Programming:** Proficiency in Python (preferred for AI/Backend) or JavaScript/React (Frontend), AI-assisted coding (using Gemini/Claude/Codex /Cursor).
- **API Interactions:** Understanding of RESTful APIs and JSON data handling.
- **Version Control:** Proficiency with Git/GitHub workflows (branching, merging, pull requests).
- **Data Handling:** Basic SQL knowledge and familiarity with data structures.
- **Testing Mindset:** Understanding of Unit Testing and Integration Testing concepts.
- **Soft Skills:** Clear communication and ability to document code thoroughly.

### **c. Skills to Be Developed During the Project**

List skills the students will gain while working with you (frameworks, tools, domain knowledge, engineering practices, etc.).

- **Applied AI/ML:** Implementing recommendation systems (collaborative filtering or content-based) and utilizing LLM APIs (e.g., Google Gemini).
- **LMS Architecture:** Understanding the backend structure of platforms like Learnworlds.
- **Enterprise Integration:** Practical experience with Zapier webhooks.
- **Cloud Deployment:** Deploying a microservice to a cloud provider (AWS) to serve recommendations.
- **Data Visualization:** Creating actionable insights for admins using charting libraries (e.g., D3.js or Recharts).

## **3. Project Title & Acronym**

- **Project Title: Personalized LMS**
- **Acronym (for GitHub repository): FoodSystemsTech**

## **4. Project Overview & Problem Statement**

Provide a clear description including:

- The problem your project addresses
- Background/context
- Target users
- Goals and expected impact

Our current LMS (Learnworlds) offers a static "one-size-fits-all" experience. Users often struggle to find the next best step in their education, leading to lower engagement and completion rates.

CogniCo LMS has data silos that are not currently communicating to benefit the learner. Learners get lost in content libraries, and administrators lack granular data on how AI impacts learning outcomes.

#### **Target Users:**

**Learners:** Professional learners seeking upskilling who need personalized guidance.

**Admins:** Staff who need to track engagement and AI effectiveness.

**Goal:** Build a dual-interface system that includes an AI-driven Learning Assistant for students to personalize their path, and an Analytics Dashboard for admins to monitor the "Health" of the education program. The goal is to create a dynamic "Intelligence Layer" that sits on top of the LMS, analyzing user behavior to suggest specific courses, resources, or learning pathways automatically.

## **5. Solution Overview & Core System Components**

Describe the main system components, modules, or functionalities required.

The solution requires three main components:

1. **Data Ingestion Engine:** A service that pulls user progress data from Learnworlds.
2. **Recommendation Microservice:** The "Brain" (Python-based). It processes the data to generate personalized outputs (e.g., "Because you finished Module A, you should try Resource B").
3. **Frontend/Integration Layer:** The mechanism to display these recommendations back to the user, likely via a custom widget injected into Learnworlds or a dashboard accessible via a student portal.

## **6. Technical Considerations**

Specify:

- Required or preferred tech stack
- APIs, SDKs, datasets, or tools
- Constraints or limitations
- Integration requirements
- Quality/testing/CI considerations
- Security/privacy needs

**Preferred Stack:** Python (FastAPI or Flask) for the backend logic; JavaScript for frontend injection.

**APIs & Tools:** Learnworlds API, Zapier (for workflow automation), and potentially Google Gemini API for generating natural language learning summaries.

**Constraints:** Must work within the limitations of the Learnworlds code-injection capabilities.

**Security:** Student data must be anonymized or handled in compliance with basic privacy standards (GDPR/PIPEDA principles) during development.

## 7. Innovation & Competitive Advantage

Describe what makes your project innovative or distinct.

Most LMS platforms lock personalization behind expensive enterprise tiers. Our project is innovative because it creates a modular, external AI brain that can plug into a standard LMS, democratizing access to adaptive learning technologies usually reserved for massive universities.

## 8. Proposed Implementation Timeline (12 Weeks)

Provide your **own** realistic 12-week timeline and expected deliverables.

- **Weeks 1-2 (Discovery):** API exploration (Learnworlds), requirement gathering, and setting up the Git repo.
- **Weeks 3-4 (Data Pipeline):** Building scripts to fetch and clean user activity data.
- **Weeks 5-7 (AI Core):** Developing the algorithm/logic for recommendations (e.g., "If user views X, recommend Y").
- **Weeks 8-9 (Integration):** Connecting the AI output to the user interface (via Zapier or direct API).
- **Weeks 10-11 (Testing & Refinement):** User acceptance testing, bug fixing, and latency optimization.
- **Week 12 (Handover):** Final presentation, documentation delivery, and code transfer.

## 9. Deployment & Support Expectations

Specify any requirements related to:

- Hosting or infrastructure
- Access credentials or resources you will provide
- Documentation or training expectations

**Hosting:** Students can develop locally or use free-tier cloud services (e.g., AWS Free Tier). CogniCo can provide AWS credits if necessary.

**Credentials:** CogniCo will provide Sandbox access to a test environment for Learnworlds.

**Documentation:** We expect a robust README.md and a "Maintenance Manual" for future developers.

## 10. Confidentiality & Intellectual Property

Provide your IP, NDA, portfolio usage, and disclosure policies.

- **IP Ownership:** All code and IP developed belong to CogniCo AI Consulting.
- **Portfolio Use:** Students are granted a perpetual, non-exclusive license to showcase the project in their portfolios and GitHub (subject to removing specific proprietary API keys or sensitive user data).
- **NDA:** Students may be asked to sign a standard non-disclosure agreement regarding specific client data.

## 11. Proposed Success Metrics

List the metrics (quantitative and qualitative) you will use to evaluate project success.

- **Functional:** The system successfully pulls data, generates a recommendation, and displays it without crashing.
- **Performance:** Recommendations are generated in under 2 seconds.
- **Quality:** The recommendations make logical sense based on the test user profiles provided.

## 12. Mentorship & Communication Plan

Specify:

- Expected meeting cadence (Professor prefers weekly meetings)
- Communication platforms
- Response time expectations
- Who will mentor or support the students
- **Cadence:** Weekly 30-minute sync meetings (via Google Meet/Slack).
- **Platform:** A dedicated Slack channel for asynchronous Q&A.
- **Response Time:** Mentors (Saif Ahmed and technical lead) will aim to respond to queries within 24 hours on weekdays.
- **Mentorship Style:** We will treat the students as junior consultants and providing guidance on architecture and business logic but expecting them to own the code implementation work.

## 13. Additional Notes (Optional)

N/A