Project Charter: Agentic Lab

หัวข้อ (Key)	รายละเอียด (Value)
Project Name	โครงการจัดตั้งห้องปฏิบัติการเสมือนสำหรับปัญญาประดิษฐ์เชิงตัวแทน (Agentic Living Lab Platform)
Codename	Agentic Lab
Request By	สาขาวิทยาศาสตร์และนวัตกรรมข้อมูล วิทยาลัยสหวิทยาการ มหาวิทยาลัยธรรมศาสตร์
Due Date	4 เดือน (นับจากวันเริ่มต้นโครงการ)

Goals (เป้าหมายหลัก)

ในยุค Digital Transformation และการมาถึงของ "Agentic AI" (AI ที่มีความสามารถในการคิดและกระทำการได้ด้วยตนเอง) ภาคการศึกษามีความจำเป็นเร่งด่วนในการปรับตัว โครงการนี้จึงมีเป้าหมายหลักเพื่อสร้าง **"Living Lab"** หรือห้องปฏิบัติการมีชีวิต ที่เป็นแพลตฟอร์ม จริง (Real-world Platform) บนโครงสร้างพื้นฐานแบบ Hybrid Cloud

เป้าหมายสำคัญคือการปิดช่องว่างทางทักษะ (Skills Gap) ให้กับนักศึกษา โดยเปลี่ยนรูปแบบการเรียนรู้จากทฤษฎีสู่การปฏิบัติจริง ผ่านการพัฒนาและบริหารจัดการ AI Agents จำนวน 9 ตัว เพื่อบริหารจัดการห้องแล็บของตนเอง ตั้งแต่กระบวนการรับสมัคร จับคู่โปรเจกต์ ไปจนถึงการส่งมอบงานให้ภาคเอกชน ซึ่งจะเป็นต้นแบบการเรียนรู้สมัยใหม่ที่เชื่อมโยงกับภาคอุตสาหกรรมอย่างแท้จริง

Objectives (วัตถุประสงค์)

- **Build (สร้างนวัตกรรม):** เพื่อจัดตั้งโครงสร้างพื้นฐานกลาง (Central Infrastructure) ที่มีความพร้อมให้นักศึกษาสามารถพัฒนา ทดสอบ และ Deploy ระบบ Agentic AI ได้จริง
- **Integrate (บูรณาการระบบ):** เพื่อเชื่อมต่อทรัพยากรระหว่างระบบภายใน (On-premise: Kubernetes/Ollama) และระบบคลาวด์ (Cloud: GCP/Gemini) ให้ทำงานร่วมกันแบบ Hybrid ได้อย่างไร้รอยต่อ
- **Prepare (เตรียมความพร้อม):** เพื่อบ่มเพาะทักษะการบริหารจัดการโครงการ (Project Management) และวิศวกรรมข้อมูล (MLOps) ผ่านการลงมือทำจริงในสภาพแวดล้อมเสมือนการทำงานจริง

Scope (ขอบเขตการดำเนินงาน)

- **In-Scope (สิ่งที่ดำเนินการ):**
  - การพัฒนาระบบ Living Lab Platform (Web Application)
  - การติดตั้งและตั้งค่าระบบ Hybrid Infrastructure (Kubernetes & Google Cloud)
  - การจัดอบรมเชิงปฏิบัติการ (Workshops) จำนวน 9 หัวข้อ ตามโมเดล 9 Agents
  - การดำเนินโครงการร่วมกับพันธมิตรภายนอก (Real-world Projects) อย่างน้อย 5 โครงการ
- **Out-of-Scope (สิ่งที่ไม่ดำเนินการ):**
  - การจัดซื้อ Hardware Server ใหม่ (ใช้ทรัพยากรเดิมที่มีอยู่หรือเช่า Cloud Services)
  - การดูแลรักษาระบบ (Maintenance) ระยะยาวเกินกว่า 6 เดือนหลัง

สิ้นสุดโครงการ

Deliverables (สิ่งที่ต้องส่งมอบ)

- **1. ระบบโครงสร้างพื้นฐานและแพลตฟอร์ม (Infrastructure & Platform)**
  - 1.1 ระบบ Hybrid Cloud Infrastructure
    - 1.1.1 การติดตั้ง Kubernetes Cluster บน Server On-premise
    - 1.1.2 การตั้งค่า Google Cloud Platform (Cloud Run, Cloud SQL, Vertex AI)
  - 1.2 ระบบ Living Lab Web Platform
    - 1.2.1 ส่วนติดต่อผู้ใช้งาน (Frontend) พัฒนาด้วย React รองรับ Responsive Design
    - 1.2.2 ระบบหลังบ้าน (Backend) พัฒนาด้วย Django รองรับ API Gateway
- **2. ระบบตัวแทนปัญญาประดิษฐ์ (AI Agents Suite)**
  - 2.1 กลุ่ม Data Intelligence (ระยะที่ 1)
    - 2.1.1 **Agent 1 (OA):** ระบบรับสมัครและแปลงข้อมูลเรซูเม่ (OCR & Data Extraction)
    - 2.1.2 **Agent 2 (CA):** ระบบจัดหมวดหมู่และแนะนำหลักสูตร (Recommendation System)
    - 2.1.3 **Agent 3 (MA):** ระบบจับคู่นักศึกษากับโจทย์งาน (Matching Algorithm)
  - 2.2 กลุ่ม Action & Orchestration (ระยะที่ 2)
    - 2.2.1 **Agent 4 (DCSA):** ระบบเชื่อมต่อ API ภายนอกและ

ชุมชน (External API Integrator)
        - 2.2.2 **Agent 5 (EOA):** ระบบจัดการอีเวนต์และตารางเวลา (Event Scheduler)
        - 2.2.3 **Agent 6 (PM):** ระบบผู้ช่วยบริหารโครงการ (Project Tracking Assistant)
    - 2.3 กลุ่ม Deployment & Impact (ระยะที่ 3)
        - 2.3.1 **Agent 7 (SWA):** ระบบดูแลสุขภาวะและ Support Desk (Chatbot Support)
        - 2.3.2 **Agent 8 (CRA):** ระบบ Deploy ผลงานอัตโนมัติ (CI/CD Pipeline Automation)
        - 2.3.3 **Agent 9 (AEA):** ระบบรายงานผลผู้บริหาร (Executive Dashboard)
- **3. กิจกรรมการเรียนรู้และผลลัพธ์ (Learning & Outcomes)**
    - 3.1 การอบรมเชิงปฏิบัติการ (Hybrid Workshops)
        - 3.1.1 สื่อการสอนและวิดีโอบันทึกย้อนหลังจำนวน 9 หัวข้อ
    - 3.2 ผลงานโครงการจริง (Real-world Projects)
        - 3.2.1 ต้นแบบนวัตกรรมจากโจทย์พันธมิตร จำนวน 5 โครงการ

Plan (Work Breakdown Structure)

1. **Phase 1: Foundation & Data Intelligence (Month 1)**
    - 1.1 Project Kick-off & Requirement Analysis
    - 1.2 UI/UX Design Phase (Wireframe & Prototype)
    - 1.3 Infrastructure Setup (Hybrid Cloud & Kubernetes)
    - 1.4 Development & Workshop: Agent 1 (Onboarding Agent)
    - 1.5 Development & Workshop: Agent 2 (Curriculum Agent)

    - 1.6 Development & Workshop: Agent 3 (Matching Agent)
2. **Phase 2: Action & Orchestration (Month 2)**
    - 2.1 Platform Core Feature Development
    - 2.2 Development & Workshop: Agent 4 (Community Agent)
    - 2.3 Development & Workshop: Agent 5 (Event Orchestration Agent)
    - 2.4 Development & Workshop: Agent 6 (Project Management Agent)
3. **Phase 3: Deployment & Impact (Month 3)**
    - 3.1 System Integration & User Acceptance Test (UAT) Phase 1
    - 3.2 Development & Workshop: Agent 7 (Support & Well-being Agent)
    - 3.3 Development & Workshop: Agent 8 (Career/MLOps Agent)
    - 3.4 Development & Workshop: Agent 9 (Alumni & Executive Agent)
4. **Phase 4: Execution & Handover (Month 4)**
    - 4.1 Real-world Project Implementation (5 Pilot Projects)
    - 4.2 Final System Optimization & Security Check
    - 4.3 Training for Admins & Handover
    - 4.4 Project Closure & Final Report

Handover Mechanism

(อ้างอิงกระบวนการส่งมอบงานมาตรฐานสำหรับการพัฒนาซอฟต์แวร์)

- ☐ **Code Repository:** ส่งมอบ Source Code ทั้งหมดผ่าน Git Repository (เช่น GitHub/GitLab) โดยแยก Branch สำหรับ Development และ Production อย่างชัดเจน
- ☐ **Documentation:**
  - ☐ *Technical Specs:* เอกสาร Architecture Design, ER Diagram, และ API Swagger
  - ☐ *Credential Document:* ไฟล์รหัสผ่านและ Key ต่างๆ (ส่งมอบผ่านช่องทางที่ปลอดภัยและมีการเข้ารหัส)
- ☐ **Environment Transfer:** การโอนสิทธิ์ความเป็นเจ้าของ (Ownership Transfer) ของ Cloud Project และ Server Access (SSH Keys)
- ☐ **Knowledge Transfer:** การจัด Session สอนการใช้งาน (Admin Training) และส่งมอบวิดีโอการอบรมทั้งหมด

Key Risks and Constraints

- **Risks (ความเสี่ยง):**
  - ความซับซ้อนทางเทคนิคในการเชื่อมต่อระบบ Hybrid (On-premise กับ Cloud) อาจทำให้เกิดความล่าช้า
  - พื้นฐานความรู้ของนักศึกษาที่แตกต่างกัน อาจทำให้การอบรมบางหัวข้อต้องใช้เวลามากกว่าที่กำหนด
  - การเปลี่ยนแปลงนโยบายการใช้งานหรือราคาของ Cloud Provider ระหว่างดำเนินโครงการ
- **Constraints (ข้อจำกัด):**
  - ระยะเวลาดำเนินโครงการจำกัดที่ 4 เดือน (Time Constraint)
  - ข้อจำกัดด้านฮาร์ดแวร์ที่ไม่สามารถจัดซื้อ Server เพิ่มเติมได้

ต้องบริหารจัดการทรัพยากรเดิม (Resource Constraint)

Logistics Checklist

1. Infrastructure & Access Setup

- ☐ **GCP Project Creation:** สร้าง Google Cloud Project และเปิดใช้งาน Billing
- ☐ **Service Account Generation:** สร้าง Service Account สำหรับ Gemini API และ Cloud Run พร้อมดาวน์โหลด Key JSON
- ☐ **Quota Request:** ยื่นขอเพิ่ม Quota สำหรับ Gemini API (ยืนยัน PoC กับ DSPy ReAct Agent)
- ☐ **On-prem Server Access:** ตรวจสอบการเข้าถึง Server ภายใน (SSH/VPN) สำหรับติดตั้ง Kubernetes
- ☐ **Domain & SSL:** จดโดเมนเนมและเตรียม SSL Certificate สำหรับ Web Platform

2. Development Environment

- ☐ **GitHub Repository:** สร้าง Private Repository และกำหนด Branch protection rules
- ☐ **CI/CD Pipeline:** ตั้งค่า GitHub Actions หรือ Cloud Build สำหรับ Deploy เบื้องต้น
- ☐ **Dev Database:** สร้าง Database Instance สำหรับทีมพัฒนา
- ☐ **Design:** อนุมัติแบบร่างหน้าจอ (Mockup Sign-off) ก่อนเริ่มพัฒนา

3. Workshop & Venue Preparation

- ☐ **Poster:** จัดทำสื่อประชาสัมพันธ์ เพื่อเชิญชวนเข้าร่วมกิจกรรม
- ☐ **Venue Booking:** จองห้องจัด Workshop สำหรับ 9 ครั้ง (ระบุวันที่

ชัดเจนตามแผน)

- ☐ **User Data:** รวบรวมรายชื่อนักศึกษาและสร้างบัญชีผู้ใช้ระบบเบื้องต้น
- ☐ **WiFi/Network Test:** ทดสอบความเร็วอินเทอร์เน็ต ณ สถานที่จัดงาน (ต้องรองรับ 50+ Connections)
- ☐ **Power Outlets:** ตรวจสอบจุดปลั๊กไฟและเตรียมปลั๊กพ่วงให้เพียงพอสำหรับผู้เข้าอบรม
- ☐ **Online Meeting Link:** สร้าง Link ประชุม (Zoom/Google Meet) สำหรับกรณี Hybrid Workshop

4. Communication Tools

- ☐ **Line OA Setup:** สมัคร Line Official Account และยืนยัน Verified Account (ถ้ามี)
- ☐ **Email Sender:** ตั้งค่า SMTP หรือ Email API Service (เช่น SendGrid) สำหรับระบบแจ้งเตือน
- ☐ **Partnership:** ส่งจดหมายเชิญพันธมิตรเข้าร่วมโครงการและยืนยันโจทย์ (Problem Statements)

5. Administrative

- ☐ **Contract Signing:** ลงนามสัญญาจ้างทีมพัฒนาและที่ปรึกษา
- ☐ **Budget Approval:** ได้รับอนุมัติงวดเงินงวดแรกสำหรับการจัดซื้อ Cloud Credit และอุปกรณ์

Agent 1 (OA): ระบบรับสมัครและแปลงข้อมูลเรซูเม่ (OCR & Data Extraction)

ภายใต้โครงการจัดตั้งห้องปฏิบัติการเสมือนสำหรับปัญญาประดิษฐ์เชิงตัวแทน (Agentic Living Lab)

1. หลักการและเหตุผล

ในกระบวนการรับนักศึกษาเข้าสู่ระบบนิเวศนวัตกรรม ข้อมูลเริ่มต้นจากนักศึกษา (Student Profile) มักอยู่ในรูปแบบที่ไม่มีโครงสร้าง (Unstructured Data) เช่น ไฟล์ Resume PDF หรือ Transcript ส่งผลให้ไม่สามารถนำไปวิเคราะห์ต่อได้ โครงการย่อยนี้จึงมุ่งเน้นให้นักศึกษาพัฒนาระบบ "Onboarding Agent" ที่ทำหน้าที่เป็นด่านหน้าในการแปลงข้อมูลดิบเหล่านี้ให้เป็นข้อมูลที่มีโครงสร้าง (Structured Data) เพื่อปูพื้นฐานข้อมูลสำหรับการทำงานของ Agent ตัวอื่นๆ ในระบบ Living Lab ต่อไป

2. วัตถุประสงค์ของโครงการ

1. เพื่อให้นักศึกษามีทักษะในการจัดการข้อมูลที่ไม่มีโครงสร้าง (Unstructured Data Modeling) และการประมวลผลภาษาธรรมชาติ (NLP)
2. เพื่อพัฒนาระบบต้นแบบ Agent ที่สามารถอ่านและดึงข้อมูล (Enรตntity Extraction) จากเอกสารสมัครเข้าร่วมโครงการได้
3. เพื่อจัดเตรียมฐานข้อมูลนักศึกษา (Student Database) สำหรับใช้ในแพลตฟอร์ม Living Lab

3. ขอบเขตของงาน

1. การอบรมภาคทฤษฎี (Online): ปูพื้นฐาน Python สำหรับ Data Engineering และ SQL ขั้นสูง
2. การอบรมภาคปฏิบัติ (On-site Hackathon):
    - พัฒนา Agent เพื่อดึงข้อมูล Skills, GPA, และ Interests จากไฟล์ตัวอย่าง
    - ออกแบบ Schema ฐานข้อมูลเพื่อรองรับข้อมูลนักศึกษา
3. การเชื่อมต่อระบบ: นำ Agent ที่ได้ไปเชื่อมต่อกับระบบ Backend

(Django) จำลอง

4. การทำงานหลัก (Core Workflow)
    - การจัดการข้อมูลที่ไม่มีโครงสร้าง (Unstructured Data): ทำหน้าที่รับไฟล์เอกสารสมัครเข้าร่วมโครงการ เช่น Resume (PDF) หรือ Transcript ซึ่งเป็นข้อมูลที่วิเคราะห์ได้ยาก
    - การสกัดข้อมูล (Entity Extraction): แปลงข้อมูลดิบจากไฟล์เอกสารให้เป็นข้อมูลที่มีโครงสร้าง (Structured Data) เพื่อใช้เป็นฐานข้อมูลหลัก
    - การเตรียมรากฐานข้อมูล: จัดเตรียมฐานข้อมูลโปรไฟล์นักศึกษา (Student Database) เพื่อเป็น Input สำหรับการทำงานของ AI Agents ตัวอื่น ๆ ในระบบ Living Lab
5. ฟังก์ชันทางเทคนิค (Technical Features)
    - เทคโนโลยีการอ่านเอกสาร: ใช้เครื่องมืออย่าง PyPDF2 หรือ LangChain สำหรับการอ่านไฟล์ และใช้ Spacy หรือ NLTK สำหรับการทำ NLP (Natural Language Processing) เพื่อดึงข้อมูลสำคัญ
    - ข้อมูลที่สกัดได้: สามารถดึงข้อมูลทักษะ (Skills), เกรดเฉลี่ย (GPA), และความสนใจ (Interests) จากเอกสารได้โดยอัตโนมัติ
    - ระบบหลังบ้าน: พัฒนาด้วยภาษา Python 3.9+ เชื่อมต่อกับฐานข้อมูล PostgreSQL (Cloud SQL) และรันบนระบบ Google Cloud Run
6. รายละเอียดหน้าจอ UI (User Interface Requirements)
    - Upload Zone: ส่วนติดต่อผู้ใช้งานสำหรับอัปโหลดเอกสาร รองรับไฟล์ PDF
    - Data Verification Form: หน้าจอสำหรับแสดงผลข้อมูลที่สกัดได้ เพื่อให้นักศึกษาตรวจสอบความถูกต้องก่อนบันทึกเข้าสู่ระบบ

- Responsive Design: ตัวแพลตฟอร์มพัฒนาด้วย React รองรับการใช้งานผ่านหน้าจอทุกรูปแบบ

4. แผนการส่งมอบผลงาน

1. Source Code: โมดูล "Student Profile Generator" ที่ทำงานได้จริง
2. Database Schema: ผังฐานข้อมูลนักศึกษาที่พร้อมใช้งาน
3. Workshop Report: รายงานสรุปผลการทำงานของ Agent และความถูกต้องของการดึงข้อมูล
4. ตัวอย่างข้อมูลในการ Workshop

5. คุณสมบัติทางเทคนิค

- Language: Python 3.9+
- Database: PostgreSQL (Cloud SQL)
- Key Libraries: PyPDF2/LangChain (สำหรับการอ่านเอกสาร), Spacy/NLTK (สำหรับการสกัด Entity)
- Infrastructure: รันบน Cloud Run Service (Development Environment)

Agent 2 (CA): ระบบจัดหมวดหมู่และแนะนำหลักสูตร (Recommendation System)
ภายใต้โครงการจัดตั้งห้องปฏิบัติการเสมือนสำหรับปัญญาประดิษฐ์เชิงตัวแทน (Agentic Living Lab)

1. หลักการและเหตุผล
การพัฒนาทักษะทางด้านวิทยาการข้อมูลจำเป็นต้องมีความเข้าใจในความสัมพันธ์ของทักษะต่างๆ (Skill Relationships) เช่น ความเชื่อมโยงระหว่าง "Python" กับ "Data Visualization" การให้นักศึกษาเรียนรู้ที่จะสร้าง Agent ที่สามารถจัดหมวดหมู่และแนะนำเส้นทางการเรียนรู้ (Learning Path) ได้ จะช่วยให้แพลตฟอร์ม Living Lab มีความชาญฉลาดในการแนะนำหลักสูตรที่เหมาะสมกับช่องว่างทางทักษะ (Skill Gap) ของผู้เรียนแต่ละคน

2. วัตถุประสงค์ของโครงการ
1. เพื่อให้นักศึกษาเข้าใจหลักการจัดกลุ่มข้อมูล (Clustering) และโครงสร้างข้อมูลแบบลำดับชั้น (Hierarchical Data Structure)
2. เพื่อพัฒนา Agent ที่สามารถสร้าง "อนุกรมวิธานทักษะ" (Skill Taxonomy) จากข้อมูลดิบได้
3. เพื่อสร้างระบบแนะนำหลักสูตรเบื้องต้น (Basic Recommendation)

3. ขอบเขตของงาน
1. การอบรมภาคทฤษฎี (Online): เรียนรู้เรื่อง Hierarchical Clustering และ Taxonomy Design
2. การอบรมภาคปฏิบัติ (On-site Hackathon):
    - พัฒนา Agent ที่ใช้ Scikit-learn ในการจัดกลุ่มทักษะ (Skill Clustering)

- สร้าง Logic ในการตรวจสอบวิชาบังคับก่อน (Prerequisite Checking)

3. การทดสอบ: ทดสอบการจัดกลุ่มวิชาเรียนให้สอดคล้องกับ Profile นักศึกษาจากโครงการที่ 1
4. การทำงานหลัก (Core Workflow)
   - การจัดหมวดหมู่ทักษะ (Skill Clustering): นำข้อมูลทักษะดิบที่ได้จากโครงการที่ 1 มาวิเคราะห์และจัดกลุ่มตามความสัมพันธ์
   - สร้างอนุกรมวิธานทักษะ (Skill Taxonomy): สร้างโครงสร้างข้อมูลแบบลำดับชั้นเพื่อให้เห็นความเชื่อมโยงของทักษะต่าง ๆ (เช่น Python เชื่อมโยงกับ Data Visualization)
   - ระบบแนะนำหลักสูตร (Recommendation System): แนะนำเส้นทางการเรียนรู้ (Learning Path) ที่เหมาะสมเพื่อปิดช่องว่างทางทักษะ (Skill Gap) ของนักศึกษาแต่ละคน
   - ตรวจสอบวิชาบังคับก่อน (Prerequisite Checking): มี Logic ในการตรวจสอบลำดับการเรียนก่อน-หลังของแต่ละวิชา
5. ฟังก์ชันทางเทคนิค (Technical Features)
   - Algorithms: ใช้การจัดกลุ่มข้อมูลด้วย Hierarchical Clustering หรือ K-Means
   - Data Processing: ใช้ Library Scikit-learn และ Pandas ในการประมวลผล
   - Integration: รับ Input เป็น JSON รายชื่อทักษะ และส่งออกผลลัพธ์เป็น Cluster ID หรือข้อมูลแนะนำหลักสูตร
6. รายละเอียดหน้าจอ UI (User Interface Requirements)
   - Course Cluster Map: หน้าจอแสดงแผนผังความสัมพันธ์ของรายวิชาและทักษะต่าง ๆ แบบเห็นภาพ
   - Learning Path Display: ส่วนแสดงลำดับวิชาที่แนะนำให้

นักศึกษาเรียนตามลำดับก่อน-หลัง
    - API Endpoint Interface: ส่วนสำหรับเรียกดูข้อมูลรายวิชาที่แนะนำ (Mockup logic)

4. แผนการส่งมอบผลงาน

1. Skill Taxonomy Engine: โมดูลจัดหมวดหมู่ทักษะอัตโนมัติ
2. Course Cluster Map: แผนผังความสัมพันธ์ของรายวิชา
3. API Endpoint: API สำหรับเรียกดูรายวิชาที่แนะนำ (Mockup logic)
4. Source Code: โค้ดโปรแกรมทั้งหมดของระบบ CA
5. ตัวอย่างข้อมูลในการ Workshop

5. คุณสมบัติทางเทคนิค

- Algorithm: Hierarchical Clustering / K-Means
- Framework: Scikit-learn, Pandas
- Integration: สามารถรับ Input เป็น JSON รายชื่อทักษะ และส่งออกเป็น Cluster ID ได้

Agent 3 (MA): ระบบจับคู่นักศึกษากับโจทย์งาน (Matching Algorithm ผ่าน Knowledge Graph)

1. หลักการและเหตุผล

หัวใจสำคัญของ Agentic Living Lab คือการจับคู่นักศึกษาที่มีศักยภาพให้ตรงกับโจทย์ปัญหาจริงจากภาคอุตสาหกรรม (Real-world Projects) การจับคู่ด้วย Keyword เพียงอย่างเดียวไม่เพียงพอ จำเป็นต้องใช้เทคโนโลยี Knowledge Graph และ Vector Similarity เพื่อค้นหาความสัมพันธ์ที่ซับซ้อนและจัดลำดับความเหมาะสม (Ranking) ได้อย่างแม่นยำ นี่คือโครงการที่สร้าง "สมอง" ของระบบ

2. วัตถุประสงค์ของโครงการ

1. เพื่อให้นักศึกษามีทักษะในการใช้งาน Graph Database และ Vector Search
2. เพื่อพัฒนาระบบจับคู่ (Matching System) ที่ให้คะแนนความเหมาะสม (Scoring) ได้
3. เพื่อเชื่อมโยงข้อมูลนักศึกษา (จาก OA) และข้อมูลทักษะ (จาก CA) เข้ากับโจทย์โครงการ

3. ขอบเขตของงาน

1. การอบรมภาคทฤษฎี (Online): ความรู้เรื่อง Graph Theory, Cypher Query Language และ Vector Embeddings
2. การอบรมภาคปฏิบัติ (On-site Hackathon):
    - สร้าง Knowledge Graph เชื่อมโยง Students ↔ Skills ↔ Projects
    - พัฒนา Algorithm ในการคำนวณคะแนนความเหมือน (Vector Similarity Score)
3. การทดสอบ: ทดสอบจับคู่นักศึกษากับโจทย์จำลองและประเมินความแม่นยำ
4. การทำงานหลัก (Core Workflow)
    - สร้างโครงข่ายความสัมพันธ์ (Knowledge Graph Construction): ทำการเชื่อมโยงข้อมูลสามส่วนหลักเข้าด้วยกัน คือ นักศึกษา (Students), ทักษะ (Skills), และโจทย์โครงการ (Projects) เพื่อให้เห็นภาพรวมของระบบนิเวศ
    - คำนวณความเหมาะสม (Similarity Scoring): ใช้เทคโนโลยี Vector Search เพื่อหาความเหมือนระหว่างคุณสมบัติของนักศึกษากับความต้องการของโครงการ

- จัดลำดับการจับคู่ (Ranking & Recommendation): ประมวลผลและให้คะแนนความเหมาะสม (Scoring) เพื่อคัดเลือกนักศึกษาที่ตรงกับโจทย์งานมากที่สุด

5. ฟังก์ชันทางเทคนิค (Technical Features)
   - Graph Database: ใช้ Neo4j หรือ Library NetworkX ในการจัดเก็บและคัดกรองความสัมพันธ์ที่ซับซ้อน
   - Vector Search: ใช้ FAISS หรือ pgvector บน PostgreSQL เพื่อประมวลผลการค้นหาเชิงความหมาย (Semantic Search)
   - Logic: ใช้ Cosine Similarity ในการคำนวณคะแนน Match Score เพื่อความแม่นยำในการจับคู่
6. รายละเอียดหน้าจอ UI (User Interface Requirements)
   - Matching Dashboard: หน้าจอแสดงผลการจับคู่ระหว่างนักศึกษาและโครงการ พร้อมแสดงคะแนนความเหมาะสมเป็นเปอร์เซ็นต์
   - Relationship Visualization: การแสดงผลกราฟความสัมพันธ์ (Graph View) ที่ทำให้เห็นว่านักศึกษาคนใดเชื่อมโยงกับทักษะและโปรเจกต์ไหนบ้าง
   - Accuracy Report: หน้าจอรายงานผลการทดสอบความแม่นยำของการจับคู่ เพื่อให้แอดมินปรับปรุง Logic ได้

4. แผนการส่งมอบผลงาน

1. Knowledge Graph: ฐานข้อมูลกราฟที่แสดงความสัมพันธ์ของระบบนิเวศ
2. Matching Scoring System: ระบบให้คะแนนความเหมาะสมในการจับคู่
3. Performance Report: รายงานผลการทดสอบความแม่นยำของการจับคู่

4. Source Code: ชุดคำสั่งทั้งหมดสำหรับการทำ Graph และ Matching Logic
5. ตัวอย่างข้อมูลในการ Workshop

5. คุณสมบัติทางเทคนิค

- Graph DB: Neo4j หรือ NetworkX (Python Library)
- Vector Search: FAISS หรือ pgvector (บน PostgreSQL)
- Logic: ใช้ Cosine Similarity ในการคำนวณคะแนน Match Score

Agent 4 (DCSA): ระบบเชื่อมต่อ API ภายนอกและชุมชน (External API Integrator)
ภายใต้โครงการจัดตั้งห้องปฏิบัติการเสมือนสำหรับปัญญาประดิษฐ์เชิงตัวแทน (Agentic Living Lab)

1. หลักการและเหตุผล
ในโลกความเป็นจริง Agent ไม่ได้ทำงานอยู่เพียงลำพัง แต่ต้องปฏิสัมพันธ์กับโลกภายนอกผ่าน API (Application Programming Interface) โครงการนี้มุ่งเน้นการเปลี่ยน Agent จาก "ผู้คิด" เป็น "ผู้กระทำ" โดยการสร้างเครื่องมือ (Tools) ให้ Agent สามารถเชื่อมต่อกับบริการภายนอก เช่น การเช็คสต็อกอุปกรณ์ การดึงข้อมูลสภาพอากาศ หรือการเชื่อมต่อกับระบบบริการท้องถิ่น เพื่อสนับสนุนชุมชนดิจิทัล

2. วัตถุประสงค์ของโครงการ
1. เพื่อให้นักศึกษาเข้าใจหลักการ API Integration และ Service Orchestration
2. เพื่อพัฒนา Agent ที่สามารถใช้ "Tools" (API Wrapper) ในการดึงข้อมูลจากภายนอกได้
3. เพื่อสร้าง Bot ที่สามารถแปลงคำสั่งภาษาธรรมชาติเป็น API Call

3. ขอบเขตของงาน
1. การอบรมภาคทฤษฎี (Online): การออกแบบ RESTful API และการสร้าง Python Wrapper
2. การอบรมภาคปฏิบัติ (On-site Hackathon):
    - ออกแบบ Tool ให้กับ Agent (เช่น inventory_check_tool)
    - เชื่อมต่อ Agent กับ Mock API ของบริการชุมชน
    - ทดสอบให้ Agent ตอบคำถามโดยใช้ข้อมูล Real-time จาก API
3. การประยุกต์: สร้าง Chatbot เบื้องต้นที่ตอบคำถามเกี่ยวกับทรัพยากร

ห้องแล็บได้

4. การทำงานหลัก (Core Workflow)
    - การเชื่อมต่อบริการภายนอก (External API Integration): ทำหน้าที่เป็นตัวกลางในการติดต่อกับบริการข้างนอก เช่น ระบบเช็คสต็อกอุปกรณ์, ข้อมูลสภาพอากาศ หรือบริการชุมชนดิจิทัล
    - การแปลงคำสั่ง (Natural Language to API Call): รับคำสั่งเป็นภาษาธรรมชาติจากผู้ใช้ แล้วแปลงเป็นคำสั่งเรียกใช้งาน API (API Wrapper) เพื่อดึงข้อมูลจริงแบบ Real-time
    - การสนับสนุนชุมชน (Community Support): ทำหน้าที่เป็น Bot ที่คอยตอบคำถามและช่วยเหลือเกี่ยวกับทรัพยากรภายในห้องปฏิบัติการ (Lab Resources)
5. ฟังก์ชันทางเทคนิค (Technical Features)
    - API Wrapper Tool: พัฒนาชุดโค้ด Python (Requests Library) สำหรับเชื่อมต่อและจัดการโปรโตคอล REST/JSON
    - Service Orchestration: บริหารจัดการลำดับการเรียกใช้ Service ต่างๆ เพื่อให้ได้คำตอบที่สมบูรณ์
    - Framework: ใช้งาน FastAPI สำหรับจำลอง Mock Server และใช้ DSPy ในการสั่งงานเอเจนต์ให้ทำงานตามโจทย์
6. รายละเอียดหน้าจอ UI (User Interface Requirements)
    - Community Service Bot Interface: หน้าจอแชทบอทที่ผู้ใช้สามารถพิมพ์สอบถามข้อมูลเกี่ยวกับทรัพยากรหรือบริการชุมชนได้
    - Tool Status Dashboard: ส่วนแสดงสถานะการเชื่อมต่อ API ต่างๆ (API Health Check) และคู่มือการใช้งาน (Documentation)

4. แผนการส่งมอบผลงาน

1. **API Wrapper Tool:** ชุดโค้ด Python สำหรับเชื่อมต่อบริการภายนอก
2. **Community Service Bot:** โปรแกรม Agent ที่สามารถตอบคำถามโดยอ้างอิงข้อมูลจาก API
3. **Documentation:** คู่มือการใช้งาน API Tool
4. **Source Code**

5. ตัวอย่างข้อมูลในการ Workshop

5. คุณสมบัติทางเทคนิค

- **Language:** Python (Requests Library)
- **Protocol:** REST / JSON
- **Framework:** FastAPI (สำหรับทำ Mock Server), DSPy (สำหรับสั่งงาน Agent)

Agent 5 (EOA): ระบบจัดการอีเวนต์และตารางเวลาแบบอัตโนมัติ (Autonomous Event Planner ด้วย ReAct Pattern)
ภายใต้โครงการจัดตั้งห้องปฏิบัติการเสมือนสำหรับปัญญาประดิษฐ์เชิงตัวแทน (Agentic Living Lab)

1. หลักการและเหตุผล
การทำงานจริงมักมีความซับซ้อนและประกอบด้วยหลายขั้นตอน (Multi-step workflows) โครงการนี้จะยกระดับความสามารถของ Agent ด้วยการใช้ **ReAct Pattern** (Reasoning + Acting) เพื่อให้ Agent สามารถวางแผน ตัดสินใจ และแก้ไขปัญหาเฉพาะหน้าได้ เช่น การจัดงาน Hackathon ที่ต้องมีการเช็คตาราง จองห้อง และส่งอีเมลเชิญ หากขั้นตอนใดล้มเหลว Agent ต้องสามารถตัดสินใจทางเลือกอื่นได้

2. วัตถุประสงค์ของโครงการ
1. เพื่อให้นักศึกษาเข้าใจรูปแบบ ReAct Pattern และการจัดการสถานะ (State Management)
2. เพื่อพัฒนา Agent ที่สามารถร้อยเรียงเครื่องมือหลายชิ้น (Tool Chaining) เพื่อทำงานที่ซับซ้อน
3. เพื่อสร้างระบบ Autonomous Event Planner ต้นแบบ

3. ขอบเขตของงาน
1. การอบรมภาคทฤษฎี (Online): เจาะลึก DSPy Framework และ Logic การตัดสินใจของ AI
2. การอบรมภาคปฏิบัติ (On-site Hackathon):
    - สร้าง Agent ที่ทำหน้าที่ "Event Organizer"
    - กำหนด Workflow: ตรวจงบประมาณ -> จองสถานที่ -> ส่งเมลเชิญ

- จำลองสถานการณ์ความล้มเหลว (เช่น งบไม่พอ) เพื่อให้ Agent ตัดสินใจแก้ปัญหา

3. การเชื่อมต่อ: เชื่อมต่อกับระบบปฏิทินและอีเมลจำลอง
4. การทำงานหลัก (Core Workflow)
   - การวางแผนและตัดสินใจ (Reasoning & Acting): ใช้ ReAct Pattern เพื่อให้เอเจนต์สามารถคิด (Reason) และลงมือทำ (Act) เป็นลำดับขั้นตอน
   - การจัดการงานที่ซับซ้อน (Multi-step Workflows): สามารถร้อยเรียงเครื่องมือหลายชิ้น (Tool Chaining) เพื่อทำงานใหญ่ให้สำเร็จ เช่น การจัดงาน Hackathon
   - การจัดการสถานะและกู้คืนข้อผิดพลาด (Error Recovery): ตรวจสอบสถานะการทำงาน (State Management) และหากขั้นตอนใดล้มเหลว (เช่น งบประมาณไม่พอ) เอเจนต์ต้องตัดสินใจหาทางเลือกอื่นได้เอง
5. ฟังก์ชันทางเทคนิค (Technical Features)
   - Framework: พัฒนาโดยใช้ DSPy หรือ LangChain เพื่อควบคุม Logic การตัดสินใจของ AI
   - Decision Logic: ใช้ระบบ State Machine เพื่อควบคุมสถานะของ Workflow ในแต่ละขั้นตอน
   - Integrations: เชื่อมต่อกับระบบปฏิทิน (Calendar) และระบบอีเมล (Email) จำลองเพื่อส่งคำเชิญและจองสถานที่
6. รายละเอียดหน้าจอ UI (User Interface Requirements)
   - Autonomous Planner Dashboard: หน้าจอสำหรับสั่งการและดูแผนการทำงานของเอเจนต์
   - Workflow Traces: ส่วนแสดงบันทึกการตัดสินใจของเอเจนต์ (Reasoning Traces) เพื่อให้มนุษย์เข้าใจว่า AI คิดและทำอย่างไร

ในแต่ละขั้นตอน

- Scenario Simulator: ระบบจำลองสถานการณ์ความล้มเหลวเพื่อทดสอบการแก้ปัญหาของเอเจนต์

4. แผนการส่งมอบผลงาน

1. **Autonomous Event Planner:** ระบบ Agent ที่จัดการงานหลายขั้นตอนได้อัตโนมัติ
2. **Workflow Logs:** บันทึกการตัดสินใจของ Agent (Reasoning Traces)
3. **Demo Scenario:** การสาธิตการกู้คืนสถานการณ์ (Error Recovery) ของ Agent
4. **Source Code**
5. **ตัวอย่างข้อมูลในการ Workshop**

5. คุณสมบัติทางเทคนิค

- **Framework:** DSPy หรือ LangChain
- **Pattern:** ReAct (Reason + Act)
- **Logic:** State Machine สำหรับควบคุมสถานะของ Workflow

Agent 6 (PM): ระบบผู้ช่วยบริหารโครงการ (Project Tracking Assistant และ Automated Documentation)
ภายใต้โครงการจัดตั้งห้องปฏิบัติการเสมือนสำหรับปัญญาประดิษฐ์เชิงตัวแทน (Agentic Living Lab)

1. หลักการและเหตุผล

การบริหารจัดการโครงการ Data Science จำเป็นต้องมีธรรมาภิบาลและการติดตามผลที่ดี โครงการย่อยนี้มุ่งเน้นการสร้าง Agent ที่ทำหน้าที่เสมือน "Scrum Master" หรือผู้ช่วยบริหารโครงการ โดยจะเชื่อมต่อกับระบบ Backend ของ Living Lab เพื่อติดตามสถานะงานบน Kanban Board อัปเดตความคืบหน้า และร่างเอกสารสำคัญ (Project Charter) ให้อัตโนมัติ ลดภาระงานเอกสารของมนุษย์

2. วัตถุประสงค์ของโครงการ

1. เพื่อให้นักศึกษาเข้าใจกระบวนการ Agile/Kanban และ MLOps Governance
2. เพื่อพัฒนา Agent ที่สามารถอ่านและเขียนข้อมูลลงในระบบบริหารจัดการโครงการ (Django Backend)
3. เพื่อสร้างระบบสร้างเอกสารอัตโนมัติ (Automated Documentation)

3. ขอบเขตของงาน

1. การอบรมภาคทฤษฎี (Online): หลักการ Project Management และ Django Signals
2. การอบรมภาคปฏิบัติ (On-site Hackathon):
    - สร้าง PM Assistant Agent
    - พัฒนา Logic ให้ Agent เฝ้าติดตาม Kanban Board
    - เมื่อมีการเริ่มโปรเจกต์ ให้ Agent ร่าง Project Charter (Scope, KPIs) จากข้อมูลดิบ

3. การเชื่อมต่อ: เชื่อมต่อ Agent เข้ากับ Database ของ Living Lab ผ่าน ORM หรือ API
4. การทำงานหลัก (Core Workflow)
    - การติดตามสถานะงาน (Project Tracking): ทำหน้าที่เฝ้าติดตามสถานะของงานบน Kanban Board อย่างใกล้ชิดผ่านระบบ Backend
    - การสร้างเอกสารอัตโนมัติ (Automated Documentation): เมื่อมีการเริ่มต้นโปรเจกต์ใหม่ เอเจนต์จะทำหน้าที่ร่างเอกสารสำคัญ เช่น Project Charter (ระบุขอบเขตและ KPIs) จากข้อมูลดิบให้อัตโนมัติ เพื่อลดภาระงานเอกสารของมนุษย์
    - การเชื่อมต่อฐานข้อมูล: ทำงานร่วมกับระบบฐานข้อมูลของ Living Lab เพื่ออัปเดตความคืบหน้าและดึงข้อมูลมาประมวลผล
5. ฟังก์ชันทางเทคนิค (Technical Features)
    - Framework: พัฒนาโดยใช้ Django (อาศัย Signals และ Commands) ในการตรวจจับการเปลี่ยนแปลงของข้อมูล
    - Background Worker: ทำหน้าที่เป็นโปรแกรมส่วนหลังที่คอยตรวจสอบสถานะโครงการตลอดเวลา
    - Reporting Generator: มีฟังก์ชันสำหรับการสร้างรายงานสรุปผลอัตโนมัติ (Automated Reporting Generator)
6. รายละเอียดหน้าจอ UI (User Interface Requirements)
    - Kanban Board Integration: ส่วนติดต่อผู้ใช้งานที่แสดงสถานะงานที่เอเจนต์กำลังติดตามอยู่
    - Document Preview: หน้าจอสำหรับแสดงร่างเอกสาร Project Charter ที่ AI สร้างขึ้น เพื่อให้ผู้ใช้ตรวจสอบและแก้ไข
    - Status Dashboard: ส่วนแสดงภาพรวมความคืบหน้าของโครงการต่าง ๆ ในระบบ

4. แผนการส่งมอบผลงาน

1. PM Assistant Agent: ระบบผู้ช่วยบริหารโครงการ
2. Automated Project Charter: ตัวอย่างเอกสารที่สร้างโดย AI
3. Integration Module: โค้ดส่วนเชื่อมต่อระหว่าง Agent และ Django Backend

5. คุณสมบัติทางเทคนิค

- **Framework:** Django (Signals & Commands)
- **Feature:** Automated Reporting Generator
- **Role:** ทำหน้าที่เป็น Background Worker คอยตรวจสอบสถานะโครงการ

Agent 7 (SWA): ระบบดูแลสุขภาวะและ Support Desk อัจฉริยะ (Text Classification และ Sentiment Analysis)
ภายใต้โครงการจัดตั้งห้องปฏิบัติการเสมือนสำหรับปัญญาประดิษฐ์เชิงตัวแทน (Agentic Living Lab)

1. หลักการและเหตุผล
เมื่อแพลตฟอร์มเริ่มใช้งานจริง (First Deployment) การดูแลผู้ใช้งานและการจัดการข้อร้องเรียนเป็นเรื่องสำคัญ โดยเฉพาะในบริบทของมหาวิทยาลัยที่ต้องใส่ใจสุขภาวะ (Well-being) ของนักศึกษา โครงการนี้จะสร้าง Agent ที่ทำหน้าที่ "Intelligent Support Desk" คอยคัดกรองข้อความ จำแนกเจตนา (Intent) ว่าเป็นปัญหาเทคนิคทั่วไป หรือปัญหาความเครียด/สุขภาวะ เพื่อส่งต่อให้ถูกต้องตามหลักจริยธรรม

2. วัตถุประสงค์ของโครงการ
1. เพื่อให้นักศึกษาตระหนักถึงจริยธรรม AI (AI Ethics) และความเป็นส่วนตัวของข้อมูล (Data Privacy)
2. เพื่อพัฒนา Agent ที่สามารถจำแนกประเภทข้อความ (Text Classification) และอารมณ์ (Sentiment Analysis)
3. เพื่อเตรียมความพร้อมระบบสำหรับการเปิดใช้งานจริง (Production Readiness)

3. ขอบเขตของงาน
1. การอบรมภาคทฤษฎี (Online): จริยธรรม AI, Bias Detection และเทคนิค NLP Classification
2. การอบรมภาคปฏิบัติ (On-site Hackathon):
    - พัฒนา Triage Agent ที่จำแนก Ticket: "IT Issue" vs "Well-being"
    - กำหนด Workflow การส่งต่อข้อมูลที่ปลอดภัย (Secure Routing)

- ทดสอบกับข้อมูลจำลอง (Mock Sensitive Data)

3. การติดตั้ง: Deploy Agent ขึ้นสู่ระบบ Cloud จริงเพื่อเตรียมใช้งาน
4. การทำงานหลัก (Core Workflow)
    - คัดกรองและจำแนกประเภท (Intelligent Support Desk): ทำหน้าที่เป็นด่านแรกในการรับเรื่องร้องเรียน โดยจำแนกเจตนา (Intent) ของข้อความว่าเกี่ยวข้องกับปัญหาทางเทคนิคทั่วไป หรือเป็นปัญหาด้านความเครียดและสุขภาวะ
    - วิเคราะห์อารมณ์และเจตนา (Triage Agent): ใช้การจำแนกประเภทข้อความ (Text Classification) และการวิเคราะห์อารมณ์ (Sentiment Analysis) เพื่อส่งต่อ Ticket ให้กับหน่วยงานที่เกี่ยวข้องได้อย่างถูกต้องตามหลักจริยธรรม
    - รักษาความปลอดภัยและจริยธรรม: คุ้มครองความเป็นส่วนตัวของข้อมูลผู้ใช้งาน โดยเฉพาะข้อมูลที่ละเอียดอ่อน (Sensitive Data) ผ่านกระบวนการส่งต่อข้อมูลที่ปลอดภัย
5. ฟังก์ชันทางเทคนิค (Technical Features)
    - AI Model: ใช้ NLP Classifier ที่ผ่านการทำ Fine-tuned หรือ Prompt Engineering เพื่อให้มีความแม่นยำในการคัดกรอง
    - Security & Privacy: มีระบบ Data Masking และ Logic สำหรับ PII Protection (Personally Identifiable Information) เพื่อปกปิดข้อมูลระบุตัวตน
    - Platform & Infrastructure: ระบบถูกติดตั้งและรันอยู่บน Google Cloud Run เพื่อเตรียมพร้อมสำหรับการใช้งานจริง (Production Readiness)
6. รายละเอียดหน้าจอ UI (User Interface Requirements)
    - Intelligent Support Desk Module: ส่วนติดต่อผู้ใช้งานสำหรับรับเรื่องร้องเรียนที่สามารถตอบสนองตามประเภทของปัญหาได้

- Secure Routing Interface: ระบบหลังบ้านที่แสดงการส่งต่อ Ticket ไปยังเจ้าหน้าที่เทคนิคหรือผู้เชี่ยวชาญด้านสุขภาวะอย่างปลอดภัย
- Ethics & Privacy Dashboard: ส่วนแสดงรายงานมาตรการความปลอดภัยและสถานะการจัดการจริยธรรมของระบบ

4. แผนการส่งมอบผลงาน

1. Intelligent Support Desk Module: ระบบรับเรื่องร้องเรียนอัจฉริยะ
2. Ethics & Privacy Report: รายงานมาตรการความปลอดภัยและจริยธรรมของระบบ
3. Deployed Service: ระบบที่รันอยู่บน Production Environment
4. Source Code: ชุดคำสั่งทั้งหมดของระบบ SWA

5. คุณสมบัติทางเทคนิค

- Model: NLP Classifier (Fine-tuned หรือ Prompt Engineering)
- Security: Data Masking / PII Protection logic
- Platform: Google Cloud Run

โครงการย่อยที่ 8: การอบรมเชิงปฏิบัติการพัฒนา "Career Agent (CRA)"
ภายใต้โครงการจัดตั้งห้องปฏิบัติการเสมือนสำหรับปัญญาประดิษฐ์เชิงตัวแทน (Agentic Living Lab)

1. หลักการและเหตุผล
เป้าหมายปลายทางของ Living Lab คือการสร้างพอร์ตโฟลิโอ (Portfolio) ให้นักศึกษาพร้อมเข้าสู่ตลาดแรงงาน โครงการนี้เน้นทักษะขั้นสูงด้าน **MLOps** และ **CI/CD** (Continuous Integration/Continuous Deployment) โดยสร้าง Agent ที่ช่วยจัดการกระบวนการนำโมเดล AI ของนักศึกษาจากเครื่องคอมพิวเตอร์ส่วนตัว ขึ้นสู่เซิร์ฟเวอร์สาธารณะอย่างเป็นระบบและปลอดภัย เปลี่ยนงานโปรเจกต์ให้เป็น "Portfolio Ready"

2. วัตถุประสงค์ของโครงการ
1. เพื่อให้นักศึกษาเข้าใจกระบวนการ MLOps และ CI/CD Pipeline
2. เพื่อพัฒนา Agent ที่สามารถ Trigger กระบวนการ Deploy อัตโนมัติเมื่อผ่านการตรวจสอบ
3. เพื่อเชื่อมโยงข้อมูลจาก Knowledge Graph มาสร้าง Context ให้กับการจัดเก็บเวอร์ชันโมเดล

3. ขอบเขตของงาน
1. **การอบรมภาคทฤษฎี (Online):** หลักการ Containerization (Docker), Git, และ CI/CD
2. **การอบรมภาคปฏิบัติ (On-site Hackathon):**
    - เขียน Pipeline Script (YAML) สำหรับ Deploy งาน
    - พัฒนา Agent ที่ทำหน้าที่ "Release Manager" ตรวจสอบคุณภาพโค้ดก่อน Deploy

    - เชื่อมโยงผลงานเข้ากับประวัตินักศึกษาในระบบ Living Lab
3. **การส่งมอบงาน:** นักศึกษาทำการ Deploy โปรเจกต์จริงขึ้นระบบ

4. แผนการส่งมอบผลงาน

1. **Automated Portfolio Deployer:** ระบบ Deploy อัตโนมัติด้วย Agent
2. **CI/CD Pipelines:** สคริปต์การทำงานของระบบ Deploy
3. **Student Portfolios:** ผลงานนักศึกษาที่เข้าถึงได้ผ่านเว็บ (Live URL)
4. **Source Code:**

5. คุณสมบัติทางเทคนิค

- **Tools:** Git, Docker, Google Cloud Build
- **Architecture:** Microservices
- **Agent Role:** ตรวจสอบความครบถ้วนของ Artifacts ก่อนสั่งรัน Pipeline

Agent 9 (AEA): ระบบรายงานผลผู้บริหาร (Executive Dashboard และ Narrative Generation)
ภายใต้โครงการจัดตั้งห้องปฏิบัติการเสมือนสำหรับปัญญาประดิษฐ์เชิงตัวแทน (Agentic Living Lab)

1. หลักการและเหตุผล

การวัดผลความสำเร็จของโครงการต้องอาศัยการวิเคราะห์ข้อมูลและนำเสนอต่อผู้บริหาร (Executive Reporting) โครงการสุดท้ายนี้จะเปลี่ยนบทบาทนักศึกษาเป็น Data Analyst/Storyteller โดยสร้าง Agent ที่สามารถดึงข้อมูลย้อนหลัง สรุปผลกระทบ (Impact Metrics) และสร้าง Dashboard หรือบทสรุปสำหรับผู้บริหารได้โดยอัตโนมัติ เป็นการปิดวงจรโครงการอย่างสมบูรณ์

2. วัตถุประสงค์ของโครงการ

1. เพื่อให้นักศึกษามีทักษะในการเล่าเรื่องด้วยข้อมูล (Data Storytelling) และการใช้ BI Tools
2. เพื่อพัฒนา Agent ที่สามารถ Query ข้อมูลสรุปผลและสร้างคำบรรยาย (Narrative Generation) ได้
3. เพื่อจัดทำรายงานสรุปผลภาพรวมของโครงการ Living Lab

3. ขอบเขตของงาน

1. การอบรมภาคทฤษฎี (Online): การใช้งาน Apache Superset และเทคนิค NLG (Natural Language Generation)
2. การอบรมภาคปฏิบัติ (On-site Hackathon):
    - เชื่อมต่อ Agent เข้ากับ Data Warehouse ของระบบ
    - สร้าง Dashboard แสดงผล KPIs (จำนวนโปรเจกต์, ทักษะที่เกิดขึ้น)

    - ให้ Agent เขียนบทสรุปผู้บริหาร (Executive Summary) จากข้อมูลใน Dashboard
3. การนำเสนอ: นำเสนอผลงานภาพรวมทั้งหมดต่อคณะกรรมการ
4. การทำงานหลัก (Core Workflow)
    - การวิเคราะห์ผลกระทบ (Impact Analysis): ทำหน้าที่ดึงข้อมูลย้อนหลังจากการดำเนินงานทั้งหมดเพื่อสรุปตัววัดความสำเร็จ (Impact Metrics)
    - การสร้างคำบรรยายอัตโนมัติ (Narrative Generation): ใช้เทคนิคการสร้างภาษาธรรมชาติ (NLG) เพื่อเปลี่ยนตัวเลขและสถิติให้เป็นบทสรุปผู้บริหาร (Executive Summary) ที่อ่านเข้าใจง่าย
    - การรายงานผลภาพรวม: จัดทำรายงานสรุปผลภาพรวมของโครงการ Living Lab เพื่อนำเสนอต่อคณะกรรมการและปิดวงจรโครงการอย่างสมบูรณ์
5. ฟังก์ชันทางเทคนิค (Technical Features)
    - Data Connection: เชื่อมต่อเอเจนต์เข้ากับ Data Warehouse ของระบบเพื่อ Query ข้อมูลสรุปผล
    - BI & Analytics Tools: ใช้เครื่องมือ Business Intelligence เช่น Apache Superset หรือ Tableau ในการจัดการข้อมูล
    - Language Tech: ใช้ LLM สำหรับการสรุปความ (Text Summarization) และการรวมข้อมูลด้วย SQL Aggregation
    - Security: ระบบรองรับการสร้าง Secure Shareable Links สำหรับการเข้าถึงรายงานอย่างปลอดภัย
6. รายละเอียดหน้าจอ UI (User Interface Requirements)
    - Executive Impact Dashboard: หน้าจอแสดงผลตัวชี้วัดความสำเร็จ (KPIs) แบบ Real-time เช่น จำนวนโปรเจกต์ที่สำเร็จ และทักษะที่นักศึกษาพัฒนาขึ้น

- Newsletter Generator Interface: ส่วนสำหรับตั้งค่าและสร้างจดหมายข่าวสรุปผลงานอัตโนมัติ (Automated Newsletter) เพื่อส่งอีเมลแจ้งเตือนผลลัพธ์โครงการ
- Final Report Preview: หน้าจอสำหรับตรวจสอบรายงานสรุปโครงการฉบับสมบูรณ์ที่สร้างโดย AI

4. แผนการส่งมอบผลงาน

1. Executive Impact Dashboard: หน้าจอสรุปผลการดำเนินงานแบบ Real-time
2. Automated Newsletter Generator: ระบบสร้างจดหมายข่าวสรุปผลงานอัตโนมัติ
3. Final Project Report: รายงานปิดโครงการฉบับสมบูรณ์
4. Source Code: ชุดคำสั่งทั้งหมดที่ใช้ในระบบ AEA

-5. คุณสมบัติทางเทคนิค

- **BI Tool:** Apache Superset หรือ Tableau
- **Tech:** SQL Aggregation, LLM for Text Summarization
- **Output:** Secure Shareable Links และ Email Content