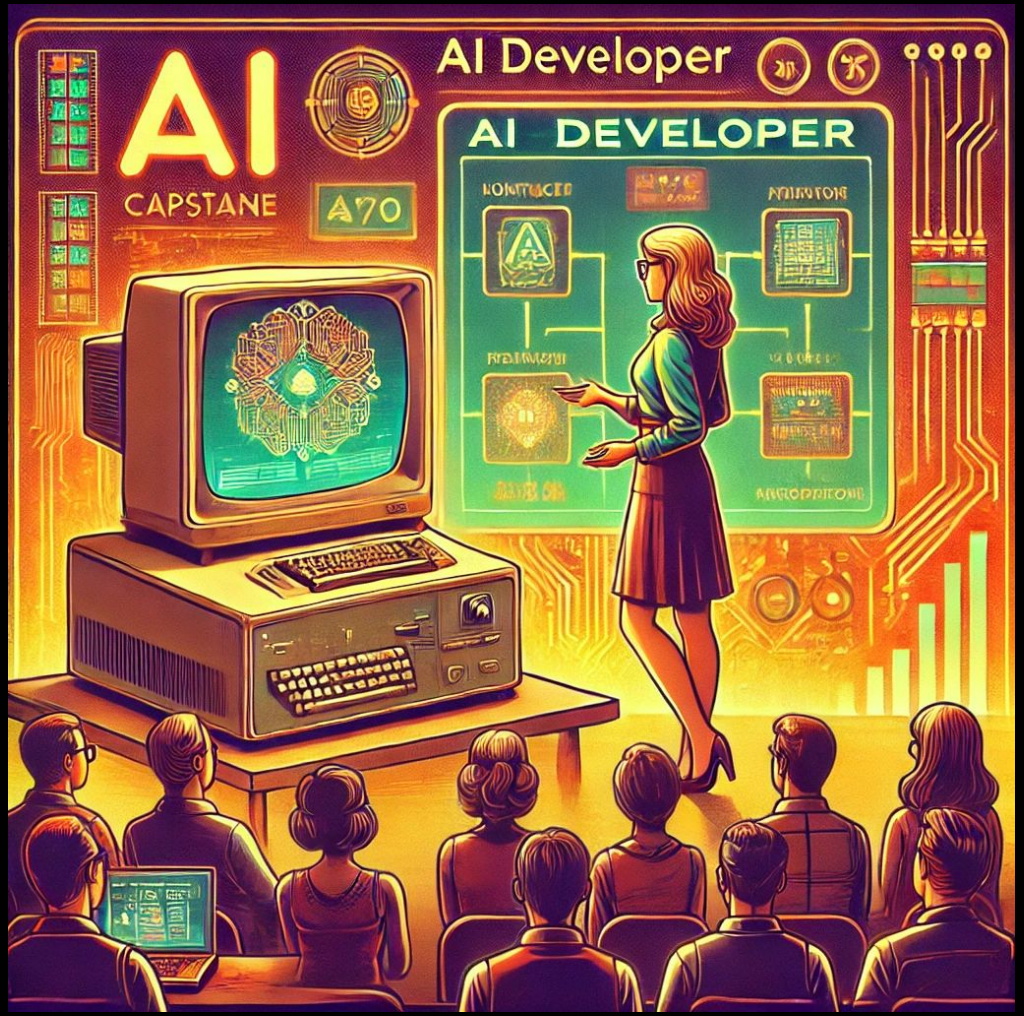
ODSC MINI-BOOTCAMP

Ai+



Capstone Project

Why Build Your Own AI Project?

🎯 **Stand Out in the AI Job Market**

🛠️ **Apply What You've Learned**

🔥 **Build Something You're Proud Of!**

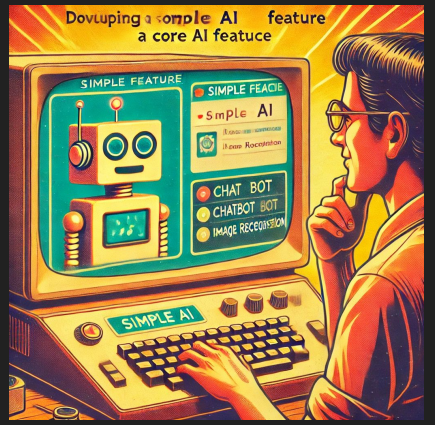
🚀 **Hone your AI Skills!**

Which Project

Keep it simple and achievable.

Pick a problem you care about.

Focus on a core AI feature rather than a full app.



CAPSTONE PROJECT CHALLENGES

Common Challenges & How to Overcome Them

🚧 **“I don’t know what to build.”** → Start small, pick a problem that interests you and will motivate you.

🚧 **“I don’t have enough data.”** → Use open datasets or pre-trained models.

🚧 **“I don’t know how to code.”** → Start at the beginning and build up your coding skills. Read, study and leverage AI. Overall be patient with yourself

🚧 **“My project isn’t working.”** → Ask for feedback, debug, and iterate. Leverage ODSC, AI+ Traning, and AI code assist tools

STEPS TO BUILDING YOUR PROJECT

✅ Step 1: Define Your Idea – What problem are you solving?

✅ Step 2: Choose Your Tools – Python, TensorFlow, Hugging Face, OpenAI API

✅ Step 3: Gather & Prepare Data – Use datasets from Kaggle, Hugging Face

✅ Step 4: Train or Fine-Tune a Model – Start small, experiment with pre-trained models.

✅ Step 5: Build a Simple Interface – Gradio, or Streamlit

✅ Step 6: Test & Improve – Debug, refine, and iterate.

✅ Step 7: Showcase Your Work! – Share on GitHub, LinkedIn, or a personal website

DATASETS FOR YOUR AI PROJECT

Accessing high-quality datasets is key to building and training effective AI models

- **Kaggle –** Massive collection of datasets with built-in tools for analysis
- **Hugging Face Datasets –** Pre-curated datasets for NLP, vision, and more
- **Google Dataset Search –** Helps find publicly available datasets
- **Data.gov –** Open government datasets across multiple domains
- **UCI Machine Learning Repository –** Classic datasets for ML practice
- **Awesome Public Datasets** – Curated list of datasets on GitHub
- **FRED (Federal Reserve Economic Data)** – economic and financial indicators

IMPROVE YOUR CODING SKILLS

Here are some beginner-friendly introductions to Python and real-world applications. Study these resources which will help you **write better code, solve problems, and think like a programmer**. Pair them with hands-on 🚀**AI code assistant practice** 🚀to get the most out of your learning journey!

- **A Byte of Python by Swaroop C.H**
- **Python for Everybody: Exploring Data in Python 3 by Dr. Charles Severance**
  - **Supplementary video lectures:**
- **Think Python: How to Think Like a Computer Scientist by Allen B. Downey**
- **Elements of Data Science by Allen Downey**

AI CODE ASSISTANTS

🚀 **Boosting Coding Productivity for AI Development**

What Are AI Code Assistants?

AI-powered coding tools help developers **write, debug, and optimize code faster** by offering real-time suggestions, autocompletions, and explanations.

Google Colab with Gemini – AI Coding in the Cloud

✅ **Cloud-Based Notebooks** – No local setup needed, runs in a browser.
✅ **AI-Powered Assistance** – Uses Google's Gemini AI for code suggestions.
✅ **Integrated GPU Support** – Free-tier GPUs for AI training.
✅ **Collaboration-Friendly** – Share and edit notebooks easily.

💡 **Best For:** Beginners or data scientists who want **quick AI coding in a cloud notebook**.

PRO TIP

Prompt ChatGPT to Create PROMPTS :use AI to build AI

~~**Create an image** of a person on a computer reviewing ChatGPT~~

Craft a prompt to produce an image of a person sitting next to a terminal in pop-art comic book style

Prompt:

"A person sitting next to a vintage computer terminal, surrounded by speech bubbles and halftone patterns, in the style of a 1960s pop-art comic book. The person has a thoughtful expression, bright bold colors (reds, yellows, blues), thick black outlines, and dramatic shading. The background includes retro tech elements, comic-style action lines, and a classic dot texture. The terminal glows slightly, and there's a caption box with comic book text like 'CHATGPT for BETTER PROMPTS'"

FROM CO-LAB TO LOCAL CODING

Why Make the Switch?

- Full Control – Install custom libraries, configure settings, and optimize performance.
- Offline Development – Work without an internet connection.
- Better Integration – Use local files, databases, and connected devices.
- No Session Timeouts – Avoid Colab's runtime disconnections.
- Prepare for Real-World Development – Professional workflows rarely depend on Colab.

AI CODE ASSISTANTS

Cursor AI provide AI-Powered Coding in Visual Studio

✅ **AI Autocomplete – Suggests complete functions and optimizations.**

✅ **In-Editor Chat – Ask coding questions directly inside VS Code.**

✅ **Code Refactoring – Helps improve efficiency and readability.**

✅ **Works Locally – No internet required after installation.**

💡 **Best if you want to become a Data Scientist, AI Engineer or improve your Developer skills**

Downsides of Using Cursor AI for AI Coding

❌ **Managing Your Own Virtual Environments – Requires setting up and maintaining envs manually.**

❌ **Dependency Management – You must install and update libraries torch yourself.**

❌ **Handling Environment Variables – manually setup configure .env files.**

❌ **No Built-in GPUs – Unlike Google Colab, tools like Cursor AI etc do not provide free cloud GPUs**

AI CODE ASSISTANTS

Which One Should You Use?

- **Use Cursor AI if you:**

  - Want AI-assisted coding inside **VS Code**.
  - Prefer **local development** with virtual environments.
  - Need AI **to refactor, debug, and generate full functions**.
  - Are building **Python scripts for production use**, rather than just experimenting in notebooks.

- **Use Google Colab with Gemini if you:**

  - Want to **skip local setup** and run code in the cloud.
  - Need **built-in AI assistance** and access to **free GPUs**.
  - Work on **notebooks for research, prototyping, or education**, rather than full production scripts.

VIRTUAL ENVIRONMENTS & CONDA

A virtual environment is an isolated Python environment that allows you to:

✔ Keep projects separate – Each project has its own dependencies.

✔ Avoid conflicts – Install specific package versions without affecting other projects.

✔ Easily reproduce environments – Share your project with others using requirements.txt or environment.yml.

CONDA **Virtual Environment Manager**

- **Manages dependencies automatically** (great for AI & ML projects).
- Works with **Python and non-Python libraries** (e.g., CUDA, TensorFlow, PyTorch).
- Allows **easy switching** between different environments.

CONDA Installation Guide

CONDA Virtual Environment Guide

⚠️ AI CODE ASSISTANT CHALLENGES

💡 **Takeaway : LLMs are great for drafting code, but not always for the latest APIs**. Use them as **helpers**, but always **validate against real documentation**

Why Do LLMs Struggle with the Latest AI APIs - LLMs like **ChatGPT, Claude, and Gemini** are trained on past data and:

❌ **Lack real-time updates** – They may not know about the latest API versions. **BTW** LLM Search != Taining

❌ **Fast-Changing Llibraries –** AI tools like LangChain, LlamaIndex, and AI Agent frameworks evolve rapidly.

❌ **Generate outdated or incorrect code** – LLM-generated solutions may use **deprecated functions or incorrect parameters**, and even sometimes hallucinate functions that don't exist.

⚠️ AI CODE ASSISTANT CHALLENGES

How to Work Around This?

✔ Understand LLM Training - It's extremely likely that an LLM was trained on an older version of a rapidly changing API

✔ Study the API docs - Quickly read the documentation and code examples to ensure you have a basic understanding.

✔ Always check the official docs – It can be quite useful to feed the LLMs the latest API documentation (just the url even!)

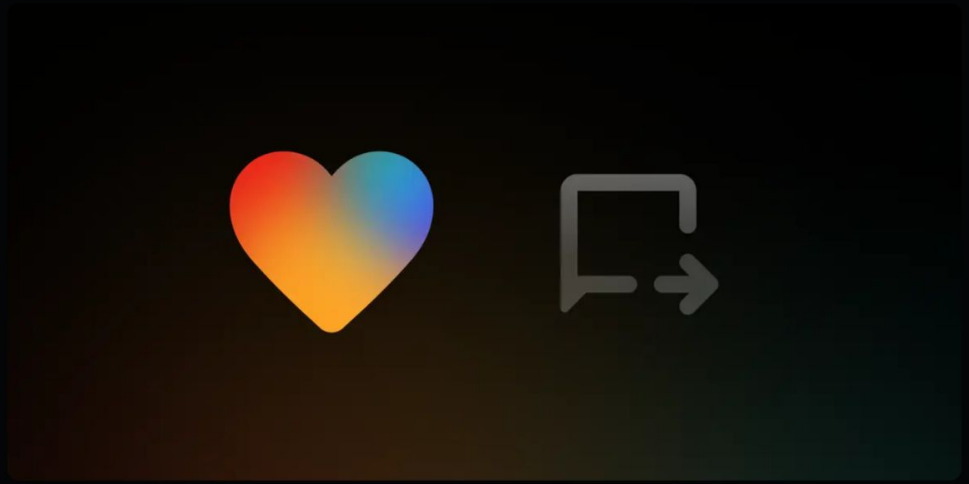
✔ Use GitHub search – Look for recent repo examples to confirm best practices.

✔ Manually debug – If an LLM-generated script fails, read the error messages and adjust accordingly.

✔ Stay updated – Follow LangChain, LlamaIndex, AI agent etc communities, newsletters etc for changes.

https://lovable.dev/blog/2025-01-16-lovable-prompting-handbook

The Lovable Prompting Bible



Author: Stephane at Lovable

17 min read

Share this

Most people think prompting is just typing a request into AI and hoping for the best. **Wrong.**

The difference between getting **a mediocre response** and having AI **build entire workflows for you** comes down to how you prompt. Whether you're a developer or a non-technical user, mastering prompt engineering can help you:

Techniques in Programmatic PE

Dynamic Prompting → Adapting prompts based on real-time inputs.

Self-Correcting Prompts → Using AI to evaluate and improve its own responses.

Function Calling & API Integration → Leveraging external tools to enhance AI capabilities.

Example: Automating Prompt Optimization

Step 1: Store prompt variations in a database.

Step 2: Run automated evaluations using LLM APIs.

Step 3: Score responses for correctness, coherence, and efficiency.

Step 4: Select the best-performing prompts for production use.

LLM & Prompt Tools

Register at aistudio.google.com

Try at aistudio.google.com/prompts/new_chat

Prompt Gallery : ai.google.dev/gemini-api/prompts

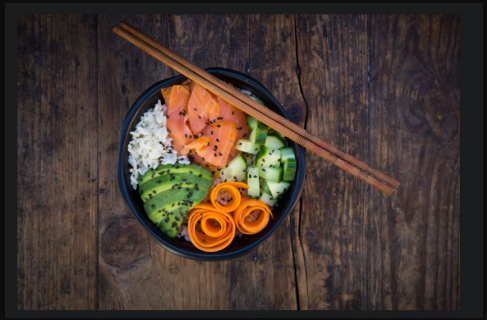
Try at colab.research.google.com

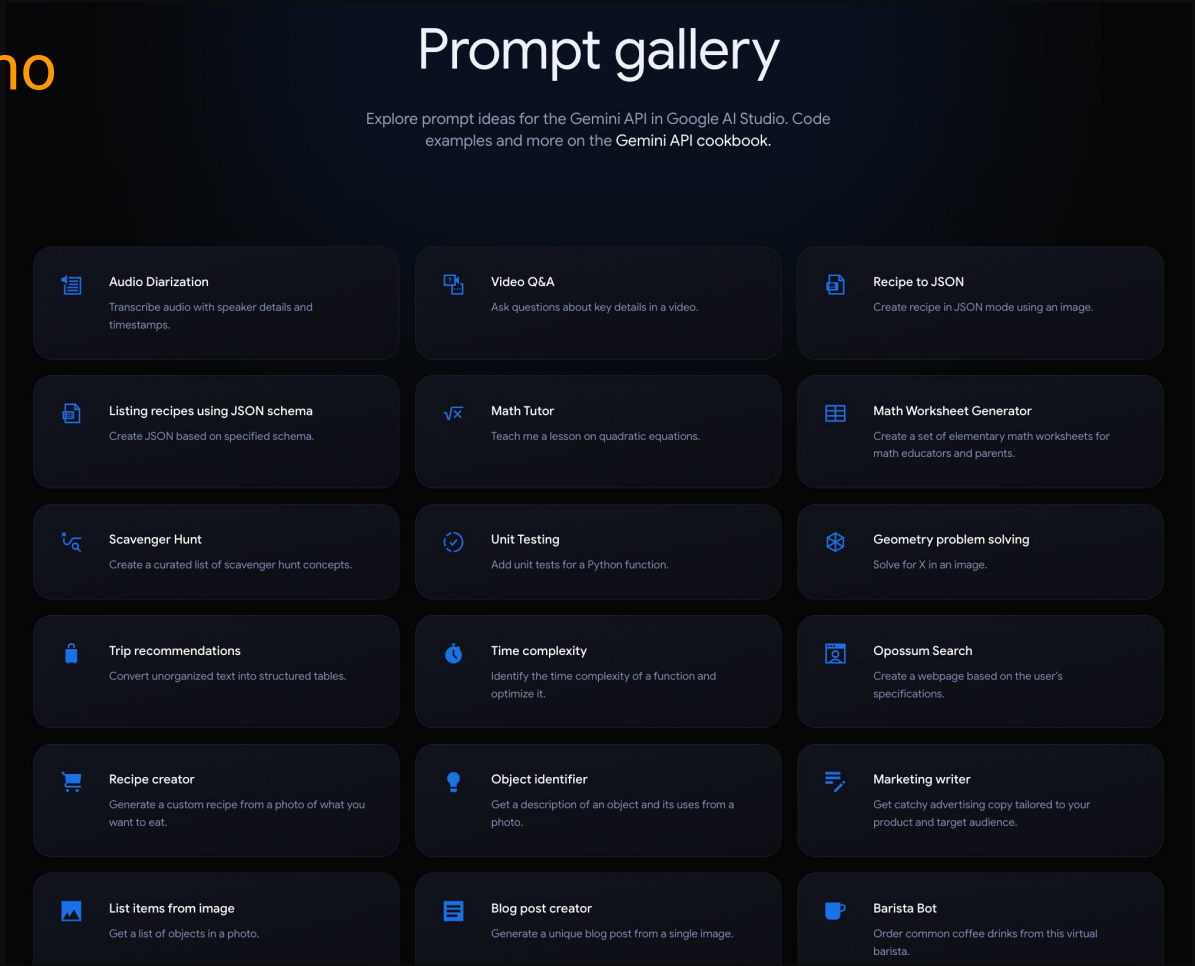
AI Studio Demo

Google Prompt Gallery

https://ai.google.dev/gemini-api/prompts

https://aistudio.google.com/app/gallery





📊 DATA COLLECTION

Why Data Collection & Storage Matters?

- AI and machine learning models rely on high-quality, well-structured data.
- Efficient data storage improves retrieval speed, query efficiency, and scalability.

Relational Databases (SQL) – Structured Data Management

✅ Best For: Storing structured data in a tabular format (rows & columns).

✅ Examples: PostgreSQL, MySQL, SQLite, Microsoft SQL Server.

✅ Sudy: The on-demand - Data Wrangling with SQL Course - and the section on creating your own Database

Vector Databases – AI-Powered Data Storage

✅ Best For: Storing and retrieving high-dimensional embeddings for AI models.

✅ Examples: Pinecone, Weaviate, FAISS, ChromaDB.

✅ Study: Various examples across this course bundle

REMEMBER

Just because you have an

LLM HAMMER

Not everything is a nail

THANK YOU