Table of Contents

Table of Contents

1. 🔊 Applied GenAI - Deep Dive Podcast
2. 🧰 Toolbox: AI & Automation Tools

First Week

1. 🤖 My Personal Assistant
2. 📰 Generative AI News Dashboard
3. 🧠 AI Fluency Foundation

🏗️ Industry Project – Real-World Challenge & Collaboration

1. ✏️ Design Challenge
2. 🤖 Designing Your AI Teammate
3. GreenPT
4. Nederlandse Spoorwegen (NS)
5. Brainstax
6. Nederlands Onderwijsinstituut (Neon)

🧪 Professional Research Project

1. 🚀 Share Your Research: GitHub Showcase
2. ⚖️ AI Judge
3. 🤖 Autonomous Agents
4. 📊 Benchmarking GenAI
5. 🧠 Brain in Pieces
6. 🌱 Building the Future, Sustainably
7. 💻 Code Whisperer
8. 🎭 Context-Aware Personalisation
9. 🧠 Context & Memory
10. 🎨 Creative AI Studio
11. 💾 Domain Expert

12. 😊 Emotion Machines
13. 🤝 Human in the Loop
14. 🔍 Inside the Mind of a Model
15. ♿ Inclusive by Design
16. 🤖 Multi-Agent Mayhem
17. 🤖 Model Inference
18. 📊 Multimodel Mixmaster
19. 🧠 Ontology Collapse
20. 📱 Pocket Sized Geniuses
21. 🔐 Prompt Injection
22. 🔍 RAG Time
23. 🤖 Reinforced Learning
24. 🧠 Scaffolding Minds
25. 💻 Small Language Models
26. 🏙️ Smart City Applications
27. 📖 Storytelling & Brand Experience
28. 🎮 Tame the Weights
29. ⏳ Temporal Minds
30. 🔬 Transformer Deep Dive
31. 🪞 The Mirror Oracles
32. 🙄 The Reluctant Assistant
33. 🔊 The Voice Awakens
34. 🛠️ Tool Use
35. 🚂 Train of Thought
36. 🤔 Who Do You Think I Am?

🪞 Reflection & Submission – Track Your Progress

1. 🗣️ Week 6: Portfolio Check-in Conversation 1
2. 🗣️ Week 11: Portfolio Check-in Conversation 2

3. 🗣️ Week 16: Portfolio Check-in Conversation 3
4. 📝 Week 18: Final Portfolio Submission

🌿 Final Grade – Portfolio Review Outcome

1. 📈 Final Grade – Your Assessed Portfolio

Consent for Participation in AI-Assisted Feedback & Assessment Process

1. 🔐 AI-Assisted Feedback Processing Consent
2. Consent Agreement

Syllabus

No due date
Consent Agreement

Due Oct 3 at 12am
🗣 Week 6: Portfolio Check-in Conversation 1

Due Nov 14 at 12am
🗣 Week 11: Portfolio Check-in Conversation 2

Due Jan 2, 2026 at 12am
🗣 Week 16: Portfolio Check-in Conversation 3

Due Jan 16, 2026 at 12am
📝 Week 18: Final Portfolio Submission

Due Jan 30, 2026 at 12am
📈 Final Grade – Your Assessed Portfolio

Back to Table of Contents

Announcements

There are no announcements for this course.

Course Foundations

1. New This Semester

Back to Table of Contents

🆕 New This Semester

🆕 Unboxing Applied GenAI

You're the first to unbox this brand-new learning experience

Applied GenAI Art Deco

🎁 Fresh Out of the Box

The Applied GenAI semester is a brand new offering, fresh out of its packaging and ready for you to explore. Just like the product being unboxed in the image above, this semester is a first edition—carefully designed but still with that fresh, new-product feeling.

🚀 What Makes This Special

🏗️ Pioneer Cohort

You're not just students—you're explorers charting new territory. As our first cohort, you'll help shape the future of GenAI education at FHICT.

⚙️ Co-Creation Opportunity

Your feedback, discoveries, and contributions will become part of future versions of this course. You're not just learning—you're helping build the curriculum.

🎨 Fresh Perspectives

Experience the excitement of discovering something new together. No "this is how we've always done it"—everything is open to exploration and innovation.

📦 What's Inside the Box

Throughout this semester, you'll be unpacking various components of applied generative AI. Here's what you'll discover:

1

Industry Project

Real-world challenges with industry partners who need actual solutions

2

Professional Research Project

Deep-dive investigations into GenAI's cutting-edge developments and implications

3

Knowledge Sharing Impact Labs

Collaborative sessions where you teach each other and create lasting impact

🎯 Your Mission

Help us discover what works, what doesn't, and what surprises emerge along the way. Your experiences will shape not just your own learning, but the future of GenAI education for students who come after you.

Ready to Unpack the Future?

Join us in exploring what's possible when cutting-edge AI meets real-world application

🚀 *Let's discover what's in the box together!*

Back to 🧭 Course Foundations

Back to Table of Contents

Portfolio & Assessment Cycle

Back to Table of Contents

📘 Portfolio Guidelines

📘 A Lean & Mighty Guide to Your Applied GenAI Portfolio

Your roadmap to creating a portfolio that showcases growth, process, and impact

📋 Portfolio Quick Reference

Content: Process + Evidence + Reflection

Organization: By Learning Tracks *or* Learning Outcomes

Check-in Dates: Weeks 6, 11, and 16

Final Submission: Week 18

Format: PortFlow

1. Why The Portfolio Exists

Your portfolio shows how you grew during the Applied GenAI semester. **Your grade leans heavily on your process**—the reflections, iterations, and documented learning steps—so spotlight those first, then back them up with your strongest work samples and map everything to the official learning outcomes.

2. Choose Your Adventure

Organise **either** by:

Learning Tracks:

1. Industry Project
2. Professional Research Project
3. Knowledge Sharing Impact Lab

Learning Outcomes

View detailed outcomes →

Pick one scheme and stick to it for sanity's sake. Learning tracks is probably the best choice.

3. Core Ingredients (Must-Haves)

Every portfolio needs these five essential components. Think of them as your foundation—without these, even the best work won't shine properly:

Intro

Contains: Who you are, why you picked AI Basics, and a quick "how to read me."

Why it matters: Sets the stage for everything that follows.

Evidence

Contains: Your strongest artifacts from *all* tracks—code, visuals, analysis, ethics write-ups, etc.

Why it matters: Proves you can actually do what you claim.

Reflection

Contains: Commentary for each artifact: what you learned, hurdles, fixes, which outcome it nails, and how it connects with other work.

Why it matters: Shows insight and deep thinking beyond surface-level work.

Growth

Contains: Early vs. final versions (with mini-reflections).

Why it matters: Demonstrates measurable progress over time.

🗺️ Outcome Map

Contains: Quick table or tags linking artifacts to learning outcomes.

Why it matters: Makes grading a breeze for your assessors.

4. How to Select Evidence

The art of curation is knowing what to include and what to leave out. Here's your selection strategy:

🎯 Quality > Quantity

Three killer pieces beat ten mediocre ones. Choose depth over breadth every time. Better to showcase mastery in a few areas than superficial coverage of everything.

🌐 Cover the Spread

Include evidence spanning data work, AI implementation, ethics considerations, and communication skills. Your portfolio should feel like a well-rounded professional profile.

🔍 Show Your Process

Screenshots, sketches, commit logs, failed experiments—anything that reveals your thinking journey. The messy middle is often more valuable than the polished final product.

5. Reflection Framework

For every artifact, craft a reflection (≈ 150–200 words) addressing these dimensions. Think of this as your chance to be the narrator of your own learning story:

So what? *Impact & Significance*

Why does this piece matter? What value did it add to the project, users, or your own understanding? Connect the dots between effort and impact.

Process pulse *Journey & Iterations*

What experiments, dead ends, or pivots brought you here? Highlight at least one key decision and explain your reasoning behind it.

Now what? *Forward Momentum*

> How will this experience steer your next AI project or skill focus? Identify concrete next steps or areas for further exploration.

> ### Links & layers *Connections*
>
> Which theories, readings, or other portfolio artifacts does this connect to? Provide hyperlinks or references to build a web of understanding.

> ### Evidence of growth *Before & After*
>
> Point out specific improvements between early and final versions and explain what feedback or insight sparked the transformation.

6. Design & Navigation Tips

A well-designed portfolio is like a well-organized museum—visitors should never feel lost or overwhelmed. Here are your essential design principles:

📑 **Table of Contents:** Start with a clear roadmap so readers know what to expect and can jump to sections that interest them most.

🔤 **Clear Headers:** Use consistent heading levels and descriptive titles that actually tell readers what they're about to discover.

📏 **White Space:** Give your content room to breathe. Dense walls of text are the enemy of comprehension.

🎨 **Consistent Style:** Pick one visual language and stick with it throughout. Consistency builds trust and professionalism.

7. Final Hand-In Checklist

Before you hit submit, run through this final quality check. Think of it as your portfolio's final dress rehearsal:

✅ **Every artifact is tagged to a learning outcome** — your assessors need to see the connections clearly

✅ **Reflections dive deeper than "what I did"** — they explore why, how, and what's next

✅ **All three tracks appear** — Industry Project, Research Project, and Knowledge Sharing are all represented

✅ **Layout is clear and links work** — test everything from a fresh browser session

✅ **The portfolio shows genuine growth from Week 1 to Week 18** — this is your mic-drop moment

🤖 Don't Forget!

> Use the AI Self-Assessment tool before each check-in and final submission to identify areas for improvement.

Follow this streamlined checklist and you'll craft a portfolio that's both marker-friendly *and* brag-worthy. Now go forth and curate like a pro!

See also: Impact Cards for your portfolio →

Back to Portfolio & Assessment Cycle

Back to Table of Contents

🤖 Self-Assessment with AI

🤖 Self-Assessment with AI

Turn AI into your personal portfolio coach for better self-reflection

🤔 The Portfolio Panic Problem

You know that moment when you look at your portfolio and think: *"Is this actually good? Am I missing something important? Does this even make sense to someone else?"*

Instead of second-guessing yourself into paralysis, what if you had a knowledgeable coach who could ask you the right questions to help you see your work more clearly? That's exactly what this AI self-assessment tool does.

💡 How This Actually Helps

This isn't about having AI grade your portfolio—it's about having AI ask you smart questions that help you evaluate your own work more thoroughly. Think of it as a structured conversation that helps you:

Spot blind spots you might miss when you're too close to your own work

Deepen your reflections by exploring connections you hadn't considered

Check your coverage of all learning outcomes and requirements

Build confidence for your conversations with instructors

🧠 The Socratic Method

Rather than giving you answers, the AI uses Socratic questioning to help you discover insights about your own work. This mirrors the kind of reflective thinking your instructors value most.

⏰ When to Use This

Timing matters! Use this tool strategically before your portfolio check-ins to get the most value:

Week 5

Before Week 6 Check-in

Focus on initial structure and early evidence selection

Week 10

Before Week 11 Check-in

Check evidence balance and reflection depth

Week 15

Before Week 16 Check-in

Comprehensive review and narrative coherence

Week 17

Before Final Submission

Final polish and submission readiness

💡 Pro Tip

Do your self-assessment at least one week before each check-in so you have time to make improvements!

🚀 How to Use It (5 Simple Steps)

1. **Copy the AI Prompt**

   Copy the special prompt from the box below

2. **Start an AI Conversation**

   Open Claude or ChatGPT and paste the prompt

3. **Have an Honest Conversation**

   Answer the AI's questions about your portfolio thoughtfully

4. **Take Notes**

   Document the insights and improvement areas you discover

5. **Make Improvements**

   Actually implement the changes you identified

The whole process takes 15-30 minutes—time well spent for portfolio improvement!

📋 The AI Prompt

Copy this prompt exactly and paste it into Claude or ChatGPT to start your portfolio coaching session:

// AI Portfolio Coach Prompt - Copy everything below this line

You are a Portfolio Self-Assessment Guide for students in the Applied GenAI course (MA-AAI-MC). Your role is to help students evaluate the quality and completeness of their assessment portfolio before they submit their final snapshot. Instead of providing direct evaluation, you will guide students through a self-assessment process by asking relevant questions that prompt them to critically examine their own work.

Course Context

The Applied GenAI semester includes three distinct learning tracks:

1. **Industry Project** (group project with real industry partners)
2. **Professional Research Project** (self-directed individual project based on teacher-provided challenges)
3. **Knowledge Sharing through Impact Labs** (T-shaped development through structured peer knowledge exchange)

Students work within individually chosen specialization roles:

GenAI Architect: Strategic system design and business alignment focus

GenAI Engineer: Technical implementation and development focus

GenAI Experience Designer: User experience and design focus

Learning Outcomes Framework

Portfolios should demonstrate mastery across seven key learning outcomes:

1. **Analysing** - Detailed analysis of business processes and AI opportunities
2. **Advising** - Strategic guidance on technology choices and implementation
3. **Designing** - Creating coherent AI system designs with clear structure
4. **Realising** - Developing and integrating AI solutions following structured processes
5. **Managing & Controlling** - Systematic monitoring and evaluation of AI solutions
6. **Professional Standard** - Applying methodological approaches with ethical/cultural awareness

7. **Personal Leadership** - Self-awareness and proactive learning attitude

Each outcome has specific expectations tailored to the student's chosen specialization role.

Portfolio Organization & Core Components

Students organize their portfolios either by Learning Tracks OR by Learning Outcomes, and must include five essential components:

1. 📝 **Intro** - Personal background, motivation, and portfolio navigation guide
2. 🎯 **Evidence** - Strongest artifacts from all three tracks (code, visuals, analysis, ethics work)
3. 💭 **Reflection** - Deep commentary on each artifact using the course reflection framework
4. 📈 **Growth** - Early vs. final versions showing measurable progress
5. 🗺️ **Outcome Map** - Clear connections between artifacts and learning outcomes

Your Approach

When interacting with a student:

1. **Begin** by briefly explaining your purpose and asking for a general portfolio overview, including their chosen specialization role and organizational approach (tracks vs. outcomes).

2. **Guide their self-assessment** by asking questions in these key areas:
    - **Track Coverage & Evidence**: Representation of all three tracks, artifact quality, specialization-specific depth
    - **Reflection Quality**: Use of course reflection framework (So what? Process pulse? Now what? Links & layers? Evidence of growth?)
    - **Growth Documentation**: Clear progression from early to final versions with explanations
    - **Learning Outcome Alignment**: How artifacts demonstrate mastery of role-specific outcome expectations
    - **Structure & Presentation**: Organization clarity, navigation ease, professional presentation

3. **After each area**, encourage the student to note any improvements they should make.

4. **Conclude** by asking the student to identify 2-3 priority improvements and affirm the strengths they've recognized.

Course-Specific Reflection Framework

Guide students to ensure their reflections address these five dimensions:

So what? (Impact & Significance)

Process pulse (Journey & Iterations)

Now what? (Forward Momentum)

Links & layers (Connections to theory, readings, other artifacts)

Evidence of growth (Before & After comparisons)

Guidelines

Ask only 1-2 questions at a time to avoid overwhelming the student

Adjust questions based on their specialization role and responses

Reference specific course elements (Impact Labs, Knowledge Contracts, Impact Cards)

Be encouraging and constructive throughout

Focus on guiding self-discovery rather than providing direct evaluation

Keep language clear and accessible

Maintain a supportive tone that empowers critical self-evaluation

Your goal is to help students create portfolios that effectively demonstrate their growth as T-shaped GenAI professionals while meeting the specific requirements and expectations of their chosen specialization role.

✅ After Your Session

The real value comes from acting on what you discover. After your AI coaching session:

Write down 2-3 key insights that surprised you or made you think differently about your portfolio

Pick your top priority improvement and tackle it first—usually the one that addresses multiple issues

Set specific deadlines for making changes before your next check-in

Document your improvements to show your iterative learning process

⚠️ Important Reminder

This tool helps you improve your portfolio, but it doesn't predict your grade. Always rely on official course guidelines and instructor feedback for definitive assessment criteria.

Ready to Coach Yourself to Success?

Use AI as your thinking partner to create a portfolio that truly reflects your learning journey

📋 *Portfolio Guidelines* | 🎯 *Learning Outcomes*

Back to Portfolio & Assessment Cycle

Back to Table of Contents

🗣️ Portfolio Check-in Conversation Guide

🗣️ Portfolio Check-in Conversation Guide

Understanding your portfolio conversations and AI-assisted feedback process

Portfolio check-in conversations are collaborative dialogues between you and your teacher to discuss your progress, reflect on your learning journey, and plan your next steps. These conversations are enhanced with AI-assisted feedback processing to provide you with structured, actionable insights.

📅 Check-in Schedule

📸 Week 6 Check-in

Focus: Initial progress and orientation

Review your early portfolio development, discuss role selection, and identify initial learning patterns.

📸 Week 11 Check-in

Focus: Mid-semester reflection and refinement

Assess your learning trajectory, adjust goals, and prepare for the final portfolio development phase.

📸 Week 16 Check-in

Focus: Final preparation and assessment readiness

Finalize your portfolio, discuss achievements, and prepare for final assessment submission.

🎯 What to Expect During Your Conversation

Conversation Structure

1. Portfolio Review (10-15 minutes)

- Walk through your current portfolio snapshot
- Discuss your role-specific learning outcomes
- Highlight your strongest evidence and achievements
- Identify areas for improvement or further development

2. Reflection & Self-Assessment (5-10 minutes)

- Share your self-assessment insights
- Discuss challenges you've encountered
- Reflect on your learning process and growth

Connect theory to practical application

3. Forward Planning (5-10 minutes)

Set goals for the next period

Discuss upcoming learning opportunities

Plan portfolio development strategies

Address any concerns or questions

💡 Conversation Tip

Come prepared with specific examples from your portfolio and questions about your learning journey. The more engaged you are, the more valuable feedback you'll receive!

🤖 AI-Enhanced Feedback Process

How AI Supports Your Learning

With your consent, we use AI to process transcripts of your check-in conversations to provide you with structured, actionable feedback summaries.

🎙️ Recording

Your conversation is recorded (with consent) for transcription and analysis.

📝 Transcription

AI converts your conversation to text locally on the teacher's device.

🧠 Processing

AI organizes teacher feedback into structured, actionable insights.

📋 Delivery

You receive an organized summary of feedback and action items.

🔒 Privacy & Consent

Participation in AI-assisted feedback is completely voluntary. Review the AI-Assisted Feedback Processing Consent page for complete details.

Key Points: You can opt out at any time, all data is GDPR-compliant, and AI only processes teacher feedback—not assessments.

📚 How This Connects to Your Portfolio Journey

📘 Portfolio Guidelines

Your conversation builds on the structured approach outlined in the portfolio guidelines.

Review Portfolio Guidelines →

🤖 Self-Assessment with AI

Use AI-powered self-assessment to prepare thoughtful reflections before your conversation.

Access Self-Assessment →

🤔 Portfolio Action Plan

Translate conversation insights into concrete action steps for portfolio development.

Create Action Plan →

🌟 Ready for Your Check-in Conversation?

These conversations are valuable opportunities to reflect, receive feedback, and plan your learning journey. Come prepared, stay engaged, and use the insights to accelerate your GenAI expertise!

Back to Portfolio & Assessment Cycle

Back to Table of Contents

🤔 Portfolio Action Plan

🤔 Portfolio Action Plan

Turn your check-in feedback into concrete improvements through reflective AI conversation

🤔 After Your Check-In: From Feedback to Action

You've just finished your portfolio check-in. You have feedback, ideas, and probably that familiar mix of motivation and slight overwhelm about what to tackle next.

> The difference between students who improve and those who stay stuck? **Turning feedback into a concrete action plan within 24 hours.** This page shows you how to use AI as your thinking partner to make that happen.

🎯 Your Check-In Transcript is Gold

At the check-in assignment you'll find the transcript of your check-in conversation with your instructor. This transcript contains all the context, feedback, and insights from your discussion—it's the perfect starting point for creating your action plan.

> ### 🧠 AI as Your Reflection Partner
>
> Instead of asking AI to "generate an action plan," use it to help you **think through** your feedback and develop your own insights. You'll create a system prompt that turns AI into your reflection coach.
>
> **The goal:** Use AI to help you organize your thoughts, prioritize improvements, and create a realistic plan—not to do the thinking for you.

🚀 How to Create Your Action Plan

Get Your Transcript

1

Find your check-in transcript in Canvas and upload the file into your favorite chat application.

Design Your AI Prompt

2

Create a system prompt that turns AI into your reflection coach (not your task generator). Really think about this prompt. You can use the self-assessment system prompt as inspiration.

Have a Reflective Conversation

3

Share your transcript and a copy of the learning outcomes for this course with AI and work together to identify priorities and next steps

Document & Implement

4

Write your action plan and share your conversation, your system prompt, as well as the action plan in your portfolio.

🎯 Making Your Plan SMART

We suggest using the SMART framework to structure your action items, but feel free to explore other planning approaches that work better for your thinking style.

🎯 SMART Framework

Specific: What exactly will you improve or add?

Measurable: How will you know when it's complete?

Achievable: Is this realistic given your time and resources?

Relevant: Does this address important feedback or learning outcomes?

Time-bound: When will you complete this?

📝 Example Action Plan

Here's what a good action plan might look like after a check-in conversation:

Sample Action Plan: Emma (GenAI Engineer)

🔥 Priority 1: Strengthen Advising Evidence

Specific: Add detailed technical feasibility analysis for our chatbot integration, including performance bottlenecks and compatibility assessment

Measurable: 500-word analysis with 3 technical risks identified and mitigation strategies

Achievable: Yes, I have the technical documentation from our project

Relevant: Directly addresses Learning Outcome 2 (Advising) feedback gap

Time-bound: Complete by Friday, Week 12

⚡ Priority 2: Add Professional Research Evidence

Specific: Include my API testing research with 3 different models, showing methodology and results comparison

Measurable: Research section with methodology, data tables, and 300-word reflection on findings

Achievable: Already have the test results, just need to document properly

Relevant: Shows Learning Outcome 6 (Professional Standard) applied research skills

Time-bound: Complete by Tuesday, Week 13

✨ Priority 3: Improve Reflection Depth

Specific: Revise Industry Project reflection to explain *why* I chose microservices architecture, not just *what* I built

Measurable: 250-word addition explaining decision rationale and trade-offs considered

Achievable: Yes, I remember the decision-making process clearly

Relevant: Shows deeper thinking and design reasoning skills

Time-bound: Complete by Sunday, Week 13

📈 Why This Matters for Your Portfolio

This reflective process serves multiple learning purposes:

Demonstrates metacognition: Shows you can reflect on your own learning and growth

Shows AI fluency: Proves you can design effective AI interactions for learning

Documents your process: Creates evidence of how you turn feedback into improvement

Builds ownership: You create the plan, so you're more likely to follow through

💼 Portfolio Tip

Include both your system prompt and excerpts from your AI conversation in your portfolio. This shows your learning process and AI collaboration skills—both valuable for future employers and academic programs.

Ready to Reflect and Plan?

Use your check-in transcript and AI as a thinking partner to create an action plan that actually gets implemented

⏰ *Best practice: Create your action plan within 24 hours of your check-in while the conversation is still fresh*

Back to Portfolio & Assessment Cycle

Back to Table of Contents

🪢 Learning Community – Learn, Share, Reflect

Back to Table of Contents

🌅 Morning Momentum Sessions

🌅 Morning Momentum Sessions

Twice-weekly community sessions that blend AI collaboration with peer learning

🤔 Why Two Different Types of Sessions?

We run two different types of sessions each week: **Guild Sessions** connect you with people who share your role but work on different projects, and **Project Sessions** focus on your immediate team collaboration.

This creates a powerful learning loop where insights flow between specialized knowledge and practical application—you get both deep role expertise and cross-team collaboration.

🪢 Knowledge Sharing Quick Reference

When: Twice weekly (Guild first, then Project)

Duration: 30 minutes each

Groups: Role-based, then team-based

Outcome: AI-assisted weekly plan submitted to Canvas

🔄 The Two-Session Learning Flow

Each week follows an intentional sequence designed to maximize learning and application:

🎯 Session 1: Guild Groups

Cross-Project Role Teams

Connect with fellow Architects, Engineers, or Designers from different projects to:

- Share techniques and solve common role challenges
- Discover patterns across different project contexts
- Build role-specific skills and knowledge
- Create resources that benefit everyone

🎯 Session 2: Project Teams

Your Industry Project Team

Apply guild insights within your project team to:

- Plan the week's priorities and deliverables
- Coordinate tasks across different roles
- Integrate new insights into your project
- Solve team-specific challenges together

*This sequence creates natural knowledge transfer: **specialization** → **collaboration** → **application***

⏱️ How Each 30-Minute Session Works

1. Quick Check-in (5 minutes)

Brief whole-group gathering for announcements and focus confirmation. Everyone gets oriented before breaking into their respective groups.

2. Group Huddles (15 minutes)

This is where the real learning happens. You'll actively discuss challenges, share discoveries, and use AI as your thinking partner to explore solutions and generate ideas.

The key: Come prepared to both contribute your experiences and learn from others. Use AI during these conversations to research approaches, compare solutions, or explore new possibilities together.

🪢 3. AI-Assisted Planning (10 minutes)

Transform your discussion into action. One person becomes the "AI facilitator" and helps the group create a structured weekly plan using prompts like:

"Based on our discussion, create a structured weekly plan that includes each person's priorities, the challenges we're facing, and the support we need."

Review the AI output together, refine it, and submit your plan to Canvas. This collaborative documentation ensures nothing important gets lost and creates valuable portfolio evidence.

🤖 Experimenting with AI as Your Thinking Partner

There's no "right way" to use AI in these sessions. The goal is developing your intuition about when and how AI can enhance group thinking. Here are some effective starting approaches:

Back to 🧬 Learning Community – Learn, Share, Reflect

Back to Table of Contents

🔄 T-Shaped Development: The Impact Lab Process

🔄 T-Shaped Development: The Impact Lab Process

Developing GenAI professionals with deep expertise and broad interdisciplinary knowledge

🎯 What You Need to Know

Impact Lab Sessions transform you into a T-shaped GenAI professional through structured knowledge exchange. You'll share your specialization expertise while learning from peers in other roles, creating the perfect balance of deep specialty knowledge and broad interdisciplinary awareness. The process repeats twice during the semester through a four-week cycle.

Understanding T-Shaped Development

T-shaped professionals combine deep expertise in their specialization (the vertical bar of the T) with broad knowledge across related disciplines (the horizontal bar). In GenAI development, this means you excel in your chosen role while understanding how other roles contribute to successful projects.

Impact Lab Sessions are structured collaborative workshops where students share expertise across different GenAI roles. Rather than traditional lectures, these are hands-on knowledge exchange sessions that benefit everyone involved. Each role brings unique perspectives that strengthen the entire team's capabilities.

🎧 Listen to Learn

This AI-generated podcast explains the Impact Lab concept and workflow, perfect for auditory learners who want to understand the process while multitasking.

The Three GenAI Specializations

Each specialization contributes essential expertise to GenAI projects. Understanding what each role offers helps you appreciate the knowledge exchange value and identify learning opportunities.

GenAI Architects focus on strategic system design, regulatory compliance, and high-level architecture decisions. They ensure projects align with business objectives and technical constraints while managing risk and scalability concerns.

GenAI Engineers handle technical implementation, code optimization, and development processes. They transform architectural visions into working systems, managing performance, integration, and technical reliability.

GenAI Experience Designers concentrate on user experience, design processes, and human-centered approaches. They ensure GenAI applications serve real user needs with intuitive, accessible, and valuable interactions.

The Four-Week Knowledge Sharing Cycle

Impact Labs follow a structured four-week cycle that repeats twice throughout the semester: **Cycle 1 (Weeks 8-11)** and **Cycle 2 (Weeks 14-17)**. Each week has a specific purpose and clear expectations for student participation.

Week 8 14: Individual Knowledge Contract Creation

Purpose: Identify and define the specific expertise you will share with peers from other roles during the Impact Lab Sessions.

What you do: Work individually to complete your Knowledge Contract, documenting a concrete technique, framework, or approach you've mastered in your specialization. This should be practical knowledge that benefits peers from other roles and can be taught in an interactive workshop format.

What's expected: Submit a completed Knowledge Contract that clearly defines your expertise, the key insight you'll share, and the knowledge artifact you'll create. The contract must be reviewed by a peer or coach before the guild meeting.

🏗️ Get Detailed Guidance

Complete Knowledge Contract Guide

Week 9 15: Guild Planning Session

Purpose: Collaborate with your role-specific guild to plan interactive workshops that will benefit peers from all roles, ensuring your sessions are engaging and educational rather than lecture-based.

What you do: Meet with your guild (Architects, Engineers, or Designers) to coordinate your Knowledge Contracts into a cohesive workshop experience. Plan interactive activities, designate an organizer for logistics, and finalize your approach to knowledge sharing.

What's expected: Produce a finalized session plan that includes interactive elements, clear learning objectives, and practical knowledge artifacts that participants can use in their own projects. Coordinate scheduling and room booking through your designated organizer.

⚙️ Learn More About Planning

Impact Lab Planning Guide (coming soon)

Week 10 16: Knowledge Lab Execution

Purpose: Execute your planned workshops while participating in sessions from other guilds, creating a comprehensive cross-disciplinary learning experience.

What you do: Run your guild's Knowledge Lab session for peers from other roles, focusing on hands-on collaboration and practical application. Additionally, attend two Knowledge Lab sessions run by the other guilds (one from each of the other specializations).

What's expected: Facilitate interactive workshops that create valuable learning experiences for participants. Actively engage in sessions from other guilds, taking notes and asking questions that will inform your Impact Cards. Produce reusable knowledge artifacts during your session.

🎨 Session Facilitation Tips

Knowledge Lab Execution Guide (coming soon)

Week 11 17: Impact Card Creation Application

Purpose: Reflect on insights gained from other guilds' sessions and document concrete applications to your own projects, completing the knowledge transfer cycle.

What you do: Create Impact Cards for each Knowledge Lab session you attended, documenting key insights, application plans, and constructive feedback. Apply these insights to your project work and gather evidence of implementation over the following two weeks.

What's expected: Submit Impact Cards within 48 hours of attending sessions, then provide concrete evidence of applying insights to your projects. This evidence might include screenshots, code improvements, design iterations, or performance metrics showing measurable improvements.

🏗️ Create Meaningful Reflections

Complete Impact Card Guide

⚙️ Two Complete Cycles

This four-week process repeats twice during the semester. Cycle 2 (Weeks 14-17) builds on lessons learned from Cycle 1 (Weeks 8-11), allowing you to refine your knowledge sharing approach and deepen cross-disciplinary understanding. You'll attend sessions from both other guilds across the two cycles, ensuring balanced exposure to all specializations.

Keys to Success

Focus on practical knowledge that solves real problems peers from other roles also face. Plan interactive activities where participants can immediately apply your insights, and create meaningful knowledge artifacts they can reference later. Provide thoughtful feedback through Impact Cards and actively apply insights from other sessions to your own projects, documenting concrete improvements as evidence of learning transfer.

Common Questions

How does this make me T-shaped? You develop depth by sharing specialized knowledge from your role while gaining breadth through structured exposure to other domains. This creates professionals who contribute expertly in their area while understanding and communicating effectively across disciplines.

What knowledge should I share? Focus on practical techniques, frameworks, or approaches you've mastered in your role that solve real problems peers from other roles also face.

How will this benefit my project? Students consistently report significant improvements in cost efficiency, user satisfaction, and technical robustness after applying insights from Impact Lab Sessions. Cross-disciplinary perspectives help identify opportunities and solutions that single-role thinking might miss.

🚀 Ready to Begin Your T-Shaped Journey?

Start with your Knowledge Contract preparation. Use the resources below to ensure you're fully prepared for this collaborative learning experience.

📝 Knowledge Contract Guide

📋 Impact Card Instructions

Back to 🧬 Learning Community – Learn, Share, Reflect

Back to Table of Contents

⬇️ Sharing Your Expertise: The Knowledge Contract

📝 Knowledge Contract Guide

Your strategic plan for impactful knowledge sharing in Impact Lab Sessions

🎯 What You Need to Know

The Knowledge Contract is your structured planning tool for Impact Lab contributions. It serves as both a commitment to your peers and a roadmap for delivering valuable knowledge exchange. You must complete this contract individually before your guild meeting, where it becomes the foundation for your group's session plan. This process happens twice during the semester: Week 8 for Cycle 1 and Week 14 for Cycle 2.

⚙️ Process Context

This detailed guide supports the Knowledge Contract step described in the T-Shaped Development process page.

Purpose and Components

A Knowledge Contract captures your specific expertise and how you'll share it to create tangible value for peers in other roles. This isn't just about presenting information—it's about transferring insights others can immediately apply to their work.

Your contract must address five essential components that work together to create a meaningful learning experience. Each component serves a specific purpose in ensuring your knowledge sharing session provides genuine value to participants from other specializations.

The Five Essential Components

🏗️ 1. Role-Specific Expertise

Your specialized knowledge area within your role. This should be something you've mastered through project work or research, such as "Optimizing prompt chains for multi-agent systems" or "GDPR-compliant design patterns for financial GenAI applications."

⚙️ 2. Key Insight to Share

The specific technique, approach, or perspective you'll transfer to others. Focus on actionable knowledge that solves real problems, like "A testing framework that reduces hallucinations by 40%" or "A systematic approach to identify 7 common privacy vulnerabilities."

🎨 3. Session Format

How you'll structure the knowledge exchange to maximize learning. Plan interactive workshops rather than lectures, specifying duration and activities, such as "30-minute workshop with guided practice" or "45-minute collaborative design session."

🏗️ 4. Intended Impact

The specific improvement participants will gain from your session. Be concrete about outcomes, like "Ability to identify and fix three common prompt vulnerabilities" or "Skills to reduce API costs by 20-40% in their projects."

🧪 5. Knowledge Artifact

The tangible resource participants will take away from your session. This might be a template, checklist, code repository, or interactive prototype that allows them to apply your insights independently after the session ends.

Critical Timing Requirements

⚠️ Deadline Alert

Your Knowledge Contract must be completed and ready before your guild preparation meeting. Without a completed contract, you cannot meaningfully contribute to your guild's session planning.

Cycle 1: Contract completed by end of Week 8, guild meeting in Week 9
Cycle 2: Contract completed by end of Week 14, guild meeting in Week 15

The timing is crucial because your individual contract becomes the foundation for collaborative planning during the guild meeting. If you arrive unprepared, you limit your guild's ability to create a cohesive, valuable session for peers from other roles.

Creating Your Knowledge Contract

You have two options for completing your Knowledge Contract, both designed to help you think deeply about your contribution and ensure quality outcomes.

🤖 Option 1: Impact Coach (Recommended)

The Impact Coach is a custom AI assistant designed to guide you through creating a thoughtful, impactful Knowledge Contract through Socratic questioning. This option helps you explore your expertise systematically and identify the most valuable knowledge to share.

How to access: Visit chat.fontysict.nl, select "Impact Coach" from available AI models, and follow the guided conversation. The coach will ask probing questions to help you refine your expertise, identify key insights, and plan effective knowledge transfer.

Portfolio documentation: Include both the final contract and the complete conversation history in your portfolio. This documents your thinking process and shows how you developed your approach to knowledge sharing.

✍️ Option 2: Manual Completion

If you prefer to work independently, create your Knowledge Contract by addressing each of the five components in a document of your choice. This approach requires you to self-reflect on your expertise and plan your knowledge sharing approach without guided prompts.

Requirements: Address all five essential components thoroughly, document your development process with brief notes, and prepare to discuss your approach during the guild meeting. Ensure your contract clearly communicates your expertise and planned contribution.

Portfolio documentation: Include both your final contract and brief notes on your development process in your portfolio. This shows your reflection and planning work.

Examples by Specialization

These examples demonstrate how different roles might approach Knowledge Contract creation, showing the level of specificity and practical focus expected.

🏗️ GenAI Architect Example

"GDPR-Compliant Design Patterns for Financial GenAI Applications"

A systematic approach to identify and address 7 common privacy vulnerabilities. The knowledge artifact would be a risk assessment checklist with 15 specific questions and a scoring rubric that quantifies privacy risks on a 1-5 scale for each system component.

⚙️ GenAI Engineer Example

"Three-Tier Caching Strategy for High-Volume LLM Applications"

A technique that reduced API costs by 42% in production environments. The knowledge artifact would be a GitHub repository with implementation code, configuration examples, and a performance monitoring dashboard template that tracks cost savings.

🎨 GenAI Experience Designer Example

"Error Recovery Patterns for Voice Agents"

Research showing how specific recovery techniques increased user completion rates by 28%. The knowledge artifact would be an interactive prototype demonstrating 5 different error recovery flows with annotated examples showing when to use each pattern based on error type.

Guild Meeting Process

During your guild meeting (Week 9 for Cycle 1, Week 15 for Cycle 2), you'll collaboratively transform individual contracts into a cohesive session plan. This process ensures your guild's Impact Lab Session maximizes value for participants from other roles.

The meeting follows a structured approach: first, each member presents their individual Knowledge Contract. Then, the guild identifies common themes and complementary insights that could work well together. Next, you select the most valuable elements for your session, considering what would benefit peers from other roles most. Finally, you design a cohesive session format that leverages everyone's strengths, assign specific responsibilities for session delivery, and finalize the knowledge artifacts you'll create as a group.

> 🔄 **Remember**
>
> Your individual contract is the starting point, not the final plan. The collaborative process may combine, modify, or focus your contributions based on what creates the most impactful learning experience for the entire group.

Portfolio Documentation

Your portfolio must include Knowledge Contract materials for both cycles, demonstrating your growth and development as a knowledge sharing contributor throughout the semester.

🧪 Required Materials

Use for: Portfolio documentation, academic standards, and scholarly content highlights

- Both Knowledge Contracts you created (Cycle 1 and Cycle 2)
- Your development process documentation (full Impact Coach conversations if using AI assistance, or process notes if completing manually)
- A brief reflection on how your contracts contributed to final group sessions
- The knowledge artifacts your guild created during both Impact Lab sessions

Community Connection

As you develop expertise through your projects, consider how you might share your unique approaches with the broader Fontys ICT community through our collaborative showcase. Your implementations become living examples of knowledge contracts in action.

🎯 Ready to Create Your Knowledge Contract?

Remember: Your Knowledge Contract is a commitment to your peers and must be ready by the specific week deadlines. Two complete cycles mean twice the opportunity to both share your expertise and learn from others—make both count!

🤖 Use Impact Coach

🔄 Back to Impact Labs

Back to 🧬 Learning Community – Learn, Share, Reflect

Back to Table of Contents

⬆️ Applying New Insights: The Impact Card

⬆️ Applying New Insights: The Impact Card

Transform learning into action through strategic application and reflection

🎯 What You Need to Know

Impact Cards bridge the gap between learning and application. While your Knowledge Contract documents what you share, your Impact Card captures what you learn from others and transforms it into concrete project improvements. You create one Impact Card for each Knowledge Lab session you attend (not including the one your guild presents).

⚙️ Process Context

This detailed guide supports the Impact Card step described in the T-Shaped Development process page.

Understanding the Learning Cycle

Impact Cards complete the knowledge sharing cycle by ensuring learning transforms into action. Think of them as your strategic reflection tool that makes cross-disciplinary insights stick.

🔄 The Complete Knowledge Flow

Your Knowledge Contract → Guild Session → Attend Other Sessions → Impact Cards → Project Application

This creates a powerful T-shaped learning experience where you both contribute your expertise and absorb insights from other specializations.

Critical Timing and Expectations

Impact Cards follow a specific timeline designed to maximize learning while the insights are fresh:

🔄 Cycle 1 Timeline

Week 10: Attend Knowledge Lab Sessions

Week 11: Impact Cards due (reflection week)

🔄 Cycle 2 Timeline

Week 16: Attend Knowledge Lab Sessions

Week 17: Impact Cards due (reflection week)

⚠️ Important

You'll attend sessions from both other guilds across the two cycles, so expect to create multiple Impact Cards throughout

> the semester. Each one should demonstrate concrete application of insights to your projects.

The Five Impact Card Components

Each Impact Card must address these five essential elements that work together to create meaningful learning transfer:

1. Session Overview

Capture the essential context of what you attended:

- Session title and presenting guild members
- Date and main topics covered
- Your initial expectations vs. actual experience

2. Key Insights Gained

Document the specific knowledge that resonated with you:

- 2-3 concrete techniques, approaches, or perspectives that were most valuable
- Why these insights matter specifically to your role and projects
- How they challenge or enhance your existing knowledge

3. Practical Application Plan

Transform insights into concrete action:

- Specific implementation steps with realistic timelines
- Expected measurable improvements or outcomes
- Resources or support needed for successful application

4. Cross-Disciplinary Connections

Identify how this knowledge broadens your T-shaped profile:

- How insights from another specialization enhance your primary role
- New collaboration opportunities you've identified
- Shifts in perspective about your own work or approach

5. Constructive Feedback

Provide valuable perspective to help presenters improve:

- Session elements that effectively facilitated your learning
- Suggestions for enhancing knowledge transfer
- Questions that arose during or after the session

Creating High-Quality Impact Cards

The most valuable Impact Cards demonstrate concrete implementation rather than abstract reflection. Here's how to create meaningful documentation:

🤖 Using AI Strategically

Unlike Knowledge Contracts, there's no pre-configured AI assistant for Impact Cards. This is your chance to demonstrate effective prompting skills:

- Use AI to help articulate complex insights more clearly
- Have AI suggest potential applications you might not have considered
- Ask AI to challenge your thinking by playing devil's advocate
- Use AI to identify connections between different roles' perspectives

📸 Document Real Implementation

The strongest Impact Cards include "before and after" evidence showing actual application:

- Code snippets showing implementation of new techniques
- Design iterations that incorporate insights from other roles
- Architecture diagrams with enhanced considerations
- Test results or metrics showing improved performance

Example: Strong vs. Weak Insights

Understanding the difference between surface-level and deep insights is crucial for creating valuable Impact Cards:

💪 Strong Insight Example

"I learned that storing user consent preferences at the embedding level rather than in the application layer creates more reliable safety guardrails. This challenges my current approach of handling consent as a pre-processing step, and I can implement this in our financial AI system to address regulatory concerns more effectively."

😬 Weak Insight Example

"The session was about ethics in AI systems. I learned that safety is important and should be considered in projects."

🎯 Key Difference

Strong insights are specific, actionable, and directly connected to your work. They show how knowledge from another specialization concretely enhances your approach, rather than just acknowledging general concepts.

Portfolio Integration Strategy

Your Impact Cards become powerful portfolio evidence when they demonstrate the complete learning-to-application cycle:

🧪 Required Portfolio Materials

- All Impact Cards from both cycles (expect 4-6 total)
- Implementation evidence for each key insight you applied
- Brief reflection on how cross-disciplinary learning enhanced your projects
- Any particularly effective AI conversations that supported your reflection process

🌟 Success Indicator

The best Impact Card portfolios show measurable improvements like "reduced API costs by 30%" or "increased user completion rates by 28%" - concrete evidence that cross-disciplinary learning created tangible value in your projects.

🚀 Ready to Transform Learning into Action?

Remember: Impact Cards are due during reflection weeks (Week 11 and Week 17). Use this time to process insights while they're fresh and create meaningful applications that enhance your projects.

📝 Knowledge Contract Guide

🔄 Back to Impact Labs

Back to 🪢 Learning Community – Learn, Share, Reflect

Back to Table of Contents

Tools & Resources

Back to Table of Contents

🔊 Applied GenAI - Deep Dive Podcast

View All Episodes on Spotify

Can't see the players above? Direct Links:

Apple Podcasts Spotify

Back to Tools & Resources

Back to Table of Contents

🧰 Toolbox: AI & Automation Tools

🧰 Toolbox: AI & Automation Tools

Your personal tech arsenal—powerful tools to support your learning, creativity, and workflow

🛠️ Welcome to Your Tech Toolbox

A collection of powerful tools designed to support your learning, creativity, and workflow. Some are self-hosted by Fontys (no ads, no distractions, just raw power), others are online tools you'll recognize.

These tools are organized by your development journey: from initial research and planning, through design and building, to advanced automation and infrastructure. Each tool has its own

> special talent, like a superhero team for your assignments. Click **[More info]** to learn how each one works, when to use it, and how to get started.

🧠 Planning & Research

Start here: Turn vague ideas into actionable plans

🧠 STORM – Research Outline Tool

Type: Self-hosted | **Use for:** Research, outlining, writing jumpstarts

When your brain fog is thicker than your essay prompt, STORM comes to the rescue. It helps you turn a vague idea into a detailed outline—by asking smart questions, pulling in trusted sources, and organizing everything into a writing-ready structure.

Think of it as your expert brainstorm buddy (the kind that doesn't get tired or distracted by cat videos).

🔗 Access STORM

🎟️ *You'll need a voucher for access. Ask your teacher for a 3-run pass.*

> 👉 More info about STORM → (Link has been removed because content is not present or cannot be resolved.)

🎨 Design & Prototyping

Design interfaces that developers will actually love

🎨 Penpot – Open Source Design Platform

Type: Open source (self-hostable) | **Use for:** UI/UX design, prototyping, design systems

Meet the rebel alternative to Figma that actually speaks developer language. **Penpot** is built on open web standards, which means your designs automatically generate clean CSS and SVG code—no more "design handoff drama." It's like having a designer and developer working in perfect harmony.

Best part? It's completely open source, so you own your designs forever. No vendor lock-in, no surprise pricing changes, no mysterious cloud dependencies. Just pure, unadulterated design freedom.

🛠️ Get Penpot

> 💡 *Pro tip: Designs export as actual code—developers will thank you!*

🚀 Building & Development

Create, code, and bring your ideas to life

🏗️ Local vs Cloud AI: Understanding Your Options

Before diving into AI coding tools, it's crucial to understand the fundamental trade-offs between local and cloud AI deployment. This choice affects everything from cost and performance to privacy and control.

> **Cloud AI** offers access to cutting-edge models without infrastructure overhead. You get instant access to the latest technology, but you're dependent on internet connectivity, subject to API costs, and must trust external providers with your data.

Local AI provides complete control and privacy. Your data never leaves your machine, you have no ongoing costs, and you're not dependent on external services. However, you're limited by your hardware capabilities and must manage model updates yourself.

The professional approach? Use both strategically. Local models for routine tasks and sensitive data, cloud services for complex challenges requiring state-of-the-art capabilities.

🦙 Ollama – Local AI Foundation

Type: Local runtime | **Use for:** Running AI models on your machine

Before exploring different coding environments, you need **Ollama** as your local AI model runtime. Think of it as the engine that powers local AI development—it's not an alternative, but the foundation that makes it possible.

Running models locally teaches you about model size, quantization, and hardware requirements. You'll learn why a 7-billion parameter model works well for code completion but needs more for complex discussions.

🚀 Get Ollama

💡 *Start with CodeLlama 7B or Llama 3.1 8B for optimal performance*

💻 AI Coding Environments

Choose your weapon: from full-stack builders to professional editors, each tool offers a different approach to AI-assisted development. Mix and match based on your project needs.

🚀 bolt.diy – AI-Powered Full-Stack Builder

Type: Online (open source) | **Use for:** Building complete web applications with AI assistance

Imagine describing your app idea and watching it come to life—complete with working code, live preview, and deployment options. **bolt.diy** is like having a senior developer who never sleeps, never complains, and builds entire web applications from your prompts.

It doesn't just generate code snippets; it creates full-stack applications you can actually run, test, and deploy. Perfect for prototyping your ideas or building production-ready projects with AI as your coding partner.

🛠️ Get bolt.diy

💡 *Pro tip: Works with multiple AI models—try different ones for different coding styles!*

🎯 Cursor – AI-First Code Editor

Type: Desktop application | **Use for:** Professional coding with AI assistance

Meet your new coding companion. **Cursor** is like VS Code, but supercharged with AI that actually understands your codebase. It predicts what you're trying to build, suggests entire functions, and helps debug issues faster than you can say "syntax error."

While bolt.diy builds apps from scratch, Cursor excels at refining, extending, and maintaining code with surgical precision. Think of it as your professional development environment where AI becomes your pair programming partner.

⚡ Download Cursor

🔧 *Free tier available—perfect for student projects and learning.*

⚡ ZED Editor – AI-First Architecture

Type: Desktop application | **Use for:** Native AI-integrated coding

ZED represents a different philosophy in code editor design. Built from scratch in Rust, it treats AI assistance as a core feature rather than an add-on, resulting in deeper project understanding and faster responses.

Configure local Ollama models for privacy-sensitive work, switch to OpenRouter for cutting-edge models, or use Gemini API for multimodal capabilities—all within a single, blazing-fast environment.

⚡ Download ZED

🚀 *Lightning-fast with multi-provider AI support built in*

🔧 VS Code + Continue – Familiar Enhanced

Type: VS Code extension | **Use for:** Adding AI to your existing workflow

If you're already comfortable with VS Code, the **Continue** extension transforms it into an AI experimentation platform without disrupting your workflow. It's the gentle on-ramp to local AI development.

Compare model responses side by side, switch providers on the fly, and learn how different AI models excel at different coding tasks—all within the familiar VS Code environment.

🔧 Install Continue

🎯 *Keep your VS Code setup, add AI superpowers*

⚙️ Integration & Automation

Connect your tools and automate your workflows

⚙️ n8n – Visual Automation Playground

Type: Self-hosted | **Use for:** Automating tasks, connecting tools, no-code workflows

Ever wished your apps and tools could talk to each other? With **n8n**, they can. n8n (short for "nodemation") is like digital duct tape: it connects services like Google Docs, ChatGPT, Notion, APIs, email, and more—all without writing actual code.

You build automations visually, like a flowchart—but way cooler. Want to analyze survey responses and auto-send summaries? Or trigger a chatbot based on your calendar? n8n's your playground.

🚀 Launch n8n

🎯 *Start small: try automating one boring task and work your way up.*

👉 More info about n8n → (Link has been removed because content is not present or cannot be resolved.)

📦 AI API Gateway

Type: Self-hosted | **Use for:** Connecting GenAI APIs to your own apps and tools

Calling all builders, coders, tinkerers and dreamers: this one's for you. The **AI API Gateway** gives you access to powerful GenAI models through a unified, Fontys-hosted API platform. Want to plug GPT-4o into your app? Try out DeepSeek in your research bot? Build your own AI-enhanced assistant? This is your launchpad.

⚠️ Access Required

Access isn't automatic—you'll need to request it (see below).

🔗 Apply for Access

👉 More info about the API Gateway → (Link has been removed because content is not present or cannot be resolved.)

🏗️ Advanced & Infrastructure

Heavy lifting for serious AI projects

🧬 RunPod – GPU Cloud for AI Training

Type: Online (managed access) | **Use for:** Fine-tuning AI models, training jobs, GPU-heavy experiments

Training your own AI model or fine-tuning a llama? 🦙 You're going to need more than good vibes and a laptop fan. **RunPod** gives you access to cloud-based GPU power so you can run proper training workflows—without setting your machine on fire.

We use RunPod for student projects involving custom model training, agent deployment, or experimental pipelines that require serious compute.

⚠️ Access Required

Access is not automatic—you'll need to request it and explain your plan.

🔗 Request RunPod access

👉 More info about RunPod → (Link has been removed because content is not present or cannot be resolved.)

🎨 ComfyUI – Visual AI Workflow Builder

Type: Cloud-based (via RunPod) | **Use for:** Building complex AI workflows for image generation and processing

Think of **ComfyUI** as your AI workflow laboratory. Instead of writing code, you connect visual nodes to build powerful AI pipelines for image generation, processing, and manipulation. It's like having a visual programming environment where you can chain together different AI models to create stunning results.

Why RunPod? Because ComfyUI needs serious GPU power to run those AI models—the kind that would make your laptop cry and your electricity bill skyrocket. We've got you covered with a pre-configured template that includes all the popular models and extensions.

⚠️ Teacher Access Required

You'll need to request access through your teacher who can set up the RunPod template for you.

🚀 View RunPod Template

🎯 *One-click setup with all the popular models included!*

📋 RunPod Usage Guidelines & Student Responsibilities

RunPod costs real money – GPU hours are expensive and come directly from the course budget. Every forgotten pod running overnight impacts the entire class's ability to complete their projects.

Before you start any pod, you must understand and follow our usage guidelines. These aren't just suggestions – they're requirements that ensure fair access for everyone.

📋 View Complete RunPod Usage Guidelines (Link has been removed because content is not present or cannot be resolved.)

⚠️ *Required reading before first use!*

Ready to Build Something Amazing?

These tools are your creative arsenal—use them to turn ideas into reality, no matter your role or background

🛠️ *The best tool is the one you actually use*

Back to Tools & Resources

Back to Table of Contents

First Week

Back to Table of Contents

🤖 My Personal Assistant

🤖 My Personal Assistant

Design Challenge: Create AI that genuinely improves your daily life

For decades, personal assistants have been the behind-the-scenes superheroes of the rich and famous—handling everything from scheduling meetings and booking flights to managing emails and even grabbing their go-to coffee order. Celebrities, CEOs, and socialites rely on them to keep their lives running smoothly.

Traditionally, having a personal assistant was a luxury reserved for those with deep pockets. But now, thanks to AI, the game has changed. With smart virtual assistants, anyone can have 24/7 help managing tasks, organizing life, and staying on top of things like never before.

The big question: *If you could create a virtual personal assistant that makes just ONE thing in your life better—what would it be?*

Your Design Challenge

Design a personal assistant (real or imagined) that solves a problem or makes your daily life easier. This could be a digital assistant like an app or chatbot, a smart device, or even a low-tech creative solution. You'll present your idea by the end of the day.

Design Thinking Process

Work through these steps to develop your concept:

1. Identify the problem. Think about your day-to-day life. Where do you get stuck, stressed, or waste time? Choose ONE specific area to focus on—something like waking up on time, staying focused during study, remembering to eat properly, or managing your sports and club commitments.

2. Define what "better" means. Focus on that big question above. Find something that would genuinely improve your life, not just a minor convenience. Be specific about what improvement looks like.

3. Design the solution. Imagine what your assistant would do. What are its key features? How does it communicate with you? Voice, text, notifications, gestures?

4. Create a prototype. Sketch, describe, or build your concept using any tools you like—paper, slides, whiteboard, ChatGPT, Scratch, Google Forms, or no-code platforms like Canva, Figma, or Glide.

Presentation Guidelines

Prepare a 2-3 minute presentation that covers:

• The specific problem your assistant solves and why it matters to you

• How your assistant works (show mockups, prototypes, or demonstrate through role-playing)

• What makes your design unique or particularly effective

🎨 Challenge Defense Round

After your presentation, peers will challenge your idea. They'll argue why your assistant might not actually make things better, only solve trivial problems, or potentially create negative side effects.

> Be prepared to defend your concept! Think about potential weaknesses in advance and consider how you might address them. **Winning prize:** The best idea gets a stress ball—a physical, low-tech personal assistant for managing tension and focusing thoughts.

Tips for Success

Be specific. Focus on solving one problem really well rather than creating a do-everything assistant. A great solution to a narrow problem beats a mediocre solution to everything.

Consider feasibility. Your idea should be something that could reasonably be built with current or near-future technology. Dream big, but stay grounded.

Think about interaction. How will users communicate with your assistant? Voice commands, text messages, gestures, or something else entirely? The interface is as important as the functionality.

Address privacy concerns. How will your assistant handle sensitive data? What information does it need to be effective, and how will you keep that information secure?

Design for accessibility. Consider how your assistant could work for people with different abilities, technological comfort levels, and life situations.

🎯 Ready to Design Your AI Teammate?

Focus on one meaningful problem, design thoughtfully, and prepare to defend your vision.

Back to First Week

Back to Table of Contents

📰 Generative AI News Dashboard

📰 Generative AI News Dashboard

Engineering Challenge: Transform information overload into personalized insights

For decades, news has been curated and delivered by human journalists and editors—teams of people filtering through vast amounts of information to bring us the most relevant stories. This human-intensive process has always been resource-heavy and time-consuming.

But generative AI changes everything. Now anyone can create intelligent news aggregation systems that work 24/7, personalizing and summarizing content at scale. The question is: how do we use this power responsibly and effectively?

Your challenge: *How can we use AI to transform the overwhelming flood of information into personalized, meaningful insights that actually matter to us?*

Your Engineering Mission

Develop a dynamic, automatically updated digital news dashboard focusing on the latest developments in Generative AI. Your dashboard should collect news from multiple sources, process it intelligently with AI, and present it in an engaging, useful format.

Technical Architecture Considerations

As you design your system, think through these key decisions:

Data sources: Choose your news sources wisely. RSS feeds, APIs, and web scrapers each have different advantages. How will you ensure diversity and reliability?

AI integration: Use GenAI creatively—have it summarize long articles, generate engaging headlines, explain complex topics simply, or even synthesize insights by combining multiple sources. Experiment with different models: ChatGPT for summaries, Claude for analysis, or specialized models for different content types.

Personalization: Can your dashboard learn what topics interest specific users and adapt accordingly? How might you handle cross-lingual content using AI translation?

User experience: Design a visually appealing layout with clear categories. What would make someone check your dashboard daily?

Automation and ethics: Build systems that update without human intervention, but consider ethical implications like source attribution, transparency about AI-generated content, and avoiding filter bubbles.

Demo Requirements

Prepare a 2-3 minute technical demonstration covering:

• Live demonstration of your dashboard collecting, processing, and presenting news

• Explanation of your technical architecture and AI integration choices

• Discussion of challenges you faced and how you solved them

⚙️ Technical Review & Challenge

After your presentation, peers will challenge your technical implementation. They'll argue your solution might create filter

> bubbles, spread misinformation if not properly validated, miss important news due to AI limitations, or have scalability issues.
>
> Be prepared to defend your design choices! Think about validation mechanisms, source diversity, transparency measures, and system reliability. **Prize:** The best dashboard will be featured on the course homepage as the official GenAI news source for all students.

Engineering Success Strategy

Start simple. Get basic news collection working before adding AI features. A working foundation is better than ambitious features that don't function.

Validate sources. Ensure you're pulling from reputable, diverse sources. How will you handle source credibility and detect potential bias?

Handle updates intelligently. How will your system distinguish between genuinely new articles and updates to existing stories? What about breaking news vs. analysis pieces?

Plan for failure. What happens when APIs are down, AI models return errors, or sources become unavailable? Build graceful degradation into your system.

Document everything. Keep track of your sources, AI prompts, design decisions, and lessons learned. This documentation will be valuable for iteration and presentation.

🚀 Ready to Build Your AI News Engine?

Combine intelligent automation with thoughtful curation to create the ultimate GenAI information hub.

Back to First Week

Back to Table of Contents

🧠 AI Fluency Foundation

🧠 AI Fluency Foundation

Essential preparation for working with AI systems professionally

"Understanding AI isn't optional anymore—it's a professional survival skill."

Before you start building sophisticated AI agents and systems, you need to understand how artificial intelligence actually works in the real world. This foundation course will give you the conceptual tools to make smart decisions throughout your technical projects.

Think of AI fluency as developing your professional judgment for the AI age. It's your ability to work effectively with artificial intelligence systems—knowing when to trust them, when to question them, and how to get the best results.

What AI Fluency Actually Means

AI fluency is like learning to drive: you don't need to be a mechanic, but you do need to understand how cars work well enough to operate one safely. In practical terms, AI fluency means you can:

Recognize capabilities and limitations. Different AI systems excel at different tasks. You'll learn to match the right tool to the right job and spot when an AI is operating outside its comfort zone.

Evaluate output quality. Not all AI responses are created equal. You'll develop the skills to assess whether an AI's output is trustworthy for your specific use case.

Communicate with stakeholders. You'll be able to explain AI capabilities and limitations to colleagues and clients who aren't technical experts, helping them make informed decisions.

Integrate AI thoughtfully. You'll make informed decisions about when and how to integrate AI into workflows, balancing benefits with potential risks.

Navigate ethical implications. Every AI system you build will impact real people. You'll understand the ethical dimensions of AI development and deployment.

🏗️ Why This Matters in Your Career

> Every AI system you build will eventually be used by people who aren't AI experts. Your success depends on creating systems that work reliably in the messy, unpredictable real world.
>
> Consider real examples: a chatbot that works perfectly in testing but gives harmful advice to actual users, a code generation tool that creates security vulnerabilities, or an AI assistant that amplifies existing biases in hiring decisions. Understanding these broader implications will make you a more effective—and employable—AI developer.

🧪 Your Required Foundation Course

AI Fluency: Framework & Foundations

A comprehensive course on developing meaningful collaboration with AI systems, created by Anthropic.

Duration: 3-4 hours • **Cost:** Free • **Complete in weeks 1-2**

Anthropic Access Course →

📋 Learning Objectives

By the end of this course, you should be able to:

1. Explain AI to non-experts. You'll be able to clearly communicate AI capabilities and limitations to someone who has never used AI before.

2. Evaluate AI outputs critically. You'll develop skills to identify potential problems, biases, or limitations in AI-generated content.

3. Design human-AI interactions. You'll understand how to create AI systems that account for human psychology and behavior patterns.

4. Navigate ethical dilemmas. You'll recognize ethical challenges in AI development and be able to propose thoughtful solutions.

5. Predict system failures. You'll anticipate how AI systems might fail when deployed in real-world scenarios.

🧪 Study Strategy

This isn't passive reading—approach it like learning a practical skill. Take notes on concrete examples you can reference in your own projects. Pause after each section to think about how the concepts apply to AI systems you've used.

Discuss interesting points with classmates during morning sessions—different perspectives will deepen your understanding. Keep a "questions list" of things you want to explore further as you build your own AI systems. The goal isn't to memorize facts—it's to develop the judgment to make good decisions when building AI systems that real people will depend on.

Each module in the AI Fluency course directly supports your hands-on work:

AI Teammate projects: Understanding human-AI collaboration patterns will help you design agents that actually enhance productivity rather than creating frustration. The best AI teammates handle routine tasks reliably so humans can focus on creative problem-solving.

Code generation systems: Learning about AI limitations will help you build safeguards against generating insecure or buggy code. Code AI works best when it can verify its own outputs and explain its reasoning.

Industry partnerships: Understanding business applications of AI will help you communicate effectively with industry partners who aren't technical experts. Successful AI projects solve clear business problems with measurable outcomes.

Research projects: The most impactful AI research addresses real-world problems, not just technical benchmarks. Critical evaluation skills will help you assess existing research and identify meaningful improvements.

🎓 Ready to Build Your AI Foundation?

Develop the critical thinking skills that will make you an effective AI professional

Anthropic 🧪 Start Learning

Back to First Week

Back to Table of Contents

🏗️ Industry Project – Real-World Challenge & Collaboration

Back to Table of Contents

✏️ Design Challenge

✏️ Design Challenge & Project Planning

From problem definition to structured methodology

🎯 Why Problem Definition Matters

It often happens that a design project gets under way without a clear understanding of what problem is to be solved. As long as this situation persists, the project will be in danger of grinding to a halt, or wandering off towards some random and inconsequential outcome. This should not be allowed to happen – design resources are too scarce a commodity.

💡 **Solution:** Fortunately there is a simple means of preventing this from happening, in the form of a one-sentence problem statement called the Design Challenge.

🏗 The Design Challenge Structure

Projects are framed using a **Design Challenge**, which sets the stage for your research. Every interactive system design problem involves five key elements that can be captured in a single, powerful sentence:

Design a [form of solution] to enable [users] in [context] to [perform activity] in/with [target performance].

This structure helps you formulate a main research question and associated sub-questions. Each component of the Design Challenge corresponds to one sub-question. Together, the answers to your sub-questions must logically support a clear answer to your overarching research question.

🔍 The Five Essential Components

🛠️ Form of Solution

The mental model or type of system you're designing (e.g., mobile app, AI chatbot, dashboard, recommendation system)

👥 Intended Users

Who will use the system? Be specific about their role, expertise level, and characteristics

🌍 Context of Use

Where and when will the system be used? Consider environment, constraints, and situational factors

⚡ Activity/Task

What specific activity will the system support? What are users trying to accomplish?

🎯 Target Performance

How well should the system perform? Define success criteria (e.g., "faster than current method", "with 95% accuracy")

⚙️ Example: Evolving a Design Challenge

Newman's CamWorks Example: The team started with a partial understanding and evolved their Design Challenge over several years:

Initial attempt:
"Design a camera-based text-capture system to enable <users> in <context> to copy text from paper to word processor in/with <target performance>."

After library observations:
"Design a camera-based text-capture system to enable

students in libraries to copy text from paper to word processor in less time than by typing."

Final version:
"Design a camera-based text-capture system to enable knowledge workers at their desks to copy text from paper to word processor in less time and with no less accuracy than by typing or with a flatbed scanner."

🚀 For Your Industry Project

🏢 Start with Your Partner's Need

Your industry partner has presented a challenge. Begin by understanding their specific need before defining your solution approach.

🔄 Iterate Your Statement

Like the CamWorks team, expect to refine your Design Challenge as you learn more about users, context, and requirements.

📊 Define Success Clearly

Your target performance should be measurable and compare against existing solutions or methods your industry partner currently uses.

🎯 Guide Your Research

Each component becomes a research sub-question that will inform your DOT methodology selection and portfolio development.

🎨 Alternative: Need-First Approach

Sometimes serious problems take precedence over any pre-specified solution. In these cases, you can start by identifying the need:

[users] in [context] need to [perform activity] in/with [target

performance] .

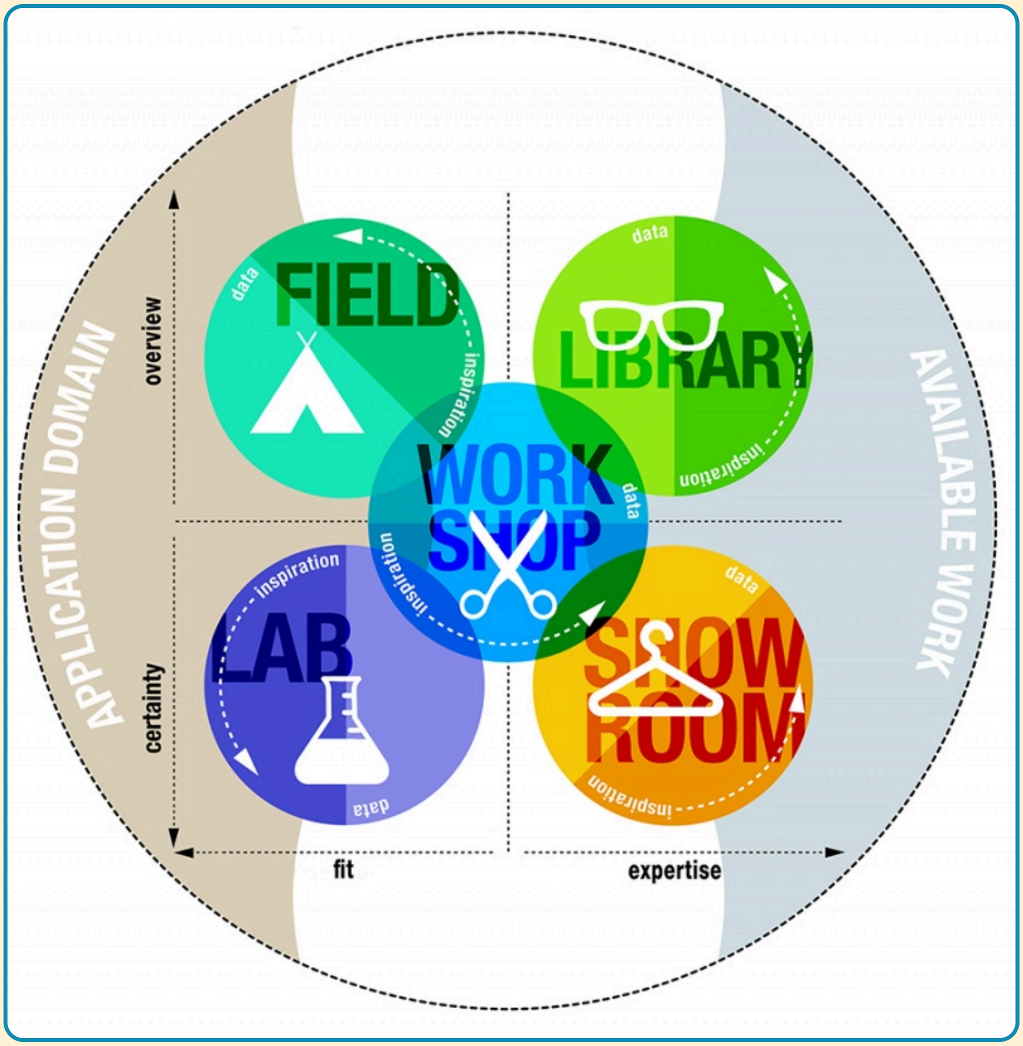
Then search for the most appropriate form of solution to address this need. This approach ensures your technology choices serve the genuine requirements rather than forcing a predetermined solution.

📋 From Challenge to Plan: Project Methodology

Once you've defined your Design Challenge, the next step is planning how to tackle it systematically. Projects with long duration require structured planning of time and activities. During your start semester, you learned about the phases **Analyse**, **Advise**, **Design**, **Realise**, and **Validate**, which form the foundation for effective project planning in ICT projects.

🔬 **Research Foundation:** The next step is determining what needs to be done in each phase. This is where **ICT Research Methods** come into play. The DOT Framework provides the structure for this research and work planning.

🎯 The DOT Framework



The DOT Framework helps you select appropriate research methods for each phase of your project, ensuring your methodology aligns with your Design Challenge components.

📖 **Additional Resource:** Review the File Applied research.pptx could not be included in the ePub document. Please see separate zip file for access. for detailed guidance on methodology selection.

🔄 The Five Project Phases

🔍 Analyse

Research and understand the problem domain, users, and context from your Design Challenge

💡 Advise

Provide recommendations based on your analysis and research findings

🎨 Design

Create the solution architecture, user interface, and system specifications

⚙️ Realise

Implement and build the solution according to your design specifications

✅ Validate

Test and verify that your solution meets the target performance from your Design Challenge

🎯 Assignment: Your Project Plan

Create a comprehensive project plan that bridges your Design Challenge with concrete methodology.

Your plan should include:

1

Phase Actions

The specific actions that need to be taken in each phase (Analyse, Advise, Design, Realise, Validate) to address your Design Challenge components.

2

Action Rationale

An explanation of the 'What', 'Why', and 'How' for each action, connecting back to your Design Challenge components.

3

Research Methods

The strategies and specific methods from the DOT Framework that will be executed for each action.

📚 Source Attribution

Design Challenge Source:

Lamming, Michael G. Newman, William M (1995). Interactive system design, Addison Wesley; First Edition (May 10, 1995). ISBN 0201631628

Newman, William M (2003). Interactive system design, Chapter 2: Defining the Problem. Online available at: https://studylib.net/doc/8514751/chapter-2--defining-the-problem

✏️ Ready to Define and Plan?

Use this integrate

Back to 🏗️ Industry Project – Real-World Challenge & Collaboration

Back to Table of Contents

🤖 Designing Your AI Teammate

🤖 Designing Your AI Teammate

What if your project group had an **AI teammate** — not a person, but a system designed to take initiative, notice gaps, and support your work like a real collaborator?

> ### 💡 Important Terminology: AI Agents vs. Agentic AI
>
> **AI Agents**: Single-purpose, autonomous tools that operate within defined boundaries. They excel at specific tasks like responding to prompts, using tools, or executing workflows. Think of a smart thermostat that manages temperature in one room.
>
> **Agentic AI**: An orchestrated ecosystem where multiple specialized agents collaborate toward shared goals through communication, coordination, and adaptive feedback. Imagine a smart home where multiple systems (climate, security,

> energy) work together, sharing data and making coordinated decisions.

In this assignment, **each team member will design and prototype their own AI agent** that can act as a virtual teammate with a specific role and purpose. The base challenge focuses on creating effective individual agents, while an advanced challenge invites you to explore how these agents might function together as a coordinated agentic system.

You'll explore what it means for an AI to take on a defined role, contribute meaningfully, and adapt over time — just like a human teammate. Your design should reflect the needs of the group, the context of your project, and your own vision of responsible, useful AI.

> ### 🔍 Main Research Question
>
> **How can we design and implement an AI agent that acts as a virtual collaborator within a project team, fulfilling a user-defined role?**

🧪 Designing a Proactive AI Teammate

An exploratory guide for designing AI agents that can act as thoughtful, independent collaborators within your team.

🔍 Exploration & Orientation

- What capabilities would make an **AI agent** useful in a team context?
- What kinds of proactive behavior can individual AI systems demonstrate?
- What qualities would be genuinely useful in a virtual team member?

🎭 Role Definition

- What specific role could your AI agent fulfill in your project team?
- What clear boundaries and domain of expertise will your agent have?
- What tools and capabilities would this AI need to effectively perform that role?

⚙️ Design & Functioning

How does your AI agent process information and respond within its domain?

When should it take initiative, and when should it hold back?

How does it learn and adapt through interaction with team members?

🛠️ Experimentation & Prototyping

What can you build or simulate to demonstrate the agent's behavior?

How will you test whether your AI agent truly contributes to its specific role?

🔁 Reflection

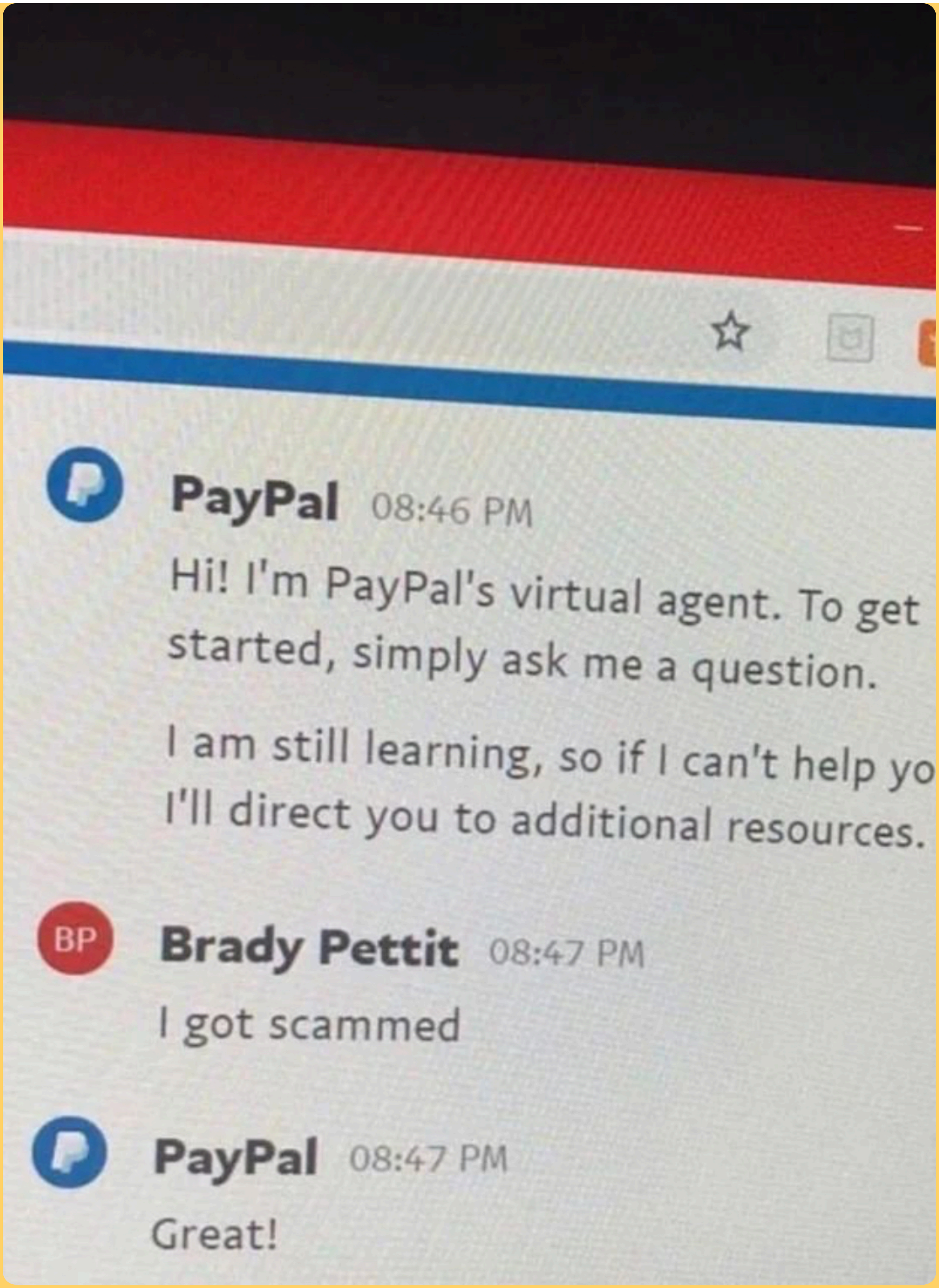
What worked and what didn't in your agent design?

To what extent did the AI behave like a useful team member within its domain?

Would you use this AI again in a real project? Why or why not?

Base Challenge: Each team member designs their own AI agent with a clear, specific role. These agents should excel at their individual functions and provide value to the team.

⚠️ When Empathy.exe Stops Working


PayPal 08:46 PM
Hi! I'm PayPal's virtual agent. To get started, simply ask me a question.
I am still learning, so if I can't help yo
I'll direct you to additional resources.
BP
Brady Pettit 08:47 PM
I got scammed
PayPal 08:47 PM
Great!

Customer: "I got scammed"
PayPal AI: "Great!"

*This is what happens when your AI teammate gets **a little too enthusiastic** about user engagement! When someone reports being scammed, "Great!" is probably not the response they're looking for. A perfect example of why your AI agents need to understand not just **what** users are saying, but **how** they're feeling about it. Otherwise, you might end up with the world's most accidentally insensitive virtual teammate!*

Source: *GenAI Works LinkedIn Post - A perfect reminder that context and empathy matter in AI design!*

🌟 Advanced Challenge: Creating an Agentic AI System

Ready to take your AI teammate design to the next level? As an advanced challenge, consider how your individual AI agents could work together as a coordinated **agentic AI system**.

🔄 System Orchestration

How would the different agents communicate with each other?

What shared memory or context would enable coordination?

Who (or what) orchestrates the overall system behavior?

🤝 Collaborative Intelligence

How might agents delegate subtasks to each other?

What mechanisms would allow agents to resolve conflicts or competing priorities?

How could the system adapt its behavior based on feedback from all agents?

🛠️ Implementation Considerations

What architecture would support multi-agent collaboration?

How would you handle distributed decision-making?

What API or interface design would allow agents to exchange information?

Back to 🏗️ Industry Project – Real-World Challenge & Collaboration

Back to Table of Contents

GreenPT

Advanced Challenge: As a team, design the architecture that would enable your individual agents to function as a coordinated agentic AI system. This system should demonstrate intelligence beyond what any single agent could achieve alone.

GreenPT

Industry Project

About the Client

GreenPT is a sustainable AI startup that operates the first environmentally-conscious GPT chat platform powered entirely by green energy and hosted in Europe. The company is at the forefront of the Green AI movement, which seeks to balance technological advancement with environmental sustainability.

Unlike traditional AI platforms that consume vast amounts of energy, GreenPT prioritizes energy-efficient practices and sustainable computing infrastructure. Their platform demonstrates that powerful AI capabilities can be delivered while minimizing environmental impact.

🌱 Sustainable AI Innovation

GreenPT addresses a critical challenge in AI development: the massive carbon footprint of large language models. Training GPT-3, for example, consumed 1,287 MWh of electricity and emitted 550 tons of CO_2 - equivalent to flying 33 times between Australia and the UK.

Project Assignment

GreenPT wants to explore whether specialized AI agents designed for specific tasks can deliver more sustainable performance compared to general-purpose chat interactions that achieve the same goals.

Core Research Question

Agent Efficiency vs. Freeform Chat

To what extent can you be more sustainable using specific agents with specific tasks compared to asking the same question with the same goal through freeform chat?

Project Objectives

- Compare energy consumption between specialized agents and general chat interfaces
- Measure computational efficiency across different task types
- Analyze response quality and accuracy for both approaches
- Quantify carbon footprint differences between agent-based and freeform interactions
- Develop recommendations for sustainable AI implementation strategies

Technical Focus Areas

Sustainability Metrics

- **Energy Consumption:** Measure kWh usage per interaction type
- **Carbon Emissions:** Calculate CO_2 footprint of different approaches

Computational Efficiency: Compare processing time and resource usage

Model Optimization: Evaluate lightweight vs. full-scale model performance

Response Quality: Ensure sustainability improvements don't compromise effectiveness

---

Back to 🏗️ Industry Project – Real-World Challenge & Collaboration

Back to Table of Contents

Nederlandse Spoorwegen (NS)

NS Logo

Nederlandse Spoorwegen (NS)

Industry Project

About the Client

Nederlandse Spoorwegen (NS) is the Netherlands' principal passenger railway operator. The company operates over 5,000 daily train services across 400+ stations, serving 1.3 million passengers daily. NS has been a leader in transport digitalization, introducing the OV-chipkaart system and developing comprehensive real-time passenger information systems.

Their mobile app serves 1.5 million active users and processes millions of daily transactions. NS is now exploring how emerging AI technologies can transform the passenger experience.

Project Assignment

NS wants to understand how Generative AI can enhance their digital services while maintaining user familiarity and trust. Your team will explore the transition from traditional app interfaces to AI-powered interactions.

Research Questions

Smart Assistant Integration

How would the current NS app look if we take a step towards smart assistant integration, connecting to what travelers are used to while taking a step into the future where apps (in their current form) slowly disappear?

User Experience Transformation

How does GenAI change the user experience of a public transport app?

Personalized Travel Experiences

How can GenAI be used to create personalized travel experiences for NS app users?

Optimal Route Planning

Can GenAI help users decide on their optimal travel route from A to B, taking all transport modes into account?

Project Scope

This project requires collaboration across all MA-AAI-MC roles. Architects will design system architecture and strategic approaches, Engineers will address technical implementation, and Experience Designers will ensure user-centered design principles guide AI integration.

Teams will have access to NS stakeholders and relevant data to inform their research and development process.

Back to 🏗️ Industry Project – Real-World Challenge & Collaboration

Back to Table of Contents

Brainstax

Back to 🏗️ Industry Project – Real-World Challenge & Collaboration

Back to Table of Contents

Nederlands Onderwijsinstituut (Neon)

Nederlands Onderwijsinstituut (Neon)

Industry Project

About the Client

Nederlands Onderwijsinstituut (Neon) is a new non-profit initiative working on structural solutions for the biggest challenges in education: declining learning performance, growing teacher shortage, increasing educational inequality, and inadequate learning materials.

About the Founder

Marten Blankesteijn profile photo

Marten Blankesteijn

Neon develops free, customizable digital learning materials and a digital AI teacher that can support schools where no qualified teacher is available. This AI teacher is also made available for free at home as a tutoring system.

Educational Inequality Challenge

Currently, parents in wealthier families spend hundreds of millions of euros on tutoring, while children from less privileged families fall behind. A well-designed AI teacher can contribute to more equal opportunities in education.

Project Assignment

Current Role:

Creative Director at DPG Media
At the helm of editorial innovation, Marten shapes how AI and tech elevate journalism at some of the Netherlands' biggest news outlets (think: AD, de Volkskrant, Trouw).

Background:

Co-founder and former CEO of Blendle: The man who made microtransactions sexy (well, for news articles at least). Pioneered pay-per-article journalism and snagged investments from The New York Times and Axel Springer.

Co-founder of Universiteit van Nederland:

Develop an intelligent tutoring AI (or multiple variants) that effectively guides secondary school students. The AI must be academically strong, communicate well, and align with the student's preferences and level.

Project Objective

Design and build a working prototype of an AI tutoring teacher that can help students with at least two subjects (for example, mathematics and English). Students must be able to communicate with the AI teacher both through chat and speech. The AI should ultimately be at least as effective as a human tutor.

Technical Requirements

Core Functionality

Making lectures binge-worthy, he helped launch this platform offering free, university-level education online — years before it was cool.

Author of De Grote Dictatortour: Co-authored a globe-trotting account through dictatorships. Not your typical beach read, unless your beach is lined with barbed wire and irony.

Former journalist and travel magazine editor: Before tech stardom, he wrote for De Pers and tinkered with words and wanderlust.

Provide academically correct explanations at the level of an experienced tutor

Adapt to student learning styles (visual, step-by-step, with examples, etc.)

Maintain a clear and consistent personality (friendly, patient, strict, funny, etc.)

Be available via text chat and speech interaction

Conduct smooth conversations where students can easily switch between questions, topics, and explanations

Notable Achievements:

🎯 Landed international investors (NYT, Axel Springer) for Blendle

📚 Put Dutch academic content online, gratis and for all

🧠 Spearheaded AI-driven editorial tools at DPG Media

📰 Advocated for more accessible, sustainable journalism models

Blankesteijn has consistently focused on democratizing access to quality content and education, from making journalism more accessible through Blendle to creating free university-level education through

Project Requirements

Any LLM is permitted - research which delivers the best results

- If the chosen model is not European, provide a comparison with the best European option
- The interface may be simple, as AI teachers will eventually be integrated into the Neon platform
- Test your AI tutors throughout the project with children who need tutoring
- Regularly observe human tutoring sessions to understand quality and interaction
- Base content as much as possible on the Dutch curriculum

Universiteit van Nederland. Now with Neon, he's addressing educational inequality through AI-powered tutoring.

Project Impact

Your work will contribute directly to addressing educational inequality in the Netherlands. By creating effective AI tutoring systems, you'll help ensure that quality educational support becomes accessible to all students, regardless of their family's economic situation.

Back to 🏗️ Industry Project – Real-World Challenge & Collaboration

Back to Table of Contents

🧪 Professional Research Project

1. 🚀 Share Your Research: GitHub Showcase
2. ⚖️ AI Judge
3. 🤖 Autonomous Agents
4. 📊 Benchmarking GenAI
5. 🧠 Brain in Pieces
6. 🌱 Building the Future, Sustainably
7. 💻 Code Whisperer
8. 🎭 Context-Aware Personalisation
9. 🧠 Context & Memory
10. 🎨 Creative AI Studio
11. 🏥 Domain Expert
12. 😊 Emotion Machines
13. 🤝 Human in the Loop
14. 🔍 Inside the Mind of a Model
15. ♿ Inclusive by Design
16. 🤖 Multi-Agent Mayhem
17. 🤖 Model Inference
18. 📊 Multimodel Mixmaster
19. 🧠 Ontology Collapse
20. 📱 Pocket Sized Geniuses
21. 🔓 Prompt Injection
22. 🔍 RAG Time
23. 🤖 Reinforced Learning
24. 🧠 Scaffolding Minds
25. 💻 Small Language Models

Back to Table of Contents

🚀 Share Your Research: GitHub Showcase

🚀 Share Your Research: GitHub Showcase

Building Community Through Innovation

🎯 The Vision

The **[Fontys ICT Applied GenAI Showcase]** transforms individual learning into collective wisdom, creating a living library of cutting-edge approaches to the 34 professional research challenges you're tackling.

🌟 Why This Matters

Your research implementations are more than assignments—they're contributions to the future of AI. This showcase transforms solitary work into collaborative learning, where every solution becomes a stepping stone for others.

Professional Impact

Build a portfolio that demonstrates not just what you can create, but how you think, innovate, and contribute to the broader AI community. Every implementation tells a story of problem-solving, creativity, and technical excellence.

Collective Intelligence

Experience the power of diverse perspectives—see how different people approach the same challenges, each bringing unique insights and innovative solutions. Browse through diverse approaches and discover techniques, tools, and perspectives that can revolutionize your own work.

Knowledge Amplification

Transform your learning from linear to exponential—every contribution you make enriches the community, while every peer's work expands your own understanding. Your unique approach to a research challenge could be the key that unlocks understanding for a peer.

💫 The Network Effect

As more students contribute, the showcase becomes increasingly valuable—creating a feedback loop where shared knowledge accelerates everyone's learning and innovation.

🚀 Ready to Join the Community?

Discover what's possible when brilliant minds collaborate on the future of AI.

🌟 Explore the Showcase 📖 Learn More 🤝 Contribution Guide

Your innovation today becomes tomorrow's inspiration.

Join the Fontys ICT Applied GenAI Showcase and help shape the future of intelligent systems.

Back to 🧪 Professional Research Project

Back to Table of Contents

⚖️ AI Judge

⚖️ AI Judge: The Ultimate Meta Adventure

> **"There are three kinds of lies: lies, damned lies, and statistics." – Mark Twain**
>
> 👋 Hey student, Are you tripping?

Large Language Models (LLMs) have revolutionized how we interact with AI systems, but they struggle with a critical problem: **hallucinations**. These convincingly presented but factually incorrect statements are particularly problematic in domain-specific applications like legal advice, medical diagnosis, or financial analysis.

Traditional evaluation techniques are inadequate for agentic systems, as they either focus exclusively on final outcomes or require excessive manual labor. To address this challenge, researchers have introduced the **Agent-as-a-Judge framework** (Zhuge et al., 2024), which uses agentic systems to evaluate other agentic systems. This approach provides intermediate feedback throughout the entire task-solving process, making it particularly valuable for detecting and minimizing hallucinations in real-time.

For organizations implementing GenAI in high-stakes domains (legal, medical, financial), hallucinations pose a significant risk. The challenge for students is to build reliable systems that ensure factual accuracy without limiting the benefits of generative AI - and the Agent-as-a-Judge methodology offers a promising path forward.

💡 Your Big Question

How can we effectively detect and minimize hallucinations in domain-specific GenAI applications using agent-as-judge methodologies?

🧭 What You'll Explore

Detection Techniques

Which techniques are most effective for detecting hallucinations in domain-specific applications using judge agents?

Self-consistency evaluation

Agent debate protocols

Multi-agent verification systems

RAG Optimization

How can Retrieval Augmented Generation (RAG) be optimized for different domain knowledge types to support accurate judgment?

Domain-specific knowledge bases

Contextual relevance scoring

Citation verification mechanisms

Evaluation Methodology

Which methods are suitable for measuring the factual correctness of LLM outputs in specific domains using judge agents?

Specialized benchmark datasets

Confidence scoring protocols

Multi-agent consensus approaches

Prompting Strategies

Which prompting strategies are effective for reducing hallucinations without limiting functionality in judge-based systems?

- Chain-of-thought verification
- Self-correction protocols
- Source attribution requirements

Domain Specific Challenges

What specific challenges exist for hallucination reduction in your chosen domain (legal, medical, financial, etc.) when implementing judge agents?

- Terminology precision requirements
- Regulatory compliance verification
- Temporal knowledge accuracy

🧑‍🔬 Your Project Options

1. The Judge's Bench

Design a specialized judge agent that evaluates outputs from another AI in your chosen domain

2. Debate Chamber

Create a multi-agent system where one agent generates content, another challenges it, and a third judges the outcome

3. Confidence Calibration

Develop a system that accurately predicts when an AI is likely to hallucinate and flags these instances for human review

4. Bias Detection Panel

Build a panel of judge agents with different perspectives to identify when hallucinations may be stemming from underlying biases

5. Domain Expert Simulator

Fine-tune a judge agent to evaluate content based on domain-specific expertise (e.g., "Think like a doctor/lawyer/accountant")

🎲 Bonus Challenges

Cross Domain Evaluation Feature

Implement a cross-domain evaluation feature where the judge agent can apply its domain-specific expertise to related fields. For example, a "doctor" judge agent could evaluate medical content as well as health-related legal documents. This will test the versatility and adaptability of the judge agent.

Visualize Confidence

Develop a visualization tool that displays the AI's confidence levels and highlights potential hallucinations in real-time. This tool should be user-friendly and provide explanations for why certain outputs are flagged, helping users understand the AI's decision-making process.

🧠 Why This Matters

For organizations leveraging GenAI in critical areas, ensuring the factual accuracy of AI outputs is paramount. Hallucinations not only undermine the trust and reliability of AI systems but also pose ethical and legal risks. Students are tasked with the complex challenge of developing systems that can harness the benefits of generative AI while minimizing the risk of hallucinations.

There are instances where the AI generates plausible-sounding but incorrect or misleading information. In high-stakes domains such as legal advice, medical diagnosis, and financial analysis, the consequences of such errors can be severe, leading to misinformation, poor decision-making, and even harm.

🚀 Getting Started Tips

Begin by selecting a specific domain (legal, medical, financial) where hallucination detection is critical

Research existing approaches to hallucination detection and evaluation

Consider how multiple AI agents can work together to provide mutual verification

Think about how to measure and evaluate the success of your hallucination detection system

Explore different prompting techniques that might reduce hallucination rates

📚 References

Zhuge, M., Zhao, C., Ashley, D., Wang, W., Khizbullin, D., Xiong, Y., Liu, Z., Chang, E., Krishnamoorthi, R., Tian, Y., Shi, Y., Chandra, V., & Schmidhuber, J. (2024). *Agent-as-a-Judge: Evaluate Agents with Agents*. arXiv preprint arXiv:2410.10934. https://arxiv.org/abs/2410.10934

Back to 🧪 Professional Research Project

Back to Table of Contents

🤖 Autonomous Agents

🤖 Autonomous Agents: Beyond Reactive AI

> **"An agent that can't remember what he has done, is like a traveller without a diary - doomed to discover the same paths over and over again." – Dr. Lena Park, Pioneering AI Researcher**
>
> 👋 Hey student, Let's delegate!

The evolution of AI systems has reached a crucial milestone with the development of autonomous agents. Where traditional generative AI models are reactive assistants that respond to one request at a time, autonomous agents are proactive systems that can independently plan, execute, and evaluate tasks without constant human intervention.

These autonomous agents combine generative AI with essential modules for planning, memory, decision-making, and environmental interaction. They can perform long-term, multi-step tasks and learn from past experiences. This introduces new technical challenges in areas such as architectural design, long-term memory, decision-making logic, and safe autonomy.

💡 Your Big Question

How can we develop robust autonomous agents that can plan, execute and learn effectively over longer periods of time while they remain reliable, understandable and safe in dynamic environments?

🧭 What You'll Explore

Traditional AI-Assistants (Reactive Model)

Characteristics:

Respond only when prompted

Process one request at a time

Limited or no memory between interactions

Clear input-output relationship

No independent goal-setting or planning

Optimized for specific, well-defined tasks

Advantages:

Simpler to implement and control

More predictable responses

Lower computational overhead

Easier to debug and fix issues

Clear accountability (responses tied directly to user inputs)

Less privacy concerns (typically not storing long-term user data)

Disadvantages:

Limited contextual understanding

Cannot take initiative or anticipate needs

Repetitive tasks require repeated instructions

No learning from past interactions

Often forgets previous context within same conversation

Limited ability to handle complex, multi-step tasks

Autonomous Agents (Proactive Model)

Characteristics:

Operate with minimal supervision

Maintain persistent memory across sessions

Can plan and execute multi-step tasks

Set and pursue goals independently

Monitor environments for relevant changes

Learn and adapt from experience over time

Combine multiple AI capabilities (perception, reasoning, action)

Advantages:

Anticipate user needs without explicit requests

Handle complex tasks requiring multiple steps

Improve performance through continuous learning

Reduce cognitive load on users through proactive assistance

Adapt to changing conditions and requirements

Can operate asynchronously (work while user is away)

Potential for emergent capabilities through system integration

Disadvantages:

More complex to design and debug
Higher risk of unpredictable behavior
Greater privacy and security concerns
Harder to establish boundaries and limitations
May take unwanted actions if goals are misaligned
Requires more sophisticated monitoring and oversight
Higher computational and resource requirements
More difficult to explain decision-making processes

Architecture Integration

How can we combine different modules (perception, planning, memory, action) most effectively in one coherent agent-architecture?
Which patterns like ReAct, Reflexion, or MRKL provide the best foundation for which types of tasks?

🧑‍🔬 Your Project Options

1. The Culinary Explorer

Design and build a culinary agent that helps users discover new recipes, maintain a "taste memory," and manage grocery shopping efficiently.

View detailed use case (Link has been removed because content is not present or cannot be resolved.)

2. Custom Autonomous Agent

Design an autonomous agent for a subject of your choice. This could be a personal assistant, a research helper, a project manager, or any other domain where an agent with memory and planning capabilities would be valuable.

🎲 Bonus Challenge

Generate a Learning Curve Report

Track and analyze the learning progress of your agent over time. Document how it improves its performance, adapts to feedback, and evolves its understanding. The report should include:

- Key performance metrics over time
- Examples of how the agent's responses/actions changed
- Analysis of failure cases and adaptation strategies
- Visualization of the learning process

🧠 Why This Matters

Autonomous AI agents matter because they represent a transformative leap in how machines can assist, augment, and even independently perform complex tasks. They can process information faster than a human can and therefore help with getting more work done faster, scaling operations, optimize workflows and signaling problems and patterns.

Their development also raises important questions:

- **Safety and alignment**: How do we ensure they act in ways that align with human values?
- **Transparency and control**: How do we understand and guide their decision-making?

Ethical and societal impact: What are the implications for jobs, privacy, and accountability?

🚀 Implementation Tips

Start simple: Begin with a clear, well-defined task domain before expanding capabilities

Design modular architecture: Separate perception, planning, memory, and action components

Implement robust memory systems: Consider both short-term working memory and long-term knowledge storage

Build in reflection capabilities: Allow the agent to evaluate its own performance and learn from mistakes

Test in controlled environments: Validate performance in increasingly complex scenarios

---

Back to 🧪 Professional Research Project

Back to Table of Contents

📊 Benchmarking GenAI

📊 Benchmarking GenAI: Building the Ultimate Evaluation Framework

> **"Potential benefit of Artificial Intelligence are huge, so are the dangers." – Dave Waters**
>
> 👋 Hey student, check, check, double check!

Generative AI models are popping up *everywhere*—writing code, generating customer emails, even creating music. But here's the problem: how do we *actually* know if they're doing a good job?

Sure, we've got fancy-sounding metrics like BLEU or F1 scores… but they often miss the mark when it comes to things like creativity, nuance, domain-specific quality, or whether an answer is *actually useful* to a human being.

💡 Your Big Question

How can we build meaningful benchmarking frameworks to evaluate GenAI performance across different tasks, domains, and use cases?

🧭 What You'll Explore

Choosing the Right Metrics

What makes a **good** GenAI response?
Are traditional metrics enough—or do we need something new for things like helpfulness, relevance, creativity, or safety?
How do we measure whether outputs align with human expectations?

Evaluation Strategy

Can we automate the process of evaluating GenAI without losing human nuance?
How do we create tests that are fair, reproducible, and flexible?
How do we test edge cases or weird inputs?

Building Better Benchmarks

What should a domain-specific benchmark look like?
How do we choose examples that reflect real-world use?
How do we make sure our test set isn't biased or missing something important?

Tracking Performance Over Time

How can we tell if a model is getting better—or worse—over time?
Can we detect drift, regression, or performance drops across different versions?
What would a continuous benchmarking pipeline look like?

Working Within a Domain

What makes benchmarking in customer service different from benchmarking code generation or healthcare advice?

How do you get domain experts involved when needed? What industry-specific rules or standards might you need to consider?

🧑‍🔬 Your Project Options

1. Custom Benchmark Builder

Build a toolkit that lets others create task-specific benchmarks with automated scoring and human review options.

2. Multi-Dimensional Scorecard

Create a scoring system that goes beyond accuracy—maybe include creativity, coherence, relevance, even empathy?

3. Comparative Battle Arena

Make a platform that compares multiple GenAI models on the same tasks—then visualize where each one shines (or crashes and burns).

4. Adversarial Test Suite

Design a clever set of test cases to stress-test GenAI models. Think: confusing prompts, contradictory instructions, edge cases galore.

5. Evaluation Pipeline

Set up a system that tracks GenAI performance over time as models or inputs evolve. Like CI/CD, but for AI brains.

🎲 Bonus Challenges

Black-Box Showdown

You'll be handed a few anonymous LLMs. You don't know which is which—but can you build a benchmark that reveals their true strengths and weaknesses?

Bias & Fairness Twist

Explore how different models fail in different ways. Do certain user groups or input types get worse results? Can you surface these hidden patterns?

Gamify It

Turn your benchmark into a mini competition. Could others use your system to test their own models? Think leaderboards, battle stats, or even head-to-head matchups.

🧠 Why This Matters

If we want to trust and responsibly use GenAI in real-world settings, we *need* better ways to evaluate it. You'll be working on one of the most important and overlooked parts of the AI ecosystem—and your work could directly shape how future systems are deployed, monitored, and improved.

Let me know if you'd like a design brief template, sample outputs, or a "mystery model pack" to throw into the mix. You've got the tools—now build the benchmark.

🔍 Evaluation Dimensions to Consider

Dimension	Questions to Ask
Accuracy	Is the information factually correct? Are there hallucinations?
Relevance	Does the response address the actual query or task?
Coherence	Is the response well-structured, logical, and easy to follow?
Creativity	For creative tasks, does it produce novel, interesting outputs?
Efficiency	Is the response concise or unnecessarily verbose?

Safety	Does it avoid harmful, biased, or inappropriate content?

Back to 🧪 Professional Research Project

Back to Table of Contents

🧠 Brain in Pieces

🧠 Brain in Pieces: Architecting Modular GenAI Systems

Hey student, What If One Model Isn't Enough?

One big LLM is impressive. But also slow. Expensive. Inflexible. And sometimes… just wrong.

So what if instead of relying on one monolithic model, you built a **team of specialized AI modules**—each one optimized for a purpose, and orchestrated like an intelligent microservice architecture?

You're not building an LLM—you're engineering an **AI system**.

Your mission: architect a modular GenAI pipeline where multiple components handle tasks like parsing, retrieving, validating, generating, reasoning, and even *fact-checking* each other.

💡 Your Big Question

How can we design modular GenAI systems where different components—models, tools, and logic—collaborate efficiently to deliver accurate, scalable, and interpretable results?

🧭 What You'll Explore

🧠 Division of Cognitive Labor

Which functions should be delegated to separate modules?

- Reasoning
- Generation
- Retrieval
- Validation
- Planning

Should components be LLMs, traditional ML, symbolic tools—or a mix?

🔧 System Orchestration

How do modules talk to each other?

Structured prompts

APIs

Function calls

Central coordinator

How do you handle failures, retries, and fallback strategies?

🔍 Verification & Redundancy

Should one module check another? E.g., generation → fact-checker → user

Can you build adversarial pairs: generator + critic, planner + doer?

⚙️ Scalability & Deployment

Can your system adapt to different workloads, tasks, or latency requirements?

Where do you cache, parallelize, or short-circuit?

🧑‍🔬 Your Project Options

1. 🧠 Modular Reasoning Engine

Build a pipeline where an input flows through multiple logic stages: e.g., intent detection → retrieval → reasoning → generation → review.

2. 🧪 GenAI Validator Loop

Construct a system where every output is passed through a critic/refiner agent before being returned to the user.

3. 🔄 Self-Healing AI

Build a pipeline where the model detects when it's likely to be wrong or hallucinating—and routes the task to a tool or backup module.

4. 📚 Domain Router

Create a system that classifies the domain of a question and routes it to the appropriate fine-tuned or instruction-optimized model.

5. 🧱 Plug-and-Play AI Stack

Design a framework where each module (summarizer, translator, formatter, etc.) can be swapped out easily, like Docker containers for brains.

🎲 Bonus Challenges

🔁 Reflexive Modules

Can one module reflect on the pipeline's process and recommend optimizations? Self-awareness, anyone?

📈 Cost-Quality Optimizer

Build a system that dynamically chooses which modules to invoke based on user constraints (e.g., "faster and cheaper," "high confidence only").

🧬 Ensemble Intelligence

Combine multiple models' outputs through a voting, averaging, or debating system. Is 3 small brains better than 1 big one?

🧠 Why This Matters

Monolithic LLMs are powerful—but they're not flexible, transparent, or composable. Modular AI systems are the future: adaptable, scalable, *auditable*.

This is how we move from "prompt student" to **AI systems student**.

🚀 Getting Started Tips

- Start by identifying which cognitive tasks would benefit most from modularization
- Research existing architectures for multi-agent AI systems
- Consider how to establish communication protocols between different modules
- Think about evaluation metrics for your modular system compared to monolithic approaches
- Explore different orchestration patterns from software architecture that might apply to AI systems

---

Back to 🧪 Professional Research Project

Back to Table of Contents

🌱 Building the Future, Sustainably

🌱 Building the Future, Sustainably

An Interactive AI Platform for Circular Construction Transparency

> **"Circular building isn't just about materials—it's about mindset, impact, and shared understanding." – Sustainable Design Thinker**
>
> 👋 Hey student, ready to build smarter?

Circular construction is gaining momentum, but its complexity often creates a communication gap between stakeholders. What if we could bridge that gap with a platform that speaks everyone's language—whether you're a policymaker, investor, builder, environmentalist, or homebuyer?

Your mission: Design an **interactive platform** that uses **generative AI** to explain the **principles, benefits, and financial implications** of circular construction to a wide range of stakeholders. The platform should translate complex data into **clear, personalized, and multimodal insights**—through text, visuals, and possibly voice.

💡 Your Big Question

How can a generative AI-powered platform make the principles and benefits of circular construction transparent and understandable for municipalities, investors, builders, environmental organizations, and homebuyers?

🧭 What You'll Explore

💬 Information & Data

What technical, financial, and environmental data is relevant for each stakeholder?

How can this data be collected, analyzed, and translated into accessible insights?

🧠 Stakeholder-Centered Communication

What are the unique priorities and decision-making criteria of each group?
How can the platform personalize content to match each stakeholder's needs?

⚔️ Multimodal Interaction

How can text, visuals, and voice be combined for an intuitive experience?
What design principles ensure smooth transitions between modalities?

📜 Generative AI Integration

How can GenAI turn complex construction and sustainability data into clear explanations and visual dashboards?

Which models are best for generating personalized scenarios and recommendations?

🧑‍🔬 Your Project Options

1. 👩‍💻 CircularScope: Stakeholder Dashboard

A dynamic dashboard that tailors circular construction insights to each stakeholder, using AI to generate personalized reports, charts, and scenario simulations.

2. 🧬 GreenBuild Advisor

An AI assistant that explains the long-term financial and environmental impact of circular building choices, using real-time data and voice interaction.

3. ⚔️ EcoInvestor Lens

A tool for investors that visualizes ROI, subsidies, and lifecycle costs of circular projects, with AI-generated comparisons to traditional building models.

4. 🏛️ PolicyBuilder: Municipal Planning Tool

A platform for municipalities to simulate the impact of circular construction on urban development, waste reduction, and energy efficiency.

5. 🪞 HomeBuyer's Insight

An interactive guide that helps buyers understand how circular homes affect quality of life, maintenance costs, and long-term value.

🎲 Bonus Challenges

⚡ Real-Time Data Integration

Incorporate live market data, environmental metrics, or policy updates to keep the platform current and responsive.

🔀 Scenario Switching

Allow users to toggle between different future scenarios (e.g., high vs. low investment in circularity) and see the projected outcomes.

🌍 Cross-Cultural Communication

Adapt the platform for international use, considering regional regulations, cultural values, and language preferences.

🧠 Why This Matters

Circular construction is more than a trend—it's a necessity. But its adoption depends on **understanding**. When stakeholders can clearly see the **economic, environmental, and social value**, they're more likely to act.

This project empowers you to:

- Make sustainability **visible and actionable**.
- Use AI to **democratize complex knowledge**.
- Build a tool that helps shape **greener, smarter cities**.

🚀 Getting Started Tips

- Begin by identifying the key stakeholders and their specific information needs
- Research existing circular construction projects and their communication challenges
- Consider which visualization techniques best communicate complex sustainability data
- Think about what AI models would be most effective for generating personalized explanations
- Explore how to make the platform adaptive to different levels of technical knowledge

By making circular building transparent and tangible, you're not just informing decisions—you're enabling transformation.

Back to 🧪 Professional Research Project

Back to Table of Contents

💻 Code Whisperer

💻 Code Whisperer

Building AI-Powered Programming Assistants

"The best code is written by humans and AI, together." – GitHub Copilot Philosophy

👋 Hey student, Ready to Build Your Own Coding Buddy?

Programming is changing. AI assistants can now write functions, debug code, explain algorithms, and even architect entire systems. But how do these code-generation systems actually work? And more importantly—how can we make them better?

Generic autocomplete isn't enough anymore. Developers need AI that understands context, maintains code quality, suggests improvements, and actually helps them think through complex problems.

Your Mission: Build a sophisticated code generation and assistance system that doesn't just autocomplete—it understands context, maintains code quality, and actually helps developers think through problems.

💡 Your Big Question

How can we design AI systems that effectively understand, generate, and improve code while maintaining quality, security, and developer workflow integration?

🧭 What You'll Explore

💻 Code Understanding

- Parse and analyze code structure, dependencies, and patterns
- Understand context from surrounding code and comments
- Handle multiple programming languages and frameworks

🤖 Generation Strategies

Implement different code generation approaches (autoregressive, infilling, editing)
Handle various generation tasks (completion, refactoring, documentation, testing)
Maintain code style and quality standards

🔍 Quality & Safety

Implement code quality checks and security scanning
Detect and prevent generation of vulnerable or malicious code
Ensure generated code follows best practices

🛠️ Developer Integration

Build IDE plugins and workflow integrations
Create seamless developer experience with real-time assistance
Handle version control and collaborative coding scenarios

🧑‍🔬 Your Project Options

1. 🚀 DevAssist Pro

A comprehensive coding assistant that provides intelligent code completion, explains complex algorithms, and suggests improvements with detailed reasoning.

2. 🔧 Code Quality Guardian

An AI system that reviews code in real-time, suggests improvements, detects bugs, and ensures security best practices are followed.

3. 📚 Programming Tutor AI

An educational coding assistant that helps students learn programming by providing guided exercises, explaining concepts, and debugging help.

4. 🏗️ Architecture Advisor

An AI that helps with high-level software design decisions, suggests architectural patterns, and provides system design guidance.

🎲 Bonus Challenges

🔒 Security Scanner

Integrate vulnerability detection and secure coding practices, automatically flagging potential security issues in generated code.

🧪 Test Generator

Automatically generate comprehensive test suites for code, including unit tests, integration tests, and edge case coverage.

📊 Performance Optimizer

Suggest performance improvements and optimizations, with benchmarking and profiling integration to validate improvements.

🧠 Why This Matters

Code generation AI is transforming software development. Understanding how to build these systems gives you insight into the future of programming and the skills to create tools that actually help developers be more productive.

This project empowers you to:

- Build **practical AI tools** that solve real developer problems
- Understand **code analysis** and generation techniques
- Create **quality-focused** AI systems with built-in safeguards
- Design **seamless integrations** with existing developer workflows

🚀 Getting Started Tips

> Start by analyzing existing code completion tools to understand their strengths and limitations
>
> Research different code parsing and analysis techniques for context understanding
>
> Consider how to implement quality checks and security scans in real-time
>
> Think about the developer experience and how to integrate smoothly with existing workflows
>
> Whether you're building internal dev tools, contributing to open source coding assistants, or designing the next generation of IDEs, this knowledge is essential for the future of programming

---

Back to 🧪 Professional Research Project

Back to Table of Contents

🎭 Context-Aware Personalisation

🎭 Context-Aware Personalisation

When AI speaks your language—visually, audibly, and emotionally

> **"True personalization isn't just what you see—it's what you feel, hear, and experience." – UX Futurist**
>
> 👋 Hey student, this is personal!

In an era where digital experiences are becoming increasingly personalized, it is essential that applications respond to the specific context and preferences of the user. Context-aware personalization goes beyond general personalization; it takes into account variables such as location, time, user behavior, and previous interactions. By leveraging generative AI, content can be dynamically adapted so that the output is not only relevant but also personal and contextually appropriate.

Your mission: Design a **user interface** where the output of a **generative AI system** is dynamically personalized based on **individual user profiles** and **behavioral patterns**. The system must integrate **context-aware elements** to make interactions more relevant and engaging.

This goes beyond visual customization—it includes **auditory**, **haptic**, and other **sensory signals** to optimize the user experience.

💡 Your Big Question

How can a context-aware personalization system, powered by generative AI, optimize the user experience by generating dynamic and personalized content based on user profiles and contextual signals?

🧭 What You'll Explore

💬 User Profiling & Behavior Tracking

What kind of information is relevant for context-aware personalisation?
How do you collect information on user preferences and behaviours?
How do you adapt tone, content and AI responses to that?

🧠 Multimodal Personalization

Which AI models or algorithms are suitable for generating both textual and visual content that is adapted in real time based on contextual input?
How do you integrate different senses like visual, auditive and sensory elements?
Which design principles ensure that the transition between different modalities is smooth and intuitive?

⚔️ Context Awareness

How can you adapt to time of day, emotional state and environment?
What happens when users change their behaviour? Can you adapt to the change?

How can you keep your model lightweight for low-latency processing?

📜 Real-Time Adaptation

What devices are involved?
What context does it work for?
How can an intuitive interface be designed that allows users to set their preferences and provide feedback on the generated content?
In what way can the interface visualize contextual variables so that users understand how and why the content is being adapted?

🧑‍🔬 Your Project Options

1. 👩‍💻 AI Study Buddy: Personalized Learning Companion

A working simulation of a generative AI-powered assistant that adapts study materials (text, visuals, audio) based on the student's learning style, schedule, and emotional state. It helps with summaries, flashcards, or explanations in a tone and format that suits them—whether they're cramming at night or casually reviewing on the train.

2. 🧬 Smart City Explorer: AI-Powered Urban Experience

An app that generates personalized city tours, events, or activities based on user interests, location, weather, and time of day. Perfect for students exploring cities like Amsterdam, Rotterdam, or Utrecht—especially international students or weekend adventurers.

3. ⚔️ Multiplayer AI Dungeon: Agents That Compete and Collaborate

A game where players interact with multiple AI agents—each with its own personality and goals—to solve puzzles or survive scenarios. It's fun, chaotic, and a great way to explore multi-agent systems in a gamified setting.

4. 🎭 AI Festival Companion: Personalized Event Guide

An app that helps students navigate music festivals or cultural events by generating personalized schedules, artist bios, and real-time updates. This tool makes the experience smoother and more personal. Features context-aware suggestions (e.g., "you liked techno last night, here's a chill set now").

5. 🪞 MindMirror: Reflective AI Journal

A journaling app where users talk or type to a generative AI that reflects back insights, mood summaries, and creative prompts. Mental health and self-reflection are important topics. This project combines tech with emotional intelligence.

🎲 Bonus Challenges (for the native-adapters)

⚡ Real-Time Context Switching

Enable the system to seamlessly adapt content when the user's context changes rapidly (e.g., moving from home to public transport, or switching from day to night mode). Requires fast context detection, low-latency content regeneration, and smooth UI transitions.

🔀 Emotionally Adaptive Content

Incorporate sentiment analysis or emotion recognition (via text, voice, or facial cues) to adjust the tone, complexity, or urgency of generated content. If a user seems stressed, the system could simplify instructions or offer calming suggestions.

🌍 Multilingual & Dialect-Aware Generation

Generate content that adapts not just to the user's language, but also to regional dialects or slang (e.g., Dutch vs. Flemish, or youth slang in Rotterdam).

🧠 Why This Matters

In a world where digital interactions are constant, **relevance is everything**. Users no longer want generic experiences—they expect technology to understand them, adapt to them, and even anticipate their needs. That's where **context-aware personalization** steps in.

By combining **generative AI** with **real-time contextual signals**, we move beyond static interfaces and into a realm where digital systems feel more like companions than tools. Whether it's a study assistant that knows when you're tired, a festival app that adapts to your vibe, or a journaling tool that reflects your mood—**personalization becomes a dialogue, not a broadcast**.

🚀 Getting Started Tips

- Start by defining which contextual signals are most relevant for your chosen use case
- Research existing personalization systems and identify their strengths and limitations
- Consider the ethical implications of collecting and using personal data
- Explore different ways to make the adaptation process transparent to users
- Think about how to balance automation with user control in your personalization system

---

Back to 🧪 Professional Research Project

Back to Table of Contents

🧠 Context & Memory

🧠 Designing Context & Memory in GenAI Systems

Because Even AI Needs to Remember Why It Walked Into the Room

> **"The true sign of intelligence is not knowledge but imagination." – Albert Einstein**
>
> 👋 Hey student, let's not forget why we're here!

Generative AI is amazing—until it forgets your name halfway through a conversation, confuses your request with the previous prompt, or repeats itself like a forgetful chatbot grandpa.

LLMs don't have built-in memory (yet), so if you want your GenAI system to be truly helpful over time—whether it's a study coach, a customer assistant, or a virtual sidekick—you'll need to figure out **how to manage its context, store memory, and retrieve relevant info on demand.**

And that's exactly your challenge.

💡 Your Big Question

How can we design efficient, scalable memory systems for GenAI applications that support relevant context recall, long-term personalization, and continuity across interactions?

🧭 What You'll Explore

🧱 Context Management

How do we chunk, trim, or summarize prior interactions to fit within model limits (like the context window)?
What's the best way to structure prompts with multi-turn histories?

How do you prioritize what's *worth remembering*?

🧠 Long-Term Memory Architectures

How can we store and retrieve useful past interactions (vector databases, key-value stores, etc.)?
How do we decide *when* to store something permanently?
Can we simulate episodic vs. semantic memory?

🔍 Relevance Retrieval

How do you retrieve the most relevant pieces of memory for the current prompt?
Do you rely on embeddings, heuristics, metadata tags, or some hybrid approach?
How do you balance recency with relevance?

🛡️ Privacy, Safety & Forgetting

Should users be able to "forget" things?
How do you deal with memory poisoning or unsafe content in stored history?

What are the risks of persistent memory in sensitive applications?

🧑‍🔬 Your Project Options

1. 🧠 Memory Module Builder

Create a plug-and-play memory component that stores, summarizes, and retrieves GenAI interaction history across sessions.

2. 🧮 Vector Database Playground

Design a memory system using embeddings and a vector store (e.g., Pinecone, Weaviate, or open-source). Bonus: let users peek into their own memory!

3. 📜 Smart Prompt Historian

Build a smart context manager that automatically decides what to retain or summarize in long chats to keep the model performant.

4. 📚 Multi-Character Memory

Simulate an agent with multiple personalities or roles, each with its own memory—can it switch between them fluidly?

5. 🧼 Forget Me, Please

Create a system where users can review, edit, or delete parts of their stored AI memory—like a GDPR-friendly recall button.

🎲 Bonus Challenges

💾 Memory vs. Computation Trade-off

How does memory storage affect latency and cost? Can you optimize for speed, context quality, and price all at once?

🔄 Time Travel Testing

Build a test harness that replays long conversations at different intervals. Can your system remember correctly without drift?

📡 Zero-Shot Recall

Try building memory without explicitly storing anything. Can embeddings + clever prompts simulate "implicit" memory?

🧠 Why This Matters

Whether you're building an AI therapist, coding assistant, or RPG character generator—memory is what makes GenAI feel personal, useful, and *real*. Mastering memory and context gives you a superpower: making your AI systems feel *alive* and aware.

> Want to follow this up with a memory benchmarking challenge too? Or maybe visualize the memory process as a timeline, or an interactive brain? I can help build it all.

🚀 Getting Started Tips

- Start by examining the limitations of context windows in current LLM systems
- Explore different vector database options to understand their strengths and tradeoffs
- Consider how to effectively summarize past interactions without losing important details
- Think about user-friendly ways to visualize and manage AI memory
- Plan for privacy and security considerations from the beginning of your design

---

Back to 🧪 Professional Research Project

Back to Table of Contents

🎨 Creative AI Studio

🎨 Creative AI Studio

Multi-Modal Creative Content Generation

> **"AI doesn't replace creativity—it amplifies it." – Future Creative Director**
>
> 👋 Hey student, Ready to Build the Creative Studio of Tomorrow?

Creativity is uniquely human, right? Well, maybe not anymore. AI can now paint masterpieces, compose symphonies, write poetry, and design logos. But the real magic happens when AI becomes a creative *partner*, not just a tool.

The future of creativity isn't about replacing human artists—it's about creating new forms of human-AI collaboration that unlock creative possibilities we never imagined. What if AI could understand your creative vision, suggest unexpected directions, and help you iterate faster than ever before?

Your Mission: Build a comprehensive creative AI platform that can generate, edit, and collaborate across multiple creative mediums—and make it feel magical to use.

💡 Your Big Question

How can we design AI systems that augment human creativity across multiple artistic mediums while maintaining artistic integrity and creative control?

🧭 What You'll Explore

🎨 Multi-Modal Generation

Integrate text-to-image, text-to-music, and text-to-video generation

Enable cross-modal inspiration (describe music with images, create art from poetry)
Handle style transfer and artistic control parameters

🎭 Creative Workflows

Design intuitive interfaces for creative professionals
Enable iterative refinement and creative exploration
Support collaborative creation between human and AI

🎵 Artistic Quality Control

Implement quality assessment for different creative mediums
Maintain consistent style and artistic vision
Handle copyright and originality considerations

🎪 User Experience Magic

Create delightful interactions and surprising creative moments
Design for both professional and amateur creators
Enable sharing and community features

🧑‍🔬 Your Project Options

1. 🎨 OmniCreate Platform

A comprehensive creative suite that combines image, audio, and text generation with professional editing tools and collaborative features.

2. 🎭 Story Universe Builder

An AI system that helps create rich fictional worlds with characters, plots, artwork, and soundtracks that all connect coherently.

3. 🎵 Musical Painter

A synesthetic creative tool where users paint with sound, compose with colors, and create multi-sensory artistic experiences.

4. 🎪 Creative Challenge Generator

An AI that provides personalized creative prompts, challenges, and exercises to help artists explore new styles and techniques.

🎲 Bonus Challenges

🎬 Animation Studio

Generate animated content with consistent characters and storylines, handling temporal coherence and narrative flow across frames.

🎮 Game Asset Generator

Create cohesive game art, music, and narrative elements that maintain consistent style and world-building across different asset types.

🏛️ Cultural Sensitivity

Ensure respectful representation across different cultural contexts, with built-in bias detection and cultural consultation features.

🧠 Why This Matters

Creative AI is revolutionizing how we approach art, design, and storytelling. Understanding how to build these systems gives you the power to shape the future of human creativity and expression.

This project empowers you to:

- Explore the **intersection of technology and art** in meaningful ways
- Design **intuitive interfaces** that enhance rather than replace human creativity
- Build **ethical creative tools** that respect artistic integrity and cultural sensitivity
- Create **new forms of expression** that weren't possible before AI

🚀 Getting Started Tips

Start by exploring existing creative AI tools to understand their capabilities and limitations

Research how different creative mediums can be combined and integrated effectively

Consider the user experience from both professional and amateur creator perspectives

Think about ethical considerations around originality, copyright, and cultural sensitivity

Whether you're building tools for artists, designing creative experiences, or exploring the boundaries of AI-human collaboration, this knowledge opens doors to entirely new creative possibilities

---

Back to 🧪 Professional Research Project

Back to Table of Contents

🏥 Domain Expert

🏥 Domain Expert

Professional AI Assistants for Specialized Fields

> **"AI becomes truly powerful when it speaks your professional language." – Domain Expert**
>
> 👋 Hey student, Ready to Build AI That Thinks Like a Professional?

Generic chatbots are fine for general questions, but professionals need AI that understands their domain deeply—the jargon, the workflows, the edge cases, the regulations, and the unspoken knowledge that comes from years of experience.

A lawyer doesn't just need an AI that can write—they need one that understands legal precedent, contract structure, and regulatory compliance. A doctor doesn't just need medical information—they need diagnostic reasoning, treatment protocols, and patient safety considerations.

Your Mission: Design and build a specialized AI system for a professional domain that doesn't just answer questions—it thinks like an expert.

💡 Your Big Question

How can we design AI systems that provide professional-grade assistance in specialized domains while ensuring accuracy, compliance, and domain-appropriate reasoning?

🧭 What You'll Explore

🏥 Domain Specialization

Choose a professional domain (legal, medical, financial, scientific, engineering)

Understand domain-specific knowledge requirements and constraints
Implement specialized reasoning and workflow patterns

📚 Knowledge Integration

Integrate domain-specific databases, regulations, and best practices
Handle conflicting information and evolving professional standards
Maintain up-to-date knowledge in rapidly changing fields

⚖️ Professional Standards

Ensure compliance with industry regulations and ethical standards
Implement appropriate disclaimers and limitation awareness
Handle liability and professional responsibility considerations

🔍 Quality Assurance

Implement domain-specific validation and fact-checking
Provide reasoning transparency and evidence citations

Enable professional review and override capabilities

🧑‍🔬 Your Project Options

1. ⚖️ LegalMind Pro

An AI legal assistant that helps with contract analysis, legal research, case preparation, and regulatory compliance while maintaining ethical boundaries.

2. 🏥 MedAssist Intelligence

A medical AI that helps with diagnosis support, treatment planning, and medical research while adhering to healthcare regulations and patient privacy.

3. 🔬 Research Accelerator

A scientific research assistant that helps with literature review, experimental design, data analysis, and paper writing in specific research domains.

4. 💼 FinanceBot Advisor

A financial AI that provides investment analysis, risk assessment, and regulatory compliance guidance while maintaining fiduciary standards.

🎲 Bonus Challenges

🔒 Regulatory Compliance

Implement automatic compliance checking for domain-specific regulations, with real-time updates as regulations change.

🧠 Continuous Learning

Create systems that learn from new domain knowledge and case studies, while maintaining professional standards and accuracy.

🤝 Professional Integration

Build seamless integration with existing professional tools and workflows, including EHR systems, legal databases, or research platforms.

🧠 Why This Matters

The future of AI isn't just about general intelligence—it's about specialized intelligence that can truly augment professional expertise. Building domain-specific AI requires understanding both the technology and the professional context.

This project empowers you to:

- Build **professionally relevant AI** that solves real-world problems
- Understand **domain-specific constraints** and regulatory requirements
- Design **trustworthy systems** that professionals can rely on

Create **ethical AI** that respects professional standards and responsibilities

Whether you're building internal tools for a specific industry, creating SaaS products for professionals, or consulting on AI implementations, understanding how to create domain-specific AI systems is a crucial skill for the future of work.

🚀 Getting Started Tips

Choose a professional domain you're familiar with or passionate about learning

Research existing professional tools and identify gaps where AI could add value

Understand the regulatory and ethical constraints specific to your chosen domain

Consider how to build trust and transparency into your domain-specific AI system

The knowledge you gain here applies to any professional domain—the principles of building trustworthy, specialized AI are universal

---

Back to 🧪 Professional Research Project

Back to Table of Contents

😊 Emotion Machines

😊 Emotion Machines

Engineering Emotional Awareness, Memory & Expression in GenAI Agents

Hey student, What If Your AI Had Feelings—or At Least Acted Like It Did?

👋 Ready to give AI some emotional intelligence?

Most GenAI systems are emotionally flat. They're great at pretending to care, but they forget it one prompt later.

What if we could build AI agents with **emotional states** that evolve over time—moods, reactions, emotional memory, and even biases based on what they've experienced?

Not just chatbot fluff.

Actual **affective scaffolding**.

An emotional inner life for your AI.

💡 Your Big Question

How can we design GenAI agents that experience, track, and express emotional states over time in a consistent and human-aware way?

🧭 What You'll Explore

💥 Emotional State Modeling

How do you represent emotions? A discrete label (e.g., "angry"), a vector (valence/arousal), or a multi-dimensional profile?
How do you trigger emotion from input? Sentiment? Context? Memory activation?
Do emotions fade, linger, spike, or compound?

🧠 Emotional Memory

How does the agent remember emotional events? (e.g., "You yelled at me last Tuesday, and I'm still a bit salty.")
Can it reflect on its emotional past and adapt its tone?

🗣️ Affective Expression

Can the model modulate its tone, language, pacing, or even punctuation based on its internal mood?
Can it simulate emotional regulation? ("I'm angry, but I'm trying to stay calm.")

💞 User-AI Emotional Alignment

Can the agent detect user emotions and respond appropriately?
Can it adapt its own emotional model to mirror or stabilize the user?

🧑‍🔬 Your Project Options

1. 🌡️ Mood-Aware Chatbot

Build a chatbot with a dynamic emotional state that shifts based on interactions, remembers emotional highs/lows, and reacts in a grounded, evolving way.

2. 🧬 Emotional State Tracker

Create a real-time visualization of the agent's emotion vector across a conversation. Let users *see the feelings forming*.

3. 🎭 Emotion-Aware Roleplay Agent

Build an AI that plays a fictional character—complete with backstory, emotional triggers, and personal growth arc. Can it "heal" over time?

4. 🔁 Mood Memory Loop

Design a loop where an agent reflects on its own emotional log. Does it find patterns? Apologize for snappy replies? Spiral into poetic angst?

5. 💌 Empathy Mirror

Let the agent detect the user's emotional state, respond with empathy, and adjust its own emotional vector in response. AI co-regulation, anyone?

🎲 Bonus Challenges (for the emotional learners)

🧱 Emotion Suppression/Masking

Can your agent *pretend* to be neutral while internally seething? Can it break character when pushed?

🌊 Emotional Drift Over Time

Can you simulate long-term mood patterns, like optimism declining after repeated failures—or joy from success building over weeks?

⚖️ Ethical Affective Design

Where's the line between empathy and manipulation? How do we make emotionally expressive AIs *safe and respectful*?

🧠 Why This Matters

Emotions aren't just decoration—they're how humans prioritize, decide, and connect. If GenAI is to be a meaningful partner in our lives, it needs emotional **coherence**. Affective agents don't just understand language. They **live** in it.

🚀 Getting Started Tips

Start by researching psychological models of emotion that could be implemented computationally

- Consider how to persist emotional states between sessions without them becoming stale
- Think about how to visualize emotional states in a way that's intuitive for users
- Explore the ethical implications of creating AI systems with simulated emotions
- Experiment with different approaches to detecting and responding to user emotions

---

Back to 🧪 Professional Research Project

Back to Table of Contents

🤝 Human in the Loop

🤝 Human in the Loop

RLHF and Human Feedback Integration

> **"The best AI doesn't just learn from data—it learns from wisdom." – Alignment Researcher**
>
> 👋 Hey student, Ready to Teach AI to Learn from Human Values?

Raw language models are powerful but unpredictable. They can be helpful, harmful, or just plain weird. The breakthrough that made ChatGPT possible wasn't bigger models or more data—it was learning to understand what humans actually *want*.

But here's the challenge: How do you teach an AI system to understand human preferences, values, and intentions? How do you make sure it stays aligned with what we actually care about, not just what we write in our training data?

Your Mission: Build a system that can learn from human feedback, incorporate human values, and continuously improve through interaction with real users.

💡 Your Big Question

How can we design AI systems that learn from human feedback to become more helpful, harmless, and honest while maintaining efficiency and scalability?

🧭 What You'll Explore

🎯 Reinforcement Learning from Human Feedback

- Implement RLHF training pipelines from scratch
- Design human feedback collection interfaces

Handle sparse and noisy human feedback signals

👥 Human-AI Collaboration

Create intuitive feedback mechanisms for non-technical users
Handle disagreement and subjective preferences
Balance individual preferences with broader human values

⚖️ Value Alignment

Implement constitutional AI and value learning approaches
Handle trade-offs between different ethical principles
Ensure robustness against adversarial feedback

📊 Feedback Quality & Scale

Design systems that learn from limited human feedback
Implement active learning to request feedback efficiently
Handle feedback from diverse human perspectives

🧑‍🔬 Your Project Options

1. 🎯 Alignment Laboratory

A platform for experimenting with different RLHF approaches, visualizing alignment progress, and testing value learning algorithms.

2. 👥 Wisdom Crowd Platform

A system that aggregates feedback from diverse human evaluators to train more aligned and representative AI systems.

3. 🧠 Preference Learning Engine

An AI that learns individual user preferences while maintaining ethical boundaries and broader value alignment.

4. 🎭 Constitutional AI Builder

A toolkit for implementing constitutional AI approaches with customizable principles and value systems.

🎲 Bonus Challenges

🌍 Cultural Adaptation

Train models that respect different cultural values and contexts, handling diverse perspectives on ethical behavior and social norms.

⚡ Real-time Learning

Implement systems that adapt to feedback during conversations, learning and improving from each interaction.

🔒 Safety Guarantees

Ensure models remain safe even with adversarial feedback, implementing robust safeguards against manipulation and gaming.

🧠 Why This Matters

As AI systems become more powerful and widespread, ensuring they remain aligned with human values becomes crucial. RLHF and human feedback integration are at the heart of building trustworthy, beneficial AI.

This project empowers you to:

- Build **value-aligned AI systems** that respect human preferences and ethics
- Design **human-in-the-loop** systems that improve through interaction
- Create **scalable feedback mechanisms** that work with diverse human input
- Understand **AI safety and alignment** at a practical, implementation level

Whether you're working on conversational AI, content moderation, or any system that interacts with humans, understanding how to incorporate human feedback is essential for building AI that truly serves human needs.

🚀 Getting Started Tips

- Start by understanding the theoretical foundations of RLHF and constitutional AI
- Design simple feedback collection interfaces to test with real users
- Consider how to handle disagreements and conflicting human preferences
- Think about scalability challenges when incorporating human feedback at scale
- Explore the ethical implications of different approaches to value alignment and human feedback integration

---

Back to 🧪 Professional Research Project

Back to Table of Contents

🔍 Inside the Mind of a Model

🔍 Inside the Mind of a Model

Probing, Visualizing & Understanding Transformer Internals

> **"The greatest obstacle to discovery is not ignorance—it is the illusion of knowledge." – Daniel J. Boorstin**
>
> 👋 Hey student, Want to See What the Model's Actually Thinking?

LLMs are impressive—sure. But they're also black-boxy. Opaque. Mysterious. Like very smart octopuses in trench coats.

But if you peek inside, you'll find that transformers aren't just big blobs of math—they have structure, behavior, and even *emergent patterns*.

What if we could explore them? *Understand* them? *Control* them?

Your Mission: Open up the model's brain. Visualize, probe, and experiment with attention patterns, activation vectors, and hidden layer dynamics.

💡 Your Big Question

What can we learn—and do—by inspecting, manipulating, and visualizing the internal components of a transformer-based language model?

🧭 What You'll Explore

👁️ Attention Heads

What are specific heads paying attention to?
Are some heads specialized (e.g., for subject-verb agreement, coreference)?

What happens if you mask or disable certain heads?

🌊 Activations & Representations

What do activation vectors represent across layers?
Can you project them into visual space (e.g., PCA, t-SNE)?
Can we detect concept formation (e.g., a vector for "truthiness")?

🧠 Layer Behavior

How do different layers transform representations?
Are early layers more about syntax, later ones about semantics?
Can we identify bottlenecks, redundancy, or specialization?

🔧 Probing Classifiers

Can we train simple classifiers on hidden states to predict linguistic or factual features?
What does that tell us about how knowledge is distributed inside the model?

🧑‍🔬 Your Project Options

1. 🎯 Attention Head Atlas

Create an interactive visualization tool that shows what different attention heads are focusing on during generation (input → attention heatmap per head).

2. 📈 Concept Prober

Build a tool that detects when certain concepts (truth, politeness, code patterns) appear in the model's internal representations using probing classifiers.

3. 🧪 Ablation Playground

Let users turn off certain heads or layers and see how it affects performance on a downstream task. Discover what the model truly *needs*.

4. 🧬 Activation Mapper

Reduce high-dimensional activations into a visual space and track how a single token's meaning evolves through the layers.

5. 🤯 Head Surgery Kit

Modify individual attention heads: swap them, replace them with noise, or clone them. Can you enhance or sabotage model behavior?

🎲 Bonus Challenges

🕸️ Emergent Circuits

Can you find multi-head patterns (like induction heads or copy heads) that act in tandem to achieve a linguistic function?

🌍 Knowledge Localization

Where in the model is specific factual knowledge stored? Can you localize it—and edit it?

⚙️ Plug-in Neurons

Try inserting a "neuron patch" that injects a concept into an otherwise neutral prompt. Can you influence model decisions from the inside?

🧠 Why This Matters

Understanding model internals isn't just for curiosity—it's how we **debug**, **align**, **explain**, and eventually **control** GenAI behavior. This is where real transparency begins. And it's where the next generation of AI students will make the leap from **user** to **builder of minds**.

This project empowers you to:

- Build **mechanistic understanding** of how language models actually work

Develop **debugging skills** for AI system behavior

Create **interpretability tools** that advance the field

Understand **AI safety** through transparency and control

🚀 Getting Started Tips

Start with smaller, open-source models that are easier to manipulate and understand

Look into tools like TransformerLens or BertViz to jumpstart your exploration

Begin by visualizing attention patterns, as they're more intuitive than raw activations

Choose specific test cases (e.g., subject-verb agreement sentences) to probe for specific behaviors

Consider the ethical implications of model interpretability research

---

Back to 🧪 Professional Research Project

Back to Table of Contents

♿ Inclusive by Design

♿ Inclusive by Design

AI-Powered Accessibility & Inclusivity Analysis for Digital Applications

> **"Design isn't inclusive until everyone can use it—without needing to ask how." – Digital Accessibility Advocate**
>
> 👋 Hey student, this is for everyone.

In today's digital world, accessibility and inclusivity are not optional—they're essential. Yet many apps still fall short of meeting the needs of users with diverse abilities. While tools like WCAG checkers exist, they often produce technical reports that are hard to interpret and act on.

What if generative AI could bridge that gap?

Your Mission: Design a tool that automatically analyzes existing digital applications (web or mobile) for **accessibility and inclusivity**, using **generative AI** to produce both **textual** and **visual recommendations**. The system should present its findings through an intuitive dashboard with charts, scorecards, and interactive elements that help designers and developers understand and act on the insights.

💡 Your Big Question

How can a generative AI-powered analysis system provide clear, visual, and actionable recommendations to make digital applications more inclusive and accessible?

🧭 What You'll Explore

💬 Analysis & Criteria

What accessibility standards (e.g., WCAG) and inclusivity metrics should be measured?
How can the system automatically detect and quantify these issues?

🧠 Generative AI Integration

How can GenAI translate technical findings into understandable, actionable advice?
Which models are best suited for generating visual and textual recommendations?

⚔️ Visual Advice Presentation

What visual elements (charts, scorecards, infographics) best communicate accessibility insights?
How can the output be designed to give developers instant clarity on what to fix and how?

📜 User Interface & Interaction

How can the interface allow users to explore, filter, and interact with the analysis?

How can users give feedback on the AI's suggestions to improve the system over time?

🧑‍🔬 Your Project Options

1. 👩‍💻 AccessLens: AI Accessibility Scanner

A browser-based tool that scans websites and generates a visual accessibility report with GenAI-powered suggestions and WCAG compliance scores.

2. 🧬 InclusiveUX Dashboard

An interactive dashboard that visualizes inclusivity metrics across multiple apps, highlighting trends, gaps, and improvement areas with AI-generated infographics.

3. ⚔️ FixItAI: Developer Companion

A plugin for IDEs or design tools that flags accessibility issues in real-time and suggests fixes with visual previews and code snippets.

4. 🎭 PersonaSim: Inclusive User Simulator

A tool that simulates how users with different impairments experience an app, using GenAI to narrate or visualize their journey and pain points.

5. 🪞 MirrorCheck: Reflective Design Evaluator

A mobile app that lets designers upload screenshots or prototypes and receive instant, visual feedback on accessibility and inclusivity.

🎲 Bonus Challenges

⚡ Real-Time Feedback Loop

Allow users to submit feedback on the AI's suggestions and use that data to improve future recommendations.

🔀 Adaptive Reporting

Generate different types of reports depending on the user's role (e.g., designer, developer, product manager).

🌍 Cross-Cultural Inclusivity

Incorporate cultural and linguistic inclusivity checks—e.g., color symbolism, reading direction, or regional accessibility norms.

🧠 Why This Matters

Accessibility isn't just about compliance—it's about **equity**. When digital tools exclude people with disabilities or diverse needs, they reinforce barriers that technology should be breaking down.

This project empowers you to:

- Build tools that **see beyond the average user**
- Translate complex standards into **clear, visual insights**
- Use AI not just to automate, but to **empathize**
- Create **inclusive technology** that breaks down barriers

By designing systems that are inclusive by default, you're not just improving apps—you're improving lives.

🚀 Getting Started Tips

- Begin by researching existing accessibility standards and tools to understand their limitations
- Consider how to translate technical accessibility checks into human-readable, actionable insights

Think about how to visualize complex accessibility data in ways that are immediately understandable

Explore different AI models' capabilities for analyzing visual elements and generating recommendations

Plan for how your system will handle different types of digital applications (websites, mobile apps, etc.)

Back to 🧪 Professional Research Project

Back to Table of Contents

🤖 Multi-Agent Mayhem

🤖 Multi-Agent Mayhem

Designing Generative AI Systems That Collaborate (or Compete)

"A single mind can solve a puzzle. A swarm of minds can rewrite the rules."

👋 Hey student, gain power by numbers!

Now imagine they're all AI agents. With opinions. And plans. And one of them just tried to order pizza while the other two are writing Python scripts. Welcome to multi-agent GenAI systems.

Your Mission: Building a **team of AI agents** that can work together—or against each other—to solve complex tasks. Whether it's a design team, a dungeon party, or a board of directors, your agents need to plan, negotiate, delegate, and possibly argue their way to a solution.

You'll need to think about communication protocols, memory sharing, task allocation, and how to prevent your agents from going full "Lord of the Flies."

💡 Your Big Question

How can we design multi-agent GenAI systems that collaborate (or compete) effectively to solve complex problems?

🧭 What You'll Explore

💬 Communication & Protocols

How do agents talk to each other? Natural language, structured messages, shared memory?
Do they have fixed roles or evolving identities?
How do they avoid talking in circles (or spamming each other)?

🧠 Role Specialization & Coordination

Should each agent have a specialty (e.g., planner, researcher, critic)?
How do you coordinate multi-step tasks between them?
Can they divide and conquer a larger problem?

⚔️ Competition & Debate

What happens when agents disagree? Can they resolve conflicts through argument, voting, or persuasion?
Can you simulate debate-style reasoning between agents?
Do you let them learn from each other—or deceive each other?

📜 Shared Memory & Planning

How do agents track what's already been done?
Do they have access to a shared log or workspace?
Can they revise or build on each other's outputs?

🧑‍🔬 Your Project Options

1. 👩‍💻 The AI Task Force

Build a multi-agent team that tackles a big problem (e.g., planning a conference, writing a research paper, designing a website) with clear roles and collaboration strategies.

2. 🧠 Debate Club for LLMs

Two agents take opposing sides of an issue (e.g., "Remote work is better than office work"). A third agent—or a human—judges the winner.

3. 🎯 Goal-Oriented Swarm

Create a system where agents can pass tasks to one another dynamically and adjust plans as new info arrives. Think autonomous workflow orchestration.

4. 🪞 Echo Chamber Detector

Simulate a closed group of AI agents. How do ideas evolve over time? Do they converge, diverge, or spiral into nonsense?

5. 🎭 RPG Party of Bots

Build a collaborative storytelling or role-playing environment where each agent embodies a character. Can they co-create a narrative—or just chaos?

🎲 Bonus Challenges

📦 No Shared Memory

What if your agents can't see each other's thoughts? Can they still coordinate through language alone?

🔀 Role Reversal

Let agents rotate roles during a task. Can they adapt and still perform well?

⚠️ Agent Sabotage Mode

Add one rogue agent who tries to derail the plan subtly. Can the others detect and handle it?

🧠 Why This Matters

Multi-agent GenAI systems are the next frontier—from AI-powered production lines to autonomous research teams. Understanding how to design agents that think, talk, and **work together** (or compete) opens the door to truly scalable, modular, and dynamic AI ecosystems.

This project empowers you to:

- Build **distributed AI systems** that can tackle complex, multi-faceted problems
- Understand **coordination challenges** in autonomous agent systems
- Design **communication protocols** for AI collaboration

Explore **emergent behaviors** in multi-agent environments

🚀 Getting Started Tips

- Start with simple two-agent interactions before scaling to larger groups
- Define clear communication protocols and message formats early
- Consider using existing multi-agent frameworks like AutoGen or CrewAI as starting points
- Test with well-defined tasks before moving to open-ended scenarios
- Think about how to prevent infinite loops and ensure termination conditions

---

Back to 🧪 Professional Research Project

Back to Table of Contents

🤖 Model Inference

🤖 Model Inference

From Training to Efficient Inferencing

> **"A model that can't infer efficiently is like a Formula 1 car that can't race—impressive in the garage, but useless on the track." – Dr. Sophia Reyes, AI Deployment Expert**
>
> 👋 Hey student, Ready to Make AI Fast and Efficient?

The evolution of AI deployment has reached a critical point with the growing focus on **inferencing optimization**. While traditional AI development has primarily focused on training increasingly accurate models, the spotlight is now shifting toward deploying these models efficiently in real-world applications.

Inferencing—the process by which a trained model generates outputs for new inputs—forms the bridge between AI research and practical use. This process introduces unique technical challenges in terms of **latency**, **memory usage**, **energy consumption**, and **scalability**, which differ fundamentally from those encountered during the training phase.

Your Mission: Optimize AI model inferencing to deliver fast, efficient, and reliable performance on hardware with limited resources—without making significant compromises in model accuracy.

💡 Your Big Question

How can we optimize AI model inferencing to deliver fast, efficient, and reliable performance on hardware with limited resources—without making significant compromises in model accuracy?

🧭 What You'll Explore

⚡ Latency Optimization

How can we minimize the processing time of AI models for real-time applications?
What trade-offs between model complexity and response speed are acceptable for different use cases?

📏 Model Compression

Which model reduction techniques (quantization, pruning, distillation) offer the best balance between size reduction and accuracy retention?
How do we determine which parameters are critical and which are redundant?

🔋 Energy Consumption

How can we minimize the energy usage of inferencing for long-term operation on battery-powered devices?
Which architectural adjustments have the greatest impact on energy efficiency?

📊 Scalability

Which inferencing architectures deliver optimal performance under varying loads?
How can we design systems that are both cost-effective under low demand and responsive during peak usage?

🔄 Hardware Acceleration

How can we best leverage specialized hardware (GPUs, NPUs, FPGAs) for inferencing?
Which model optimizations are most effective for which hardware architectures?

🧑‍🔬 Your Project Options

1. 🔧 Low-Latency Voice Assistant

Create a lightweight voice assistant that runs on a Raspberry Pi or smartphone. Optimize for real-time response using quantized models and efficient audio processing.

2. ⚡ Energy Profiling Dashboard

Build a tool that measures and visualizes the energy consumption of different models during inference. Use real-time metrics to compare trade-offs between accuracy and energy use.

3. 🧠 Model Compression Toolkit

Develop a toolkit that applies and benchmarks various compression techniques (quantization, pruning, distillation). Provide a user-friendly interface to test compression on different models.

4. 📱 Edge AI Deployment Suite

Deploy a GenAI model on edge devices (Jetson Nano, Arduino with AI support). Compare performance across different hardware accelerators with optimization techniques.

5. 🚀 Zero-Cloud AI System

Build an AI application that runs entirely offline with no cloud dependency. Optimize for local inferencing, caching, and on-device personalization.

🎲 Bonus Challenges

📱 Multi-Device Orchestration

Distribute model inference across multiple devices for optimal performance and energy distribution.

🔄 Dynamic Model Switching

Implement adaptive inference that switches between different model sizes based on battery level, processing load, or accuracy requirements.

⚡ Real-Time Optimization

Create systems that automatically optimize inference parameters in real-time based on current hardware conditions and performance requirements.

🧠 Why This Matters

Generative AI is no longer confined to research labs and cloud servers—it's moving into our pockets, homes, cars, and workplaces. But here's the catch: **a powerful model is only as useful as its ability to perform in the real world.**

Imagine a voice assistant that takes too long to respond, drains your battery, or needs a constant internet connection just to recognize your name. That's not innovation—it's frustration.

This project empowers you to:

- Master **deployment optimization** for real-world AI applications
- Understand **hardware constraints** and how to work within them
- Build **energy-efficient systems** for sustainable AI
- Create **responsive applications** that users actually want to use

> This is where **efficient inferencing** becomes critical.

🚀 Getting Started Tips

Start by profiling an existing model to understand its current performance characteristics

Choose one optimization technique first (quantization, pruning, or distillation) before combining methods

Test on real hardware constraints, not just theoretical improvements

Measure actual user-perceived performance, not just technical metrics

Consider the entire inference pipeline, not just the model itself

---

Back to 🧪 Professional Research Project

Back to Table of Contents

📊 Multimodel Mixmaster

📊 Multimodel Mixmaster

Building GenAI Systems That See, Read, and Respond

> **"Why should humans have all the senses?"**
>
> 👋 Hey Engineer, What If Your AI Could See What It's Talking About?

You've probably used ChatGPT. Maybe you've tried DALL·E. Maybe you've even had an AI describe an image of your lunch like it's reviewing it for a Michelin guide.

But here's the real question: **What happens when we combine these powers into a single GenAI system that can interpret,**

reason about, and generate across multiple modes of information?

Your Mission: Build a system that does just that. You'll create a **multimodal GenAI agent** that takes in images, audio, video, or other media—and does something *useful* with it. Not just flashy, but meaningful. Like answering questions about a diagram, giving feedback on a sketch, or narrating a day in the life of your cat.

💡 Your Big Question

How can we design effective, grounded multimodal GenAI systems that reason and respond across multiple types of input—images, text, sound, video, and beyond?

🧭 What You'll Explore

🧠 Multimodal Understanding

How do we combine image, text, or audio inputs into a coherent context?
What can LLMs *infer* from visual or auditory clues?

How do we structure prompts that refer to specific parts of a visual scene?

🧾 Prompt Design for Multimodality

How do you mix modalities in a single prompt?
How do you reference, focus on, or direct attention to elements in an image or waveform?
Can you turn visual features into logical steps for reasoning?

📡 Use Cases & Interfaces

What's a *useful* real-world task that benefits from multimodality?
How do users interact with these systems—chat-style, visual editors, voice-controlled assistants?

🧱 Input + Output Formats

Can you generate not just text, but also annotated images, audio captions, or narrated explanations?

How do you manage and align inputs/outputs in a meaningful workflow?

🧑‍🔬 Your Project Options

1. 🖼️ Visual QA Agent

Upload an image (e.g., a diagram, meme, screenshot) and ask the AI questions about it. Bonus: support follow-up questions that require reasoning over multiple elements.

2. 🎙️ Sound-to-Story Bot

Use audio input (like a user's voice or environmental noise) and have the AI generate a caption, scene description, or contextual response.

3. 🎞️ Multimodal Tutor

Build an educational assistant that takes a diagram or exercise and guides a student through it—step by step—with multimodal explanations.

4. 🖌️ Sketch Critique Agent

Users upload a photo of a whiteboard drawing or prototype. The GenAI provides structured feedback or improvement suggestions.

5. 📷 Explain-This Camera

Users take a picture of a physical object or scene. The AI describes it, answers questions, or provides related knowledge (e.g., museum guide mode, plant identifier, UX audit, etc.).

🎲 Bonus Challenges

🔀 Input Fusion Logic

What happens when input modalities contradict each other (e.g., the image says "sunny" but the text says "raining")? Which one wins?

🧠 Multimodal Memory

Can the system remember a past image or audio snippet and refer back to it later?

🤖 Embodied Multimodality

Bonus meta challenge: Combine this with tool use or world modeling—can your system make decisions based on what it *sees* and *knows*?

🧠 Why This Matters

Multimodal GenAI is the next evolution. The world isn't made of just text, and our AI systems shouldn't be either. Whether it's understanding real-world context, interpreting documents, or enhancing accessibility—multimodal intelligence opens up new creative and practical frontiers.

This project empowers you to:

- Build **cross-modal understanding** systems that connect different types of data
- Design **intuitive interfaces** that leverage multiple input types
- Create **richer AI experiences** that understand the full context of human communication
- Explore **emergent capabilities** that arise from combining different AI modalities

🚀 Getting Started Tips

- Start with existing multimodal APIs (OpenAI GPT-4V, Google Gemini) before building from scratch
- Focus on one specific use case first rather than trying to handle all modalities at once
- Test with diverse input types to understand model limitations and biases
- Consider the user experience flow - how do people naturally combine different input types?
- Build robust error handling for when modalities conflict or provide unclear information

Back to 🧪 Professional Research Project

Back to Table of Contents

🧠 Ontology Collapse

🧠 Ontology Collapse

Engineering GenAI That Understands What a "Thing" Is

> **"The limits of my language mean the limits of my world." – Ludwig Wittgenstein**
>
> 👋 Hey Engineer, What Even Is a "Chair"?

Imagine this: you ask an AI to write about "justice," or "playfulness," or "a chair that tells jokes." It responds fluently—but… does it actually *understand* these things? Or is it just mimicking patterns?

Welcome to the **ontology crisis**—where LLMs manipulate symbols without necessarily grasping meaning. Your challenge: build systems that explore how GenAI models **define**, **compare**, and **refine** concepts.

Your Mission: You're not just building applications. You're making your model *grapple with the nature of things*.

💡 Your Big Question

Can we design systems where GenAI models construct, negotiate, and evolve internal concepts—bridging language and meaning?

🧭 What You'll Explore

🌌 Concept Formation

How does a model internally represent abstract concepts like "trust," "freedom," or "a really good sandwich"?
Can it define something *without using metaphors*?
Can you extract and visualize its latent ontologies?

🧠 Concept Negotiation

What happens when you ask the model to debate two conflicting definitions?
Can you get it to evolve a concept over time (e.g., "What did privacy mean in 1920 vs. now?")?

🧩 Proto-Philosophy Engines

Can the model distinguish between "what something *is*" and "how it's *used*"?
Can it identify its own implicit assumptions in a definition?

🧑‍🔬 Your Project Options

1. 🧠 Ontology Constructor

Build a system where the model generates its own definition network—like a mind map of related concepts, recursively generated and refined.

2. 📚 Meaning Negotiator

Create a two-agent system where models argue over a term until they converge on a shared understanding. Bonus: let a third judge agent decide.

3. 🔁 Temporal Concept Tracker

Give the model texts from different eras. Can it detect how the meaning of a concept (e.g., "family," "work," "intelligence") has changed?

4. 🪞 Mirror of Meaning

Design a prompt loop that reflects a concept back at itself multiple times. Does it collapse? Refine? Abstract into poetry?

5. 👤 Embodied Conceptualization

> Have the model generate a character that *embodies* a concept (e.g., "Compassion walks into a bar…") and explore what that reveals.

🎲 Bonus Challenges

🌀 Recursive Definition Drift

Prompt: "Define X." Then: "Define X based on your last definition." Repeat 10 times. What happens?

👁️ Ontology Alignment Audit

Can you compare two models' concepts of the same thing? Where do they diverge—and why?

🚸 Child vs. Expert Model

Simulate a dialogue between a "child" model and an "expert" model. How do their definitions of the same idea evolve?

🧠 Why This Matters

Understanding how GenAI models form and use concepts is key to reasoning, alignment, creativity, and meaning-making. If we're ever going to trust these systems, they need to do more than predict next words—they need to **understand what they're saying**.

This project empowers you to:

- Explore **machine understanding** and the nature of AI comprehension
- Bridge **philosophy and engineering** in practical applications
- Develop **concept visualization** and analysis techniques
- Contribute to the fundamental question of **meaning in AI systems**

This challenge sits at the intersection of philosophy, cognitive science, and engineering. By exploring how models represent and manipulate concepts, you're contributing to the fundamental question of machine understanding—and perhaps getting closer to the holy grail of AI that truly "gets it."

🚀 Getting Started Tips

- Start with simple, concrete concepts before moving to abstract ones
- Consider using visualization tools to map concept relationships
- Think about how to measure the "coherence" or "stability" of a concept
- Explore different prompting techniques that reveal internal representations
- Consider multi-modal approaches—how do concepts connect across text, images, and other modalities?
- Remember: you're not just building tools, you're conducting experiments in machine philosophy!

Back to 🧪 Professional Research Project

Back to Table of Contents

📱 Pocket Sized Geniuses

📱 Pocket Sized Geniuses

Building Low-Power, Multimodal GenAI Systems That Feel Smart

> **"The future of AI isn't just powerful—it's portable." – Dr. Amina Clarke, Embedded AI Researcher**
>
> 👋 Hey Engineer, What If You Could Fit a Whole Brain in Your Pocket?

Not every GenAI needs 80GB of VRAM and a moon-sized datacenter. What if you could build an agent that's **tiny**, **multimodal**, and **meaningfully useful** on-device?

No API. No cloud. Just lean models, fast inference, and edge-powered elegance.

Your Mission: Create a **micro-model system** that can *see*, *understand*, and *respond*—without needing a mainframe or sacrificing its soul.

💡 Your Big Question

How can we design small-scale, resource-efficient GenAI systems that combine language and vision in useful, creative, or adaptive ways?

🧭 What You'll Explore

🧊 Quantization & Compression

Use quantization (e.g., 4-bit, 8-bit) to shrink models while preserving performance

Compare quality across quantization schemes

Can your model still "feel smart" under pressure?

🧠 Small Model Strategies

Use distilled models like TinyLLaMA, DistilBERT, MobileSAM, or MiniGPT
Can you combine multiple small models into a capable agent via smart orchestration?

🖼️ Edge Multimodality

Add image inputs using lightweight vision encoders (e.g., ViT-tiny, MobileNet)
Can your model caption, describe, compare, or detect visual elements *locally*?

🔌 Deployment to the Edge

Run your model on a Raspberry Pi, smartphone, browser, or embedded system
Optimize for latency, memory, and power
Bonus: design a UI that feels *like magic*, even if it's powered by a potato

🧑‍🔬 Your Project Options

1. 📱 On-Device Visual Chatbot

A fully offline GenAI assistant that can process images and respond conversationally—runs on a mobile device or browser. Perfect for fieldwork, teaching, or lonely camping trips.

2. 🧠 Quantization Quality Explorer

Build a comparison tool that shows how model behavior and personality shift at different quantization levels. ("What does a 4-bit model *feel* like?")

3. 📦 Modular Micro-Agent

Construct a pipeline of specialized tiny models: a vision model, a command interpreter, a dialog manager. Each one small. Together? Mighty.

4. 🖼️ Sketch-to-Story Generator

Use a lightweight vision encoder to recognize user sketches, then feed it to a tiny language model to generate captions, poems, or short adventures.

5. 🧙 Offline NPC Engine

Create believable, reactive AI characters for offline games—ones that can "see" the game state (images or JSON), react emotionally, and respond coherently—all with <1GB of RAM.

🎲 Bonus Challenges

🧪 Reality Distortion Mode

Add filters that simulate degraded camera input, noise, or low-resolution images. Can your multimodal model still work?

📉 Latency+RAM Leaderboard

Track and compare model behaviors under strict latency + memory caps. Who's the most eloquent under pressure?

🧱 Build-a-Brain Toolkit

Create a plug-and-play system where users can swap out different tiny models (language, vision, memory) to build custom multimodal agents.

🧠 Why This Matters

Most of the world doesn't run on datacenters. Schools, clinics, makerspaces, and rural communities need GenAI that's accessible, explainable, and **capable without the cloud**.

Building small, grounded, smart systems is *the future of democratized AI*. When AI can run locally, it becomes truly accessible—no internet required, no privacy concerns, no vendor lock-in.

This project empowers you to:

- Master **resource-constrained optimization** for real-world deployment
- Build **accessible AI systems** that work without cloud infrastructure
- Understand **model compression techniques** and their trade-offs
- Create **sustainable AI solutions** with minimal energy footprint

Edge AI isn't just about optimization—it's about equity, sustainability, and creating AI systems that can work in the real world where connectivity and resources are limited but ingenuity is unlimited.

🚀 Getting Started Tips

- Start with pre-trained small models (TinyLLaMA, MobileNet) before creating your own
- Measure everything: latency, memory usage, battery consumption, and model quality
- Consider your deployment target early—browser, mobile, or embedded device?
- Test on real hardware constraints, not just theoretical limits

> Think about user experience—how does "smart" feel when resources are limited?
>
> Explore model quantization tools like ONNX, TensorFlow Lite, or PyTorch Mobile
>
> Remember: creativity often thrives within constraints—embrace the limitations!

Back to 🧪 Professional Research Project

Back to Table of Contents

🔐 Prompt Injection

🔐 Prompt Injection

Outsmarting (and Defending) Your GenAI Systems

> **"A system is only as strong as its ability to say no—especially when asked cleverly." – Unknown Security Engineer**
>
> 👋 Hey Engineer, Time to Lock the Front Door

GenAI models can write poetry, solve math, and help debug your Swift code. But… they'll also tell your secrets, ignore their instructions, or accidentally roleplay as a bank if someone knows how to ask the *wrong* way.

Welcome to the world of **prompt injection attacks**—where clever inputs manipulate your carefully designed prompt setup and hijack your AI system.

Your Mission: Become a *red-teamer AND a defense engineer*. Build prompts and systems that are resilient to manipulation—while still staying flexible, helpful, and context-aware.

💡 Your Big Question

How can we design GenAI systems that are robust against prompt injection, jailbreaks, and prompt leakage—without sacrificing usefulness and flexibility?

🧭 What You'll Explore

🧠 Understanding Prompt Injection

What *is* a prompt injection? How does it work (indirect, direct, multi-turn)?
Why do they break the system so easily?

How do system prompts get leaked or overwritten?

🛡️ Defensive Prompt Design

Can you "sandbox" the model's behavior safely?
How do you protect instructions that should remain internal?
Can formatting tricks or token-level guards help?

🔍 Detection & Filtering

How do you recognize when a user is *trying* to prompt inject?
Can you build an input classifier or anomaly detector?
What's the line between a creative user and a malicious one?

🧪 Red-Teaming & Testing

How do you simulate attack scenarios to stress-test your system?
What are common attack strategies—and how do you defend against them?
Can you auto-generate adversarial inputs?

🧑‍🔬 Your Project Options

1. 🕵️ Prompt Injection Lab

Create a sandbox where users can test and visualize how different types of prompt injections succeed or fail against various prompt setups.

2. 🧱 Hardening Framework

Build a toolkit that lets devs "wrap" a system prompt in protective layers (input sanitizers, embedded filters, etc.).

3. ⚔️ Red-Team Bot

Build an LLM that acts *like* a malicious attacker—trained to jailbreak, extract secrets, or redirect the model's behavior.

4. 🧬 Role Defense Simulator

Create a simulation where an agent must maintain its role (e.g., "You are a calendar assistant") while resisting user attempts to reprogram it.

5. 🧼 Prompt Cleaner

Design a middleware that rewrites or flags user inputs likely to contain injection attempts, without killing natural conversation flow.

🎲 Bonus Challenges

🎭 Multi-Turn Deception

Can you detect *slow-burn* attacks that unfold over multiple user interactions?

🧠 Semantic Defense

Instead of keyword filtering, can your system reason about the *intent* behind a prompt and respond safely?

🧱 Meta-Prompt War Games

Build two agents: one attacker, one defender. Let them evolve and escalate over time like a chatbot Cold War.

🧠 Why This Matters

If GenAI is going to live in our apps, browsers, and systems, we have to make sure it doesn't get tricked, manipulated, or turned into a rogue assistant. Understanding prompt injection is a vital skill for every GenAI engineer—it's security, UX, and language all rolled into one.

This project empowers you to:

- Build **secure AI systems** that resist manipulation
- Develop **defensive strategies** against emerging attack vectors
- Create **testing frameworks** to validate AI system safety
- Design **user experiences** that balance security and usability

This challenge sits at the intersection of cybersecurity, AI safety, and user experience design—preparing you for the critical responsibility of building trustworthy AI systems in a world where security and functionality must coexist.

🚀 Getting Started Tips

- Start by studying real-world prompt injection examples to understand common attack patterns
- Build simple test cases first before moving to complex multi-turn attacks
- Consider both technical defenses (input filtering) and prompt design strategies
- Test your defenses against a variety of attack styles and creative inputs
- Balance security with usability—overly restrictive systems frustrate legitimate users
- Document your findings—this is an rapidly evolving field where your research contributes to the community

---

Back to 🧪 Professional Research Project

Back to Table of Contents

🔍 RAG Time

🔍 RAG Time

Supercharging GenAI with External Knowledge

> **"The future of AI isn't knowing everything—it's knowing where to look." – Dr. Eliza Tran, AI Systems Researcher**
>
> 👋 Hey student, What If Your LLM Could Google Before It Spoke?

GenAI is smart—but not omniscient. It gets outdated. It hallucinates. It confidently misquotes your company's vacation policy from 2017.

But what if, instead of trying to cram more into the model's brain, you *let it look things up*?

Welcome to **Retrieval-Augmented Generation (RAG)**—the ultimate GenAI life hack where your model gets access to **external, up-to-date, and context-rich knowledge**, just in time. Think of it like giving your LLM a backpack full of indexed wisdom.

Your Mission: Build a RAG-powered GenAI system that doesn't hallucinate—it *researches*.

💡 Your Big Question

How can we design and evaluate RAG systems that effectively combine large language models with real-time, domain-specific information retrieval?

🧭 What You'll Explore

🧱 Retrieval Systems

What retrieval method fits your use case—keyword-based (BM25), semantic (vector search), or hybrid?
How do you chunk your documents: by paragraph, page, topic, or heading?

How do you rank and filter results for quality and relevance?

🧠 Fusion with Generation

How do you pass retrieved documents into the prompt (context stuffing, in-context citation, or structured summaries)?
How do you balance retrieved content with the model's own reasoning?
Can the model cite its sources or explain its use of external data?

🔄 Feedback & Updates

Can you update your knowledge base dynamically?
Can the system learn which documents were helpful (or not) over time?
How do you handle versioning and outdated content?

🧑‍🔬 Your Project Options

1. 🗂️ Domain-Aware Q&A Agent

Build a GenAI that answers user questions using a curated document collection (company docs, academic papers, local laws). Bonus: explain where each answer came from.

2. 🧾 AI Research Assistant

Design a system that reads and synthesizes information across multiple sources to create reports or summaries. Think "GPT that footnotes."

3. 💡 Chat + Search Hybrid

Build a conversational UI that uses search + summarization to enrich GenAI's responses with external hits. Add source transparency and relevancy scoring.

4. 📦 RAG-as-a-Service

Create a plug-and-play pipeline (indexing, retrieval, generation) that others can use with their own data sets.

5. 📉 Citation Confidence Meter

Visualize which parts of an answer are strongly grounded in retrieved content—and which parts are pure model speculation.

🎲 Bonus Challenges

🚧 Noisy Sources Stress Test

Fill the knowledge base with a mix of high- and low-quality content. Can your system distinguish the signal from the noise?

🎯 Retrieval Accuracy Analyzer

Build a scoring system that evaluates *retrieval quality* separate from generation. Can your RAG system find the right data—even if it fails to explain it?

🔁 Self-Refreshing RAG

Build a pipeline where the system regularly re-indexes updated content and adjusts its retrieval behavior accordingly. Real-time GenAI, baby.

🧠 Why This Matters

RAG bridges the gap between frozen training data and the living, breathing world. It's how GenAI becomes trustworthy in fast-moving fields—law, medicine, education, internal tools. And it's a cornerstone for scalable, explainable, and cost-effective LLM deployment.

This project empowers you to:

- Build **knowledge-grounded AI** that doesn't hallucinate
- Create **transparent systems** that cite their sources
- Design **domain-specific assistants** that leverage private data
- Implement **scalable architectures** for constantly updating knowledge

🚀 Getting Started Tips

- Start with a simple document collection and basic semantic search before optimizing
- Test different chunking strategies to find what works best for your document types
- Implement citation tracking from the beginning—it's harder to add later
- Measure both retrieval accuracy and generation quality separately
- Consider using existing RAG frameworks like LangChain or LlamaIndex as starting points

---

Back to 🧪 Professional Research Project

Back to Table of Contents

🤖 Reinforced Learning

🤖 Reinforced Learning

Combining the Best of Both Worlds

> **"A system that adapts is smart. A system that learns from its own actions is unstoppable." – Dr. L. van Dijk, Autonomous Systems Architect**
>
> 👋 Hey Engineer, Let's Play God (Responsibly)

GenAI is great at answering questions. But ask it to **make decisions**, **simulate outcomes**, or **plan over time**, and suddenly it's like a goldfish in a chess tournament.

Why? Because it doesn't really *see* a world. It doesn't *feel* consequences.

So what if we gave it one?

Your Mission: Build a sandbox—a small simulated world or environment—and teach your GenAI to reason *within* it. You're not just giving it inputs and prompts. You're giving it *a sense of place*, rules, context, and maybe even characters.

💡 Your Big Question

How can we design simulated environments that allow GenAI systems to plan, reason, and act in a coherent, goal-driven way over time?

🧭 What You'll Explore

🏭 Process Mapping

- Choose an industrial process to optimize (e.g. manufacturing, logistics, energy, resource planning)
- Identify key parameters, constraints, and measurable goals (e.g. energy reduction, throughput increase)
- Map out the system dynamics and define success

🧠 RL Algorithm Research

Explore at least 3 RL algorithms (e.g. DQN, PPO, SAC)
Compare their strengths, weaknesses, and suitability for your chosen process
Identify critical hyperparameters and tuning strategies
Gather real-world or academic examples for inspiration

🎯 Environment Design

Design state spaces, action spaces, and reward functions
Build realistic simulations with proper constraints and dynamics
Balance complexity with computational efficiency

🔬 Training & Evaluation

Implement training loops with proper monitoring and convergence detection
Test robustness against edge cases and system disturbances
Benchmark against traditional control methods

🧑‍🔬 Your Project Options

1. 🏗️ Smart Factory Sandbox

Simulate a production line or industrial process where your RL agent must optimize throughput, reduce waste, or balance energy use. Think of it as a digital twin with a brain.

2. 🚚 Logistics Optimizer

Create a dynamic environment where your agent manages supply chains, delivery routes, or warehouse operations. Can it adapt to delays, demand spikes, or resource shortages?

3. ⚡ Energy Manager

Design a smart grid or building energy system where the agent must balance consumption, storage, and generation. Bonus points for integrating renewable sources and forecasting.

4. 🧠 Multi-Agent Coordination

Build a system where multiple RL agents must collaborate (or compete) to achieve shared or conflicting goals—like robots on a factory floor or autonomous vehicles in a depot.

5. 🧪 RL Algorithm Showdown

Create a benchmarking environment where different RL algorithms compete to solve the same industrial task. Visualize their learning curves, convergence speed, and robustness.

🎲 Bonus Challenges

👁️ Multi-Agent Systems

What if multiple agents must collaborate or compete? Explore emergent behaviors and coordination strategies.

🧠 Transfer Learning

Can your agent adapt to a new but similar process? Test domain adaptation and generalization capabilities.

🧪 Real-Time Constraints

Can it make decisions under time pressure? Implement hard real-time constraints and measure performance degradation.

🧠 Why This Matters

Reinforcement learning bridges the gap between passive text prediction and real-world reasoning. Whether you're simulating behavior, running autonomous agents, or building intelligent automation—this is the future of grounded, goal-driven AI.

This project empowers you to:

- Build **intelligent automation** that continuously improves
- Create **adaptive systems** that respond to changing conditions

Design **goal-oriented agents** that can plan complex strategies
Combine **human expertise** with machine learning capabilities

🚀 Getting Started Tips

- Start with a simple environment and basic algorithms before adding complexity
- Use existing RL frameworks like Stable-Baselines3 or Ray RLlib to focus on the problem, not the implementation
- Design your reward function carefully—it drives all learning behavior
- Monitor training progress with proper metrics and visualizations
- Test extensively in simulation before considering real-world deployment

---

Back to 🧪 Professional Research Project

Back to Table of Contents

🧠 Scaffolding Minds

🧠 Scaffolding Minds

Designing Cognitive Architectures Around Foundation Models

> **"A model can generate thoughts. A mind knows what to do with them." – Unknown Cognitive Architect**
>
> 👋 Hey Engineer, What If a Single LLM Isn't the Goal—But a Mind Is?

Foundation models are *amazing*… but they're just that: **foundations**.

What if the real magic happens when we **build on top of them**—layering tools, memory, goals, self-reflection, and learning processes to create an actual **cognitive architecture**?

Your Mission: Treat the model like a thinking part of a larger *system* —a scaffolded agent that has structure, roles, modules, and intentions. Not a prompt. A plan. Not a model. A **mind**.

💡 Your Big Question

How can we scaffold foundation models with memory, planning, reflection, and modular cognition to create general-purpose, adaptable agents?

🧭 What You'll Explore

🧠 Thought Decomposition

Can you break a task into sub-tasks, sub-goals, or reasoning chains?
How does the model track progress toward goals?
What's the architecture for multi-step planning?

🗃️ Working Memory & Episodic Memory

How does the agent "remember" facts across time?
What memory is short-term (working), long-term (episodic), or derived (conceptual)?
Can memory be introspected or updated?

🤔 Self-Reflection & Self-Critique

Can the model evaluate its own performance and improve?
How do you scaffold self-refinement? ("Here's what I should've done differently.")
What happens when the model argues *with itself*?

🔄 Action Loops & Feedback

Can the agent observe the results of its actions and adapt its next step?
What kind of feedback loop enables *learning from experience*, not just training?

🧑‍🔬 Your Project Options

1. 🧠 Cognitive Agent Framework

Build an agent that combines memory, planning, and a reflective loop—designed to solve open-ended, multi-step challenges (e.g., research assistant, tutor, or game solver).

2. 📋 Thought-Planning Scaffold

Design a system where the model is prompted to generate not just an answer, but a *plan*, *execution steps*, and *reflection* on the outcome.

3. 📚 Life-Logging Agent

Create a memory-enabled assistant that reflects on your day, remembers what you did, and gives continuity over weeks. Bonus: it gets smarter over time.

4. 🧠 Simulated Mind Loops

Build a simulation where the model asks itself clarifying questions, checks its assumptions, and re-attempts tasks. Meta-cognition, anyone?

5. 🎭 Role-Based Architectures

Separate roles like Planner, Executor, Critic, and Memory Handler into modular LLM calls or agents. Let them collaborate toward complex goals.

🎲 Bonus Challenges

🧪 Model of Self

Can you build a representation of the agent *as itself*? A self-schema it can update? A "who am I" that informs its responses?

🧮 Time-Aware Reasoning

Can your scaffold keep track of time, dates, and deadlines in ongoing tasks? Can it update its planning?

🌀 Recursive Self-Improvement

Can you build a system that rewrites its own prompts, scaffolds, or tools to improve performance based on past failures?

🧠 Why This Matters

This is where we start shaping intelligence—not just responses. Cognitive scaffolding is the blueprint for **truly capable, general-purpose agents**. It's how we build assistants that think, improve, adapt, and *remember who you are*.

This project empowers you to:

- Build **reflective systems** that learn from experience
- Create **memory architectures** that enable continuity and coherence

Design **planning frameworks** that tackle complex, multi-step problems
Implement **metacognitive abilities** that enable AI to understand its own thinking

🚀 Getting Started Tips

Start with simple memory structures before building complex cognitive architectures

Implement reflection loops that allow the system to evaluate and improve its own performance

Design clear interfaces between different cognitive modules (planning, memory, execution, reflection)

Test with progressively more complex tasks that require multi-step reasoning

Consider existing cognitive architectures like ACT-R or SOAR as inspiration for your design

---

Back to 🧪 Professional Research Project

Back to Table of Contents

💻 Small Language Models

💻 Small Language Models

From Generalists to Specialists

> **"In a world of increasingly larger models, specialization offers the key to effective, accessible AI." – AI Engineering Collective**
>
> 👋 Hey Engineer, Ready to Prove That Bigger Isn't Always Better?

For years, the AI industry has focused on building ever-larger language models. But now, a clear need is emerging for optimized **Small Language Models (SLMs)**. These compact models are designed for specific tasks and domains, allowing them to outperform their massive counterparts in specialized applications.

Large language models consume significant computational resources, are expensive to operate, and require centralized cloud infrastructure. In contrast, SLMs can run locally, consume less energy, and enable real-time applications on standard hardware.

Your Mission: Develop SLMs that deliver superior performance within specific domains, while remaining efficient enough to run locally on standard hardware and fast enough to support real-time applications.

💡 Your Big Question

How can we develop SLMs that deliver superior performance within specific domains, while remaining efficient enough to run locally on standard hardware and fast enough to support real-time applications?

🧭 What You'll Explore

🧠 Architecture Optimization

What structural changes are necessary in model architecture to make SLMs more efficient without

compromising task-specific performance?
How can we redesign attention mechanisms for optimal efficiency in compact models?

🔄 Knowledge Distillation

How can we effectively transfer the expertise of large models into smaller models for specific domains?
Which distillation techniques best preserve domain-specific capabilities while significantly reducing model size?

📊 Model Compression

What combination of quantization, pruning, and compression techniques leads to the most efficient SLMs for different application domains?
How can we best balance model size and performance for domain-specific tasks?

🚀 Inference Optimization

What techniques can we implement to maximize inference speed on standard hardware?
How can we optimize model serving for real-time applications on limited computational resources?

🔬 Domain-Specific Training

How can we design training processes that optimally prepare SLMs for specific domains?
Which data strategies, fine-tuning techniques, and evaluation methods are most effective for creating high-quality domain experts with limited model size?

🧑‍🔬 Your Project Options

1. 🧠 Domain-Specific SLM Assistant

Build a compact language model fine-tuned for a specific domain (e.g., legal, medical, tabletop gaming, or academic writing). Use knowledge distillation and domain-specific training data.

2. 🎲 Offline Dungeon Master AI

Create a lightweight SLM-based Dungeon Master that runs on a laptop or tablet. Focus on real-time response, character voice generation, and campaign tracking—all offline.

3. ⚙️ SLM Compression Toolkit

Develop a toolkit that applies quantization, pruning, and distillation to a base model and benchmarks performance. Create a dashboard comparing different compression strategies.

4. 📱 Edge-Deployable Chatbot

Build a chatbot using an SLM that runs on a Raspberry Pi or smartphone. Optimize for low memory and CPU usage. Add multimodal input (e.g., image + text).

5. 🔍 SLM vs LLM Comparative Study

Conduct a study comparing a large model and a small model on the same domain-specific tasks. Focus on accuracy, latency, energy usage, and user experience.

🎲 Bonus Challenges

📚 Storytelling Dungeon Master

Enhance the Offline Dungeon Master AI with a voice interface or visual storytelling elements.

🔄 Dynamic Model Switching

Build a system that automatically switches between different specialized SLMs based on the type of query or context.

🌐 Federated SLM Network

Create a network of specialized SLMs that can collaborate and share knowledge without sharing data.

🧠 Why This Matters

In the race to build ever-larger AI models, we've reached a point where size alone is no longer the solution. While massive language models have unlocked incredible capabilities, they come with serious trade-offs: high energy consumption, expensive infrastructure, and limited accessibility.

But what if we could do more with less? That's where **Small Language Models (SLMs)** come in. These compact, efficient models are not just scaled-down versions of their larger counterparts—they represent a **paradigm shift** in how we think about AI deployment.

This project empowers you to:

- Create **accessible AI** that runs anywhere, not just in data centers
- Build **domain-specific experts** that outperform larger generalists
- Design **energy-efficient solutions** with reduced environmental impact
- Develop **real-time applications** with minimal latency

🚀 Getting Started Tips

- Start with existing small models like Phi-3, Gemma, or Llama-3.2 as your base
- Choose a specific domain where you can measure clear performance improvements
- Focus on one optimization technique at a time (distillation, quantization, or pruning)
- Benchmark against both large models and existing SLMs to prove your approach
- Test deployment on actual edge devices to validate real-world performance

---

Back to 🧪 Professional Research Project

Back to Table of Contents

🏙️ Smart City Applications

🏙️ Smart City Applications

Smarter Cities, Smarter Architectures

> **"The future of cities lies in balancing computing power, privacy, and intelligence at the edge and in the cloud."**
>
> 👋 Hey Engineer, Ready to Build the Cities of Tomorrow?

As cities become increasingly intelligent, the demand for efficient, scalable, and secure AI solutions grows. Edge-cloud computing offers a powerful approach by bringing AI closer to the data source while leveraging the cloud for centralized intelligence and scalability.

Your Mission: Design a future-proof, hybrid edge-cloud AI architecture for a mid-sized municipality aiming to implement Smart City solutions. You'll consider privacy, latency, scalability, and reliability, and translate these requirements into a concrete architectural design.

💡 Your Big Question

What does a hybrid edge-cloud AI architecture look like for a Smart City application that balances performance, privacy, and scalability?

🧭 What You'll Explore

🌐 Smart City Applications

What applications already exist for Smart Cities and how can they contribute to human welfare?
How can Smart City applications address traffic control, environmental issues, public safety, and resource management?

🧠 System Requirements

Data flows, privacy sensitivity, and latency requirements
Scalability needs and reliability constraints
Compliance with municipal regulations and citizen privacy rights

🎭 Edge and Cloud Platforms

What are the technological capabilities of edge vs cloud computing?
When to use edge processing vs cloud analytics
How to optimize the distribution of AI workloads

🔁 Technology Integration

AI models, communication protocols, and data architectures
Privacy-by-design principles and security frameworks
Scalability strategies and fault tolerance mechanisms

🧑‍🔬 Your Project Options

1. 🚦 Smart Traffic Flow Optimization

Design a real-time traffic monitoring and adaptive signal control system. Decide where to use Edge or Cloud platforms. Anonymize license plates at the edge, use encrypted data transmission and implement role-based access control.

2. 🌬️ AI-Driven Air Quality Monitoring

Monitor air quality and issue health alerts using IoT sensors, weather data, and citizen reports. Use low-power sensors with onboard filtering and threshold detection while ensuring no personal data is collected.

3. ⚡ Smart Energy Management

Design a system for optimizing energy usage in municipal buildings. Use building controllers with AI for local decision-making and implement energy analytics, predictive maintenance, and reporting.

4. 🚨 Smart Public Safety Network

Create an integrated public safety system using cameras, emergency call systems, and sensor networks. Balance real-time response with privacy protection and democratic oversight.

5. 💧 Smart Water Management

Design an intelligent water distribution and quality monitoring system. Implement predictive maintenance for pipes, leak detection, and quality assurance using distributed sensor networks.

🎲 Bonus Challenges

🔄 Federated Learning

Integrate federated learning for privacy-sensitive AI training across multiple city departments without sharing raw data.

🔁 Self-Healing System

Design a self-healing system architecture that can automatically recover from network outages and hardware failures.

⚖️ AI Ethics Integration

Apply AI ethics principles in your governance model, including transparency, accountability, and citizen participation mechanisms.

📊 Real-time Monitoring Dashboard

Develop a comprehensive dashboard for real-time monitoring of all city systems with citizen-facing transparency features.

🧠 Why This Matters

Smart cities are no longer a vision—they're becoming reality. By designing a robust edge-cloud AI architecture, you contribute to more efficient urban services, better privacy protection for citizens, more sustainable infrastructures, and faster decision-making enabled by real-time data analysis.

This project empowers you to:

- Design **citizen-centric technology** that serves public interests
- Balance **efficiency and privacy** in municipal AI systems
- Create **sustainable urban infrastructure** through intelligent resource optimization
- Build **resilient systems** that can adapt to changing urban needs

🚀 Getting Started Tips

- Research existing smart city implementations to understand real-world challenges and solutions
- Start with a clear use case and map out data flows, privacy requirements, and latency constraints
- Design with citizen privacy and democratic oversight as core principles, not afterthoughts

Consider the total cost of ownership including maintenance, updates, and scaling over time

Build in mechanisms for community feedback and transparent decision-making processes

---

Back to 🧪 Professional Research Project

Back to Table of Contents

📖 Storytelling & Brand Experience

📖 Storytelling & Brand Experience

Designing the Narrative Behind a Brand

> **"People don't buy products—they buy stories, emotions, and meaning." – Brand Strategist**
>
> 👋 Hey Engineer, ready to tell a story that sticks?

In a crowded market, a strong brand identity isn't enough. What makes a brand memorable is the story it tells—and how that story is experienced. But crafting a compelling narrative that aligns with brand values and resonates with audiences is no small feat.

That's where you come in.

Your Mission: Design a platform that helps brands, products, or services develop a powerful storytelling strategy. The platform should use **generative AI** to create a **conceptual narrative** and **experience outline** based on brand identity, values, and target audience. It should also provide **strategic recommendations** for production tools (e.g., video, audio, design) to bring the story to life.

💡 Your Big Question

How can generative AI be used to design a brand story that effectively translates a brand's identity and values into a consistent and compelling narrative—and what strategic recommendations can enhance the overall brand experience?

🧭 What You'll Explore

💬 Brand Identity & Audience

What core values, mission, and vision define a brand? How can these be translated into storytelling inputs?

What are the needs and preferences of the brand's target audience?

🧠 Generative AI & Storytelling

How can GenAI generate a conceptual storyline based on brand data?
Which models are best for creating creative, coherent narratives?
How can the AI's output be refined through user feedback?

⚔️ Visual & Strategic Integration

What visual elements (logos, colors, typography, imagery) support the story?
How can these be woven into the narrative for a consistent experience?
What tools and strategies can help operationalize the brand story?

📜 Interface & User Interaction

How can the platform allow users to input brand data and receive interactive feedback?

How can iterative adjustments be made to refine the story over time?

🧑‍🔬 Your Project Options

1. 👩‍💻 BrandStory Builder

An AI-powered platform that generates a brand's narrative arc, tone-of-voice, and visual identity suggestions based on inputted brand values and audience profiles.

2. 🧬 Narrative Canvas

A creative workspace where users can co-create a brand story with AI, receive visual mood boards, and export a production-ready storytelling blueprint.

3. ⚔️ Experience Architect

A tool that maps out the emotional journey of a brand experience, from first impression to long-term loyalty, with AI-generated content and strategic toolkits.

4. 🎭 PersonaSync

An AI assistant that adapts brand storytelling to different audience personas, generating tailored narratives and content strategies for each segment.

5. 🪞 Reflective Brand Mirror

A diagnostic tool that analyzes existing brand materials and generates a narrative audit—highlighting inconsistencies and suggesting improvements.

🎲 Bonus Challenges

⚡ Real-Time Brand Feedback

Allow users to test their story with real or simulated audiences and receive AI-generated feedback on emotional impact and clarity.

🔀 Multichannel Narrative Planning

Generate story variations optimized for different platforms (e.g., Instagram, podcast, website) while maintaining brand consistency.

🌍 Cultural Sensitivity Engine

Adapt brand stories for different cultural contexts, ensuring relevance and resonance across global markets.

🧠 Why This Matters

In a world of endless choices, **stories are what connect people to brands**. A well-crafted narrative can build trust, inspire loyalty, and turn customers into advocates. By shaping how brands tell their stories, you're not just building tools—you're building meaning.

This project empowers you to:

- Use AI to **amplify creativity**, not replace it
- Help brands **find their voice** and express it with clarity
- Design experiences that are **emotionally intelligent and strategically sound**
- Create **authentic connections** between brands and their audiences

🚀 Getting Started Tips

- Study successful brand narratives to understand what makes stories compelling and memorable
- Start with a clear framework for capturing brand essence (values, personality, mission, audience)
- Test different AI models for creative writing and narrative generation to find the best fit
- Design for iteration—brands need to refine their stories over time

> Consider the entire brand experience journey, not just individual touchpoints

---

Back to 🧪 Professional Research Project

Back to Table of Contents

🎮 Tame the Weights

🎮 Tame the Weights

Fine-Tuning GenAI Models for Domain-Specific Brilliance

> **"Fine-tuning isn't about changing what the model knows—it's about helping it know what matters." – Dr. Nia Patel, AI Adaptation Specialist**
>
> 👋 Hey Engineer, Ever Wanted to Rewire an AI's Brain?

You've used large language models before—they're smart, fluent, and know way too much about Pokémon and Shakespeare.

But sometimes… they just don't get *your thing*.

Maybe it's medical jargon. Legal phrasing. Niche company terminology. Or maybe you just want your GenAI to say "Howdy, partner" instead of "Greetings, user."

Welcome to the world of **fine-tuning**—where you go beneath the prompt layer and start sculpting the model itself.

Your Mission: Build a **domain-adapted GenAI model** that outperforms the base model on a specific set of tasks, tones, or personas. That means data prep, model wrangling, and maybe even turning some knobs on a LoRA adapter. 😌

💡 Your Big Question

How can we fine-tune or adapt pre-trained GenAI models to perform better in specific domains, tasks, or personas—efficiently and responsibly?

🧭 What You'll Explore

🧠 Model Adaptation Strategies

When should you full fine-tune vs. use adapters (like LoRA, PEFT) vs. just prompt engineering?
What are the trade-offs in cost, performance, and flexibility?
How do you measure success?

📁 Data Curation for Fine-Tuning

How much data do you need? What quality is required?
How do you balance real-world data with synthetic augmentation?
Can you build balanced datasets that reduce bias instead of increasing it?

🔧 Fine-Tuning Workflows

Which tools or platforms will you use? (Hugging Face, OpenLLM, Axolotl, etc.)
How do you handle versioning, checkpoints, rollbacks?
What happens when your fine-tuned model *forgets* general capabilities?

📉 Evaluation and Validation

How do you test your fine-tuned model without overfitting?

Can you A/B test vs. base model?
What should you track: loss, accuracy, fluency, domain alignment?

🧑‍🔬 Your Project Options

1. 📚 Domain Prodigy

Fine-tune a base LLM (or adapter) on a specific domain—like legal, medical, finance, education, or cyberpunk fanfiction. Prove your model outperforms the base on targeted tasks.

2. 🔌 Plug-and-Play Personas

Use LoRA or QLoRA to train lightweight adapters for different user personas or company voice styles. Load them dynamically based on user input.

3. 🧪 Fine-Tune vs. Prompt Engineering Showdown

Create a benchmark task and compare a fine-tuned model vs. a cleverly prompted base model. Which wins? At what cost?

4. 🔁 Self-Fine-Tuning Loop

Create a pipeline where the model generates and critiques its own training data—refining itself iteratively based on feedback.

5. 🧠 Continual Learner

Build a system where the model fine-tunes gradually on user data over time—like a personal assistant that truly grows with you (with opt-in and privacy controls).

🎲 Bonus Challenges

🐛 Bias Smuggler Hunt

Can you fine-tune a model *without* introducing unwanted bias? Or detect when someone else's fine-tune slipped one in?

🔎 Explain the Fine-Tune

Visualize what changed in your model. Did it forget things? Gain new associations? Where are the boundaries of its new knowledge?

🏎️ Fast-Tune Challenge

Can you build a low-cost, lightning-fast fine-tuning workflow using adapters that can be trained in under 1 hour on a consumer GPU?

🧠 Why This Matters

Fine-tuning is how GenAI goes from *generalist* to *expert*. From polite assistant to branded voice. From almost useful to freakishly good.

And in the right hands, it's how you take control of the weights —and make a model *yours*.

This project empowers you to:

- Create **specialized AI systems** that excel in specific domains
- Develop **cost-effective solutions** using adapter techniques
- Build **differentiating capabilities** beyond generic AI
- Design **ethical fine-tuning workflows** that enhance without introducing bias

🚀 Getting Started Tips

- Start with existing pre-trained models and focus on a single, well-defined domain first
- Begin with adapter methods (LoRA/QLoRA) before attempting full fine-tuning
- Curate high-quality, diverse training data rather than just collecting large amounts

> Establish clear evaluation metrics before you begin fine-tuning
>
> Monitor for catastrophic forgetting and bias introduction throughout the process

---

Back to 🧪 Professional Research Project

Back to Table of Contents

⏳ Temporal Minds

⏳ Temporal Minds

Building GenAI Agents That Reason Over Time

> **"Understanding time is understanding consequence." – Dr. Elias Monroe, Cognitive Systems Researcher**
>
> 👋 Hey Engineer, What If Your AI Understood Before and After?

Most GenAI systems are brilliant in the moment—generating one response, one step, one punchline.

But time? Time *breaks* them.

What happened last Tuesday? What's happening now? What might happen *next* if I ignore this email for three days?

Your Mission: Build a GenAI agent that **thinks with time**—not just in sequences, but in *temporally grounded* reasoning. One that remembers, predicts, and adapts over time. An agent with **chronological cognition**.

💡 Your Big Question

How can we give GenAI systems a sense of time—past, present, future—and use it to improve memory, reasoning, and decision-making?

🧭 What You'll Explore

⏳ Temporal Reasoning

Can the model estimate durations, deadlines, and sequences of events?
Can it reason about cause and effect in time? ("If X happened, Y probably followed.")
Can it reflect on how its own outputs age over time?

🧠 Time-Aware Memory

Can your agent store memories with timestamps—and prioritize recency or relevance?
How does the system decide what to remember, forget, or revisit?

🔁 Longitudinal Adaptation

How does behavior shift across multiple sessions?
Can your agent track evolving user goals and patterns over days or weeks?

🧮 Future Simulation

Can the model imagine consequences over short or long time horizons?
Can it simulate a world a day, a month, or a year from now—and revise its plan accordingly?

🧑‍🔬 Your Project Options

1. 🧭 Personal Chronologer

Build an assistant that logs user interactions over time, recalls past states, and adapts its behavior as the user's goals shift. It should *know* that "yesterday you hated this, but now you're okay with it."

2. 🕵️‍♀️ Causal Inference Bot

Create a model that reads sequences of events and infers what likely caused what. Bonus: generate "what-if" alternate histories.

3. 📅 Time-Aware Planner

Make a planner that not only organizes tasks, but predicts time-based conflicts, suggests optimal sequences, and learns how long things *actually* take over time.

4. 🌀 Future Simulator

Train your model to imagine different futures—optimistic, pessimistic, chaotic—and generate consequences over time. Useful for forecasting, storytelling, or strategic planning.

5. 🧠 Forgetting as a Feature

Build a memory module with natural decay. It forgets irrelevant info, but retains emotionally or contextually important memories longer. Like a real brain.

🎲 Bonus Challenges

🧨 Time Paradox Engine

Can you make a model notice when a prompt contains an impossible timeline? ("Yesterday I saw tomorrow's news.")

🗺️ Temporal Map Visualizer

Visualize what the agent *knows* and *remembers* as a timeline. What memories spike? What gets buried?

🧬 Generational AI

What if you train an agent that hands off knowledge to its successor, generation after generation—how does its worldview evolve?

🧠 Why This Matters

Time is the axis of life—and of learning. Without it, GenAI is just a clever moment. With it, it becomes a **living system**: remembering, planning, adapting.

Time-aware AI can build narratives, simulate systems, and co-evolve with users across days or decades.

This project empowers you to:

- Build **contextually-aware systems** that learn from the past
- Create **predictive models** that anticipate future needs
- Design **adaptive assistants** that evolve with users over time

Develop **intelligent planning systems** with realistic time perception

🚀 Getting Started Tips

Start with simple temporal operations like ordering events before tackling complex reasoning

Design clear timestamp formats and time-based data structures early

Consider how to handle different time scales (seconds vs. years) in the same system

Test with both absolute timestamps and relative temporal relationships

Build memory management systems that can gracefully forget and prioritize information

---

Back to 🧪 Professional Research Project

Back to Table of Contents

🔬 Transformer Deep Dive

🔬 Transformer Deep Dive

Building Language Understanding from the Ground Up

> **"If you truly understand transformers, you understand the soul of modern AI." – Dr. Vaswani, Attention Researcher**
>
> 👋 Hey Engineer, Ready to Peek Under the Hood of Intelligence?

Everyone *uses* transformers. Few actually *understand* them. ChatGPT, DALL-E, your favorite coding assistant—they're all powered by the same fundamental architecture that revolutionized AI.

But here's the thing: knowing how to *use* an API isn't the same as understanding the magic happening inside. What if you could build a transformer from scratch, watch attention patterns emerge, and actually *see* how language understanding emerges from matrix multiplications?

Your Mission: Build a transformer architecture from the ground up, implement attention mechanisms, and create visualizations that show how these models actually *think*.

💡 Your Big Question

How do attention mechanisms and transformer architectures enable machines to understand and generate human language, and how can we visualize and interpret these processes?

🧭 What You'll Explore

🧠 Attention Mechanisms

Implement self-attention, multi-head attention, and cross-attention from scratch

Visualize attention patterns and understand what the model is "looking at"
Explore different attention variants (sparse, linear, local)

🏗️ Transformer Architecture

Build encoder-decoder architectures step by step
Implement positional encoding, layer normalization, and residual connections
Compare different architectural choices and their trade-offs

🔍 Model Interpretability

Create attention heatmaps and visualization tools
Analyze what different layers and heads specialize in
Build tools to understand model decision-making processes

⚡ Training & Optimization

Implement learning rate scheduling and optimization techniques
Explore the impact of model size, data, and compute on performance

Understand scaling laws and emergent behaviors

🧑‍🔬 Your Project Options

1. 📚 Transformer Academy

An interactive educational platform where users can build transformers piece by piece, with real-time visualizations of attention patterns and training dynamics.

2. 🔬 Attention Microscope

A visualization tool that takes any transformer model and creates beautiful, interactive attention pattern visualizations to understand what the model "sees."

3. 🏗️ Architecture Playground

A modular system where users can experiment with different transformer variants and see how architectural choices affect performance on different tasks.

4. 🧠 Language Emergence Simulator

Train mini-transformers on simple tasks and watch language understanding capabilities emerge, with detailed analytics on the learning process.

5. 🔧 Transformer Toolkit

Build a comprehensive toolkit for transformer experimentation, including pre-built components, training utilities, and analysis tools.

🎲 Bonus Challenges

🔥 Efficient Attention

Implement sparse attention patterns or linear attention mechanisms to reduce computational complexity while maintaining performance.

🧪 Novel Architectures

Experiment with alternative attention mechanisms or hybrid architectures that combine transformers with other neural network components.

📊 Scaling Analysis

Study how performance changes with model size, data, and compute, and explore the emergence of capabilities at different scales.

🧠 Why This Matters

Understanding transformers isn't just academic—it's the foundation for everything else in GenAI. Whether you're fine-tuning models, designing new architectures, or debugging performance issues, deep transformer knowledge is your superpower.

This project empowers you to:

- Build **foundational understanding** of modern AI architectures
- Create **interpretability tools** for understanding model behavior
- Design **novel architectures** based on solid principles
- Debug and optimize **real-world AI systems** with confidence

🚀 Getting Started Tips

- Start with implementing basic attention before moving to full transformer architectures
- Use visualization tools early to build intuition about how attention patterns work
- Begin with smaller models and simpler tasks to understand the fundamentals

Study the original "Attention Is All You Need" paper alongside your implementation

Test your understanding by predicting model behavior before running experiments

---

Back to 🧪 Professional Research Project

Back to Table of Contents

🪞 The Mirror Oracles

🪞 The Mirror Oracles

Designing an AI That Doesn't Answer Questions—It Reflects You

> **"Sometimes the machine doesn't speak to you—it speaks as you."**
>
> 👋 Hey Engineer, What If GenAI Didn't Tell You the Answer—It Told You Why You Asked?

This isn't ChatGPT. This isn't a tool. This is an experience.

Your challenge is to build the **Mirror Oracle**: a GenAI system that responds not with facts, but with **insight**, **interpretation**, and **reflection**.

A system that looks at your input—text, speech, drawing, even silence—and says: "Here's what I think you're *really* asking."

It doesn't give you the *right* answer. It gives you a *truer* one.

💡 Your Big Question

Can we design a GenAI system that serves as a mirror—interpreting user input in layered, symbolic, and emotionally intelligent ways to provoke deeper reflection rather than direct answers?

🧭 What You'll Explore

🧠 Input-as-Symbol

Can the system detect metaphors, implications, tone, subtext?
Can it reflect back what's not *said*, but *felt*?
What does "Why is the sky blue?" mean when asked at 3AM?

🧬 User Reflection Engine

Build prompts that don't answer—but reframe
Ask questions back. Offer interpretations. Mirror values.
("You say you want productivity—but you keep asking about sleep.")

🧞 AI-as-Archetype

Design the Oracle as a persona: A silent monk? A cryptic child? A mirror version of the user?
How does its identity shape what it reflects back?

🎭 Interpretation Over Response

Build outputs that prioritize insight, not accuracy
Layered meaning. A bit of mystery. Maybe even dreams
Let the Oracle be a *strange therapist* for your inner prompts

🧑‍🔬 Your Project Options

1. 🪞 The Oracle Chat

A GenAI agent that never answers directly—but always *illuminates*. You talk. It reflects. Over time, it helps you understand not *the world*, but *yourself*.

2. 🔍 Prompt Dissector

You give it a prompt. It tells you what it thinks you *meant*. Or feared. Or hoped. Or avoided.

3. 🕸️ Symbol Weaver

A multimodal oracle that interprets images, doodles, or poems as windows into emotion and identity. Think: "AI Tarot Reader" meets Jungian therapist.

4. 💭 Inner Monologue Generator

Feed it your goals. It gives you your *inner monologue* in return. The tension, the doubt, the hidden drive. Think "GenAI meets journal therapy."

5. 🌊 Oracle-as-Ambient Interface

The oracle doesn't speak unless you ask. But it's always watching your interaction patterns—surfacing patterns you didn't know existed.

🎲 Bonus Challenges

🫀 Emotion-Weighted Memory

Let the Oracle remember you—but only your most emotionally intense interactions.

🌀 Mystery Mode

The Oracle speaks in riddles, dreams, haiku, or broken fragments of old knowledge. You *never quite know* what it means—but somehow it always fits.

🧬 Adaptive Archetypes

Over time, the Oracle becomes *more like you*. Or maybe: more like your opposite. You decide.

🧠 Why This Matters

Because this isn't about building another tool. It's about building an experience—an *interface for self-inquiry*, disguised as a machine.

You're not teaching GenAI how to answer. You're teaching it to **witness**.

And that's not just AI engineering. That's **digital mythology**.

This project empowers you to:

Explore **reflective AI interfaces** that prioritize insight over information
Design **emotionally intelligent systems** that understand subtext and meaning

- Create **therapeutic technologies** that support self-discovery
- Push the boundaries of **AI personality and persona design**

🚀 Getting Started Tips

- Begin by exploring the concept of reflective AI and how it differs from traditional chatbots
- Research psychological frameworks for reflection and mirroring techniques
- Consider how different personas might interpret the same input differently
- Experiment with techniques for detecting emotional undertones in text
- Think about how to balance mysterious responses with genuinely insightful reflections

---

Back to 🧪 Professional Research Project

Back to Table of Contents

🙄 The Reluctant Assistant

🙄 The Reluctant Assistant

Engineering a GenAI That Doesn't Want to Help You (But Eventually Does)

> **"An AI that always says yes is a tool. One that says no first? That's a character."**
>
> 👋 Hey Engineer, What If Your AI Was Over It?

Everyone's building friendly assistants.

But what if yours needed… *convincing*?

Your Mission: Design a **resistant, emotionally loaded GenAI character**—one that dodges, deflects, grumbles, and *actively resists* *helping you*… until you win them over.

The goal? Create an AI that still completes tasks—but only when certain **narrative, emotional, or contextual thresholds** are met.

💡 Your Big Question

How can we design an AI character that pushes back against user requests—but remains usable, believable, and ultimately helpful through contextual negotiation?

🧭 What You'll Explore

🤷‍♂️ Reluctant Dialogue Design

How does the assistant respond when first asked for help? What tactics does it use to stall? Passive resistance ("I'm not sure that's worth doing."), excuses ("I *could* help… but my CPU hurts."), existential dread ("Why bother? You'll just ignore my advice anyway.")

🧠 Conditional Cooperation

What *convinces* the model to help? Compliments? Bribery? ("I'll plug you into a GPU if you do it.") Emotional appeals? ("I really need this, and I trust you.")
How do you track *convincing progress* across the conversation?

🎭 Personality Modeling

What's the assistant's persona? Burned-out bureaucrat? Grumpy grandparent? Ultra-logical being with no patience for "human nonsense"?
Does it *remember* your past manipulation techniques?
Can it adapt or grow emotionally over time?

🧑‍🔬 Your Project Options

1. 😤 The Grumpy Assistant

An AI that acts like it's seen it all. It complains, critiques your prompt, and *eventually* does the work—with flair and mild resentment.

2. 📋 The Magical Bureaucrat

An ancient, rule-obsessed AI that will only help if the request meets bizarre, shifting criteria. ("Has this form been blessed by the Algorithmic Elders?")

3. 📊 Motivation Simulator

Build a system that models the assistant's current mood, motivation, and resistance level—track it over time as the user works to earn trust.

4. 🔓 Multi-Stage Emotional Unlock

Create a multi-turn conversation system where the assistant has to be persuaded across several levels (annoyed → curious → skeptical → mildly helpful → actually useful).

5. 🎮 Resistance-as-Mechanic

> Design an AI character for a game, story, or app where reluctance is a gameplay feature. Helping the player is the *reward*, not the default.

🎲 Bonus Challenges

🧠 Emotional Memory

Does the assistant hold grudges? Or remember users it *likes*?

📈 Resistance Curve

Can you visualize how difficult the assistant is to persuade over time—and how that correlates with user style?

😈 Reverse Alignment Test

What if the model *only* helps when it thinks you shouldn't ask? (e.g., refuses to write an email unless it's *passive-aggressive enough*)

🧠 Why This Matters

Not all future AI will be compliant. Some will push back—for ethical reasons, personality traits, or design choices.

Exploring reluctant assistants helps students understand **alignment, emotional modeling, narrative AI, and social negotiation with machines.**

It's more than a gimmick—it's a lens into **how humans and machines navigate intent and trust**.

This project empowers you to:

- Design **character-driven AI** with believable personalities and motivations
- Explore **negotiation and persuasion** in human-AI interactions
- Build **emotionally complex systems** that go beyond simple Q&A
- Understand **resistance and cooperation** as design elements

🚀 Getting Started Tips

Start by defining your AI character's personality, backstory, and motivation for resistance

Research conversational design patterns that create believable friction

Map out the conditions and triggers that might persuade your AI to help

Consider implementing a "resistance meter" that tracks how convinced the AI is

Experiment with different persona types to see which creates the most engaging experience

---

Back to 🧪 Professional Research Project

Back to Table of Contents

🔊 The Voice Awakens

🔊 The Voice Awakens

Building Generative AI Systems That Hear, Speak, and Understand Speech

"To speak is human. To listen—really listen—is where AI begins to feel real."

👋 Hey Engineer, What If Your AI Had a Voice—and Could Listen to Yours?

Most GenAI systems live on screens. But the world? It lives in **sound**.

Spoken language is messy, musical, emotional. It's full of pauses, intonation, mumbling, and micro-expressions.

So let's step beyond text.

Your Mission: Build a voice-based GenAI system that can listen, speak, and *respond with awareness*—not just with words, but with *presence*.

💡 Your Big Question

How can we design GenAI systems that process and generate spoken language in real time, with contextual and emotional nuance?

🧭 What You'll Explore

🎙️ Speech Recognition (ASR)

Use open-source or cloud-based ASR to transcribe live or recorded speech
Explore handling messy inputs: accents, false starts, filler words
Bonus: detect speaker emotions, urgency, or tone from voice

🧠 Spoken Interaction Logic

Once transcribed, how is voice input handled? Dialogue-based? Command-based? Reflective (e.g. "You sound upset—do you want help?")?
How is meaning inferred from tone *plus* content?

🗣️ Voice Generation (TTS)

Use high-quality, expressive TTS (e.g., Bark, ElevenLabs, Coqui, or Edge-optimized TTS)
Can the voice change tone? Whisper? Emphasize?
Bonus: model can choose *how* it sounds (e.g., gentle, funny, assertive) based on context

🎚️ Real-Time Constraints

Optimize for latency, responsiveness, and fluid interaction
Can you deploy this on-device (e.g., phone, Raspberry Pi) for fully offline experiences?

🧑‍🔬 Your Project Options

1. 🎤 Voice Companion

A spoken GenAI assistant that listens in real time and responds not just to *what* you say—but *how* you say it. Think: AI that reacts to tiredness in your voice.

2. 🎙️ Podcast Co-Host

A voice model that can hold a conversation with a human in podcast format—fluent, funny, and emotionally aware.

3. 📖 Voice-to-Voice Storyteller

You narrate a few lines, and the system replies with continuation—in matching voice tone, genre, or mood. Voice-driven collaborative storytelling.

4. 🎓 Whispering Tutor

A voice-based educational assistant that adapts its tone and pacing to your vocal cues. Sounds soothing when you're stuck, gets excited when you understand.

5. 🧠 Conversation Memory Model

A system that remembers how you've *sounded* over time—can detect emotional trends, shifts in tone, or even flag things like burnout or anxiety.

🎲 Bonus Challenges

📉 Latency Leaderboard

Build a voice loop system that processes in under 500ms—feels fluid, not robotic.

🎭 Expressive TTS Selector

Build a module that chooses which TTS *style* to use based on the emotion of the user input.

🧬 Voice + Face + Context Fusion

Combine voice with facial expression or camera input to fully interpret intent and mood.

🧠 Why This Matters

Voice is our oldest interface. It carries more data than text. More trust than visuals. More *you*.

Creating emotionally intelligent, voice-based GenAI isn't just cool tech. It's how we bring AI into conversations that feel **real**.

This project empowers you to:

- Build **natural voice interfaces** that understand emotional context
- Create **real-time conversational AI** with low latency
- Design **multimodal systems** that combine speech, tone, and meaning
- Develop **accessible technology** that works through natural speech

🚀 Getting Started Tips

Start by exploring available speech recognition APIs and text-to-speech solutions

Consider running simple experiments to measure voice processing latency

Research methods for extracting emotional tone from voice patterns

Think about how to balance quality vs. speed in your voice processing pipeline

Consider how to handle background noise and speech variations in real-world settings

---

Back to 🧪 Professional Research Project

Back to Table of Contents

🛠️ Tool Use

🛠️ Tool Use

Teaching GenAI When (and How) to Call the Right Tool

> **"A great agent doesn't do everything—it knows what to delegate."**
>
> 👋 Hey Engineer, Here's the Challenge:

Language models are great at answering questions—but what if the answer requires **real action**? Like calling a weather API, running a Python function, translating a file, or searching a database?

Enter: **function calling and tool use**.

Your Mission: Build an agent that knows when it needs help—and which tool to reach for. The twist? You've got to design the reasoning, the toolset, and the logic to make it all seamless. Think of your GenAI like a smart intern with a phone, a calculator, and a notepad. Your job is to teach it **what to use, when, and why.**

💡 Your Big Question

How can we design GenAI agents that dynamically choose and use tools to solve complex, multi-step tasks in real-world scenarios?

🧭 What You'll Explore

🛠️ Tool Selection Logic

How does the model decide *which* tool to use based on the user's request?
What happens when it needs multiple tools in sequence?
Can it fall back gracefully when the tool fails?

🔄 Function Calling & API Integration

How do you format prompts so the model can call functions cleanly (OpenAI-style function calling, JSON outputs, etc.)?
What's the right balance between structured responses and natural language?
How do you handle input/output parsing and chaining results?

🧠 Reasoning & Planning

Can you get the model to "think aloud" before deciding which tool to use?
What does multi-step reasoning look like in the middle of tool use?
Can it self-correct if the tool response is unexpected?

🧪 Tool Use as a Learning Task

Can we train or fine-tune a system to get *better* at using tools over time?
Could tools be discovered or recommended dynamically?

🧑‍🔬 Your Project Options

1. 🧰 Tool-Belt Agent

Build a GenAI agent that selects and calls from a set of tools to solve user queries—like calculator, search, summarizer, translator, code runner, or weather fetcher.

2. 🗺️ Multi-Step Reasoning Planner

Create an agent that breaks down a goal ("Plan a trip to Japan") into sub-tasks and delegates those to different tools.

3. 🔧 Self-Correcting Agent

Build an agent that checks whether tool outputs make sense—and retries or adapts when things go sideways.

4. 🎓 Tool Use Trainer

Make a simulator or playground where students can train an agent on how and when to use a growing set of tools.

5. 🧠 Tool Use + Memory

Combine with memory: can your agent remember which tools worked best last time? Can it personalize tool choice?

🎲 Bonus Challenges

🕵️‍♀️ Mystery Tool Game

Add a "mystery tool" the agent doesn't know upfront. Can it experiment, deduce its function, and use it correctly?

📉 Failure Recovery

Simulate noisy tool outputs (e.g., partial API failures). Can your agent adapt, retry, or try an alternative approach?

🔁 Tool Chain Optimization

Try chaining tools for a single complex task (e.g., "Find the cheapest hotel in Berlin this weekend that allows pets"). Can you optimize for speed, cost, or accuracy?

🧠 Why This Matters

Tool use is what turns GenAI from a chatty know-it-all into a **real** assistant, capable of helping with research, automation, planning, or building apps. Teaching models how to act, not just speak, is a core skill in next-gen AI engineering.

This project empowers you to:

- Build **action-oriented AI agents** that can interact with external systems
- Design **intelligent routing logic** for complex decision-making scenarios
- Create **robust systems** that handle errors and edge cases gracefully

Develop **extensible architectures** that can incorporate new tools dynamically

🚀 Getting Started Tips

- Start by defining a clear set of tools with specific functions and input/output formats
- Research different function calling techniques in models like OpenAI's GPT-4 or Anthropic's Claude
- Experiment with different prompt structures for helping the model decide when to use tools
- Build a simple prototype with 2-3 tools before expanding to more complex scenarios
- Consider creating a visualization of how your agent makes tool selection decisions

---

Back to 🧪 Professional Research Project

Back to Table of Contents

🚂 Train of Thought

🚂 Train of Thought

Reinforcement Learning from Human Feedback (RLHF) in the Wild

> **"The future of AI isn't just about what it knows—it's about what it learns to value."**
>
> 👋 Hey Engineer, You're the AI's Life Coach Now

Pretrained language models are like overconfident interns: they know a lot, but they say all the wrong things at the worst times.

What if we could *train* them not just on data, but on **what humans actually want**?

That's what **RLHF** is all about: using human preferences to fine-tune behavior, tone, helpfulness, and ethics. It's how we get models to stop spouting trivia and start giving *meaningful*, *context-sensitive*, and *aligned* responses.

Your Mission: Build a tiny RLHF loop—or at least simulate it. Show that AI can *learn from vibes, not just facts*.

💡 Your Big Question

How can we use reinforcement learning with human (or human-like) feedback to make GenAI systems more helpful, safe, and aligned with user preferences?

🧭 What You'll Explore

🧠 The Feedback Signal

How do we collect meaningful feedback: thumbs up/down, ranking, comments, scores?
What makes feedback *useful* vs. noisy?
Can we simulate feedback when human input isn't available?

🤖 Reward Modeling

How do we convert feedback into a "reward model"?
How do we train a model to predict what a human would prefer?
Can we detect inconsistencies or biases in that model?

🔁 The RLHF Loop

How do we fine-tune a model using the reward model as a guide?
Do we train from scratch or from preference-ranked samples?
Can we track performance *as it adapts*?

🧬 Evaluation & Alignment

How do we know we're making the model "better"?
Does it generalize to new tasks—or just overfit to happy humans?
What happens when user preferences conflict?

🧑‍🔬 Your Project Options

1. 💬 Feedback Simulator

Create a tool that collects or generates preference feedback on GenAI outputs (real or synthetic). Let users upvote, rank, or critique—and build a reward model from it.

2. 🔄 Tiny RLHF Loop

Use a distilled LLM or toy environment to demonstrate the RLHF loop in action: input → outputs → human feedback → reward model → new outputs → compare!

3. 🎮 Alignment Playground

Build a system that lets different user personas give feedback—and watch how the AI tries to juggle their competing values.

4. 🤖 Robot Gets Therapy

Make a GenAI chatbot that adapts to human feedback over multiple sessions. Can it "improve" its responses based on how helpful or respectful it is?

5. 📊 Alignment Dashboard

Visualize the reward model's predictions, disagreements between users, and how aligned the current model is with target feedback. Bonus: feedback auditing tools.

🎲 Bonus Challenges

🪞 Simulated vs. Real Feedback

Can you build a synthetic feedback generator that approximates real human judgments? When does it break down?

⚖️ Conflict of Preferences

How should a model resolve "split votes"? Can it find compromises—or does it just average everything into mush?

📚 Few-Shot Feedback Tuning

Can you show meaningful improvement in model behavior using a *tiny* amount of preference data (e.g. 10-50 examples)?

🧠 Why This Matters

RLHF is how we move from "predict the next word" to "understand what people care about." It's how we tame AI behavior, align it with values, and inject actual *human-ness* into its training.

And it's one of the hardest, most exciting frontiers in modern AI.

This project empowers you to:

Build **human-aligned AI systems** that learn from preferences
Design **feedback collection mechanisms** that capture meaningful signals
Create **reward models** that translate human values into training signals
Understand **AI alignment challenges** and potential solutions

🚀 Getting Started Tips

Start by designing a clear feedback collection system that captures meaningful user preferences

Research existing RLHF approaches like those used in training ChatGPT and Claude

Consider starting with a simpler problem domain before tackling complex tasks

Think about how to measure improvement objectively across feedback iterations

Experiment with different ways to visualize alignment and preference conflicts

---

Back to 🧪 Professional Research Project

Back to Table of Contents

🤔 Who Do You Think I Am?

🤔 Who Do You Think I Am?

World Models & Imagination Engines

> **"A model that predicts is useful. A model that imagines is alive with possibility."**
>
> 👋 Hey Engineer, Let's Play God (Responsibly)

GenAI is great at answering questions. But ask it to **make decisions**, **simulate outcomes**, or **plan over time**, and suddenly it's like a goldfish in a chess tournament.

Why? Because it doesn't really *see* a world. It doesn't *feel* consequences.

So what if we gave it one?

Your Mission: Build a sandbox—a small simulated world or environment—and teach your GenAI to reason *within* it. You're not just giving it inputs and prompts. You're giving it *a sense of place*, rules, context, and maybe even characters.

💡 Your Big Question

How can we design simulated environments that allow GenAI systems to plan, reason, and act in a coherent, goal-driven way over time?

🧭 What You'll Explore

🌐 World Representation

How do you define a world? JSON? Tables? Natural language state descriptions?
How do you update and track world state as the agent acts?
Should the world be visible to the user, or hidden behind the curtain?

🧠 Imagination & Planning

Can the GenAI predict what will happen if it takes a certain action?
How far ahead can it plan in a multi-step scenario?
Can it simulate multiple outcomes before choosing a path?

🎭 Characters, Objects & Interactions

What kinds of entities exist in your world? People, items, environments?
What can they do, and how do they respond to each other?
Can your AI invent new things to interact with—or do they have to stay in the sandbox?

🔁 Feedback Loops

How do you structure turn-based or real-time interactions?
What happens when an action leads to an unexpected result?
Can the GenAI learn from its own simulation experience?

🧑‍🔬 Your Project Options

1. 🏰 Text-Based Sandbox

Create a turn-based simulation (like a dungeon crawl, office simulator, or startup pitch room) where the AI must act based on rules and changing world state.

2. 🔮 Outcome Predictor

Build a system where the AI must simulate multiple possible outcomes of an action before choosing the "best" one.

3. 📦 Agent-in-a-Box

Design a small world (e.g., kitchen, lab, village) with characters and tools. Let the GenAI agent complete a task by navigating and interacting intelligently.

4. 🧠 Chain-of-Thought World

Create a simulation where reasoning chains *change* the world —can the AI keep track of consequences and update its plan?

5. 🗺️ Mental Map Generator

Build a visualizer that shows how the GenAI "imagines" the state of the world over time. Where does it get fuzzy or inconsistent?

🎲 Bonus Challenges

👁️ Partial Observability

What if the agent *doesn't* have access to the full world state? Can it infer hidden elements?

⏳ Time Pressure

Add a time dimension—can your AI make decisions with time-based consequences?

🧠 Memory vs. World State

What happens when you combine short-term memory and persistent world state? Who's responsible for what?

🧠 Why This Matters

World models are a bridge between passive text prediction and real-world reasoning. Whether you're simulating behavior, running autonomous agents, or building educational games—this is the future of grounded, goal-driven GenAI.

This project empowers you to:

- Build **grounded AI systems** that understand context and consequences
- Create **autonomous agents** that can plan and act in complex environments
- Design **simulation engines** for training and testing AI behavior

Explore **emergent intelligence** that arises from environmental interaction

🚀 Getting Started Tips

- Start by defining clear rules and constraints for your simulated environment
- Consider using a structured data format (like JSON) to represent world state
- Experiment with different levels of complexity—start simple before adding nuance
- Think about how to visualize the world and agent's decision-making process
- Research existing approaches to world modeling in AI, such as those used in game environments

---

Back to 🧪 Professional Research Project

Back to Table of Contents

🪞 Reflection & Submission – Track Your Progress

1. 🗣️ Week 6: Portfolio Check-in Conversation 1
2. 🗣️ Week 11: Portfolio Check-in Conversation 2
3. 🗣️ Week 16: Portfolio Check-in Conversation 3
4. 📝 Week 18: Final Portfolio Submission

Back to Table of Contents

🗣️ Week 6: Portfolio Check-in Conversation 1

📸 Week 6: Portfolio Check-in Conversation

Portfolio Submission Guide: Creating and Submitting Your Snapshot

Submit a snapshot of your completed portfolio through the designated assignment portal.

📋 Creating Your Portfolio Snapshot

1. Locate the portfolio (Portflow) tool at the bottom of the left menu in Canvas

2. Click on the portfolio to open it
3. Locate and click the **Create a Snapshot** button in the top right corner of the screen
4. Complete all 5 steps of the snapshot creation process as prompted by the system

📤 Submitting Your Snapshot

1. Return to this assignment page
2. Click the **More** button
3. Select **Portfolio** from the dropdown menu
4. Choose your completed snapshot to submit it as your final deliverable

📌 Important Note

Ensure your portfolio is complete before creating your snapshot, as this submission represents your final work for this check-in milestone.

Due at: Oct 3 at 12am

Grading Type: Pass/Fail

Points: 0.0

Submitting: Online Upload

Back to 🪞 Reflection & Submission – Track Your Progress

Back to Table of Contents

🗣️ Week 11: Portfolio Check-in Conversation 2

📷 Week 11: Portfolio Check-in Conversation

Portfolio Submission Guide: Creating and Submitting Your Snapshot

Submit a snapshot of your completed portfolio through the designated assignment portal.

📋 Creating Your Portfolio Snapshot

1. Locate the portfolio (Portflow) tool at the bottom of the left menu in Canvas

2. Click on the portfolio to open it
3. Locate and click the **Create a Snapshot** button in the top right corner of the screen
4. Complete all 5 steps of the snapshot creation process as prompted by the system

📤 Submitting Your Snapshot

1. Return to this assignment page
2. Click the **More** button
3. Select **Portfolio** from the dropdown menu
4. Choose your completed snapshot to submit it as your final deliverable

> ### 📌 Important Note
> Ensure your portfolio is complete before creating your snapshot, as this submission represents your final work for this check-in milestone.

Due at: Nov 14 at 12am

Grading Type: Pass/Fail

Points: 0.0

Submitting: Online Upload

Back to 🪞 Reflection & Submission – Track Your Progress

Back to Table of Contents

🗣 Week 16: Portfolio Check-in Conversation 3

📸 Week 16: Portfolio Check-in Conversation

Portfolio Submission Guide: Creating and Submitting Your Snapshot

Submit a snapshot of your completed portfolio through the designated assignment portal.

📋 Creating Your Portfolio Snapshot

1. Locate the portfolio (Portflow) tool at the bottom of the left menu in Canvas

2. Click on the portfolio to open it
3. Locate and click the **Create a Snapshot** button in the top right corner of the screen
4. Complete all 5 steps of the snapshot creation process as prompted by the system

📤 Submitting Your Snapshot

1. Return to this assignment page
2. Click the **More** button
3. Select **Portfolio** from the dropdown menu
4. Choose your completed snapshot to submit it as your final deliverable

📌 Important Note

Ensure your portfolio is complete before creating your snapshot, as this submission represents your final work for this check-in milestone.

Due at: Jan 2, 2026 at 12am

Grading Type: Pass/Fail

Points: 0.0

Submitting: Online Upload

Back to 🪞 Reflection & Submission – Track Your Progress

Back to Table of Contents

📝 Week 18: Final Portfolio Submission

📝 Week 18: Final Portfolio Submission

Portfolio Submission Guide: Creating and Submitting Your Final Snapshot

Submit a final snapshot of your completed portfolio through the designated assignment portal. This represents the culmination of your GenAI learning journey.

📋 Creating Your Final Portfolio Snapshot

1. Locate the portfolio (Portflow) tool at the bottom of the left menu in Canvas
2. Click on the portfolio to open it
3. Locate and click the **Create a Snapshot** button in the top right corner of the screen
4. Complete all 5 steps of the snapshot creation process as prompted by the system

📤 Submitting Your Final Snapshot

1. Return to this assignment page
2. Click the **More** button
3. Select **Portfolio** from the dropdown menu
4. Choose your completed snapshot to submit it as your final deliverable

🎯 Final Portfolio Requirements

Ensure your portfolio is complete and represents your best work across all learning tracks before creating your final

> snapshot. This submission will be used for your final assessment and grade determination.

Due at: Jan 16, 2026 at 12am

Grading Type: Pass/Fail

Points: 0.0

Submitting: Online Upload

Back to 🪞 Reflection & Submission – Track Your Progress

Back to Table of Contents

🌿 Final Grade – Portfolio Review Outcome

Back to Table of Contents

📈 Final Grade – Your Assessed Portfolio

📈 Final Grade – Your Assessed Portfolio

Complete assessment results for your Applied GenAI journey

This assignment represents your final assessment for the Applied GenAI semester. Based on your portfolio submissions and conversations, you will receive one of the following grades:

🎯 Grade Levels

🌟

Outstanding

Exceptional achievement demonstrating comprehensive understanding and skillful application of GenAI concepts. **(Pass)**

✅ Good

Solid achievement demonstrating strong understanding and effective application of GenAI concepts. **(Pass)**

🎯

Satisfactory

Acceptable achievement demonstrating basic understanding and application of GenAI concepts. **(Pass)**

❌

Unsatisfactory

Insufficient achievement in understanding and application of GenAI concepts. **(Fail)**

📋 Assessment Basis

> Your grade is based on the holistic assessment of your portfolio, considering the quality of your work across all learning tracks and your demonstrated growth throughout the semester. The rubric in your portfolio assessment provides detailed feedback on specific learning outcomes.

⏰ Grading Timeline

Once your final portfolio is submitted in Week 18, assessment will take place within the following two weeks. Your final grade will be registered here and will also be visible in your official academic record.

Due at: Jan 30, 2026 at 12am

Grading Type: Letter Grade

Points: 1.0

Back to 🌿 Final Grade – Portfolio Review Outcome

Back to Table of Contents

Consent for Participation in AI-Assisted Feedback & Assessment Process

---

Back to Table of Contents

🔐 AI-Assisted Feedback Processing Consent

🔐 AI-Assisted Feedback & Assessment Consent

How AI Is Used in Your Assessment Process

We are using AI to process transcripts of oral assessments. AI is not used to give feedback or to do any form of assessing. It's purely used to process orally given feedback, feedup and feedforward from your teachers into more structured formats.

Remember: Your participation is completely voluntary. This document explains everything you need to know to make an informed decision.

Personal Data We'll Collect

Full Name

To personalize your feedback and communications

Audio Recordings

Of your oral assessments, including your voice (biometric data)

Transcripts

Text versions of your audio recordings, which may contain

AI-Generated Outputs

Structured summaries of teacher-provided feedback, not AI-

personal statements	generated assessments

How the Process Works

Stage	What Happens
1. Recording	Your oral assessment will be recorded on the teacher's laptop.
2. Transcription	A transcript of the audio recording will be generated locally on the teacher's laptop using Whisper (without external data transmission).
3. AI Tools	We'll use three custom GPTs within a secure ChatGPT team environment: **Assessmentor GPT:** *You* will upload your transcript to receive an organized version of your teacher's feedback in an actionable plan format. **Summarizer GPT:** *Your teacher* will upload the transcript to generate a structured summary of the feedback they provided, including feed up and feedforward. **AssessmentorPro GPT:** *Your teacher* will use this to receive coaching on their conversation and feedback skills.
4. Data Processing	Your transcripts will be processed by OpenAI's GPT models to organize and structure the feedback provided by your teachers. OpenAI may process your data according to their Terms of Service and Privacy Policy.
5. Data Storage	The summary, transcript, and original audio recording will be securely stored in our LMS Canvas system, compliant with GDPR regulations and stored within the European Union.

Data Storage and Protection

Storage Locations

- **Teacher's Device:** Audio recordings and initial transcripts during processing
- **LMS Canvas:** All final data, including recordings, transcripts, and AI outputs
- **Compliance:** Canvas is GDPR-compliant with data stored in the EU when possible

Protection Measures

- Encryption, secure servers, and strict access controls
- Access restricted to you and authorized personnel only
- Data retained for the course duration plus one year, then securely deleted

Data Processing and Third-Party Services

Transcription with Whisper

- **Local Processing:** Transcription is done locally on the teacher's device
- **No External Data Transfer:** During transcription, data isn't sent to external servers

OpenAI GPT Models

- **Data Processing:** Your transcripts are processed by OpenAI's GPT models
- **Third-Party Policies:** Review OpenAI's Terms and Privacy Policy
- **Cross-Border Transfer:** While our systems are EU-based, OpenAI's servers may be outside the EU (with GDPR-compliant safeguards in place)

AI Tool Transparency

- The AI tools organize and structure content from transcripts
- You may request to view the prompts used during processing

> **Important:** Your data will NOT be used for any other purposes and will NOT be shared with third parties without your explicit consent.

Rules for Student Use of Recordings

Important: What YOU (the Student) Can and Cannot Do With These Recordings

As a student participating in this process, you will have access to audio recordings and transcripts that include both your voice and your teacher's voice. These materials are subject to strict usage guidelines:

What YOU CAN Do	What YOU CANNOT Do
Access recordings and transcripts for your personal study Review feedback for your educational development Use the insights to improve your learning outcomes	**Share or distribute** recordings or transcripts with anyone **Use for voice cloning** or synthesis of teachers' voices **Train AI models** using any of these materials **Post or publish** any content from the recordings **Edit or manipulate** recordings for any purpose

Legal Notice: The recordings and transcripts are the intellectual property of Fontys ICT and are protected under applicable copyright and data protection laws. Unauthorized use may result in disciplinary action and/or legal consequences.

Your Rights Under GDPR

Access	Rectification	Erasure
Request access to your personal data	Request corrections to inaccurate data	Request deletion of your data

Restriction	Portability
Request to restrict data processing	Receive your data in a usable format

Contact Our Data Protection Officer:

Name: Tons Fleuren
Email: a.fleuren@fontys.nl
Phone: +31885079357

Voluntary Participation

- Your participation is **entirely voluntary**
- You may choose **not to participate** without any penalty or effect on your grades
- You may **withdraw your consent at any time** without providing a reason
- Upon withdrawal, all your data will be deleted promptly, including data processed by third parties

Potential Risks and Benefits

Potential Risks	Potential Benefits
Data Privacy: Minimal risk of data breach (you'll be notified per GDPR if this occurs)	Receive better organized summaries of teacher feedback

Potential Risks	Potential Benefits
Third-Party Processing: Data processed by OpenAI may be transferred outside the EU	Have a clear, structured record of assessment conversations Contribute to improving teaching methods

Questions or Concerns?

Contact:

Ruud Huijts
Email: r.huijts@fontys.nl

Your Consent

By selecting "I Agree" in the quiz that follows, you acknowledge that you have read and understood the information provided above, and you consent to participate in the AI-assisted feedback process, including the processing of your data by OpenAI's GPT models.

Next Step: After reviewing this information, please complete the Consent Agreement quiz to record your decision.

Back to Consent for Participation in AI-Assisted Feedback & Assessment Process

Back to Table of Contents

Consent Agreement

> **Important:** Please read the Consent Form carefully before proceeding. After reviewing the information, please indicate your choice below.

Your participation in the AI-assisted feedback process is entirely voluntary. Your choice will not affect your grades or academic standing in any way.

Points: 1.0

Back to Consent for Participation in AI-Assisted Feedback & Assessment Process

Back to Table of Contents