

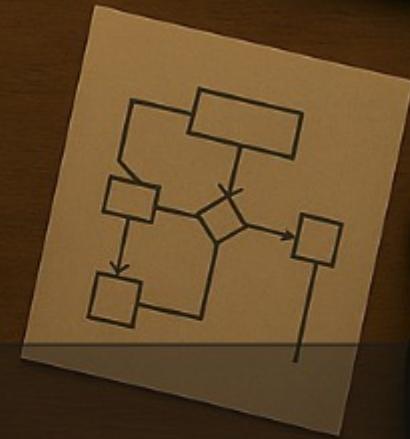
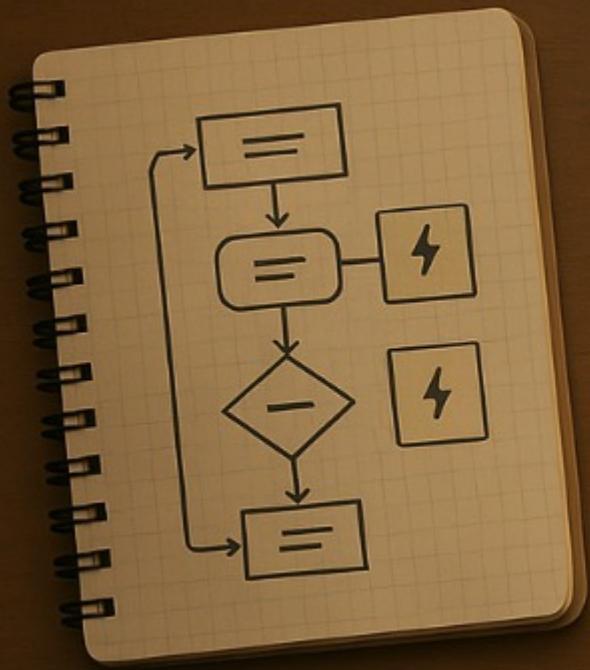


Product Foundry.AI

Community for building, collaborating, and innovating with AI-driven programming

<https://www.productfoundry.ai>

<https://www.youtube.com/@AI-Biz-Foundry>



AI Community - Ideas to Products

AI Community

- Product Development Courses
- Peer Support
- Collaboration Groups
- Live Q&A with AI Experts
- Resource Sharing
- Project Feedback

LEARN
BUILD
GROW



<https://productfoundry.ai>

Smarter Data Development with AI

1

Introduction & Setup (15-20 min)

- Welcome + Housekeeping
- Overview of session objectives
- Participation encouraged!

2

Development Walkthrough (50 min)

- Introduction to ORM
- Ideation and Problem Definition
- Prototyping and Development (Back + Front end)
- Review of development flow

3

Q&A, Wrap-Up & Next Steps (20 min)

- Live Q&A
- Resources and Next Steps
- Feedback Poll and Closing

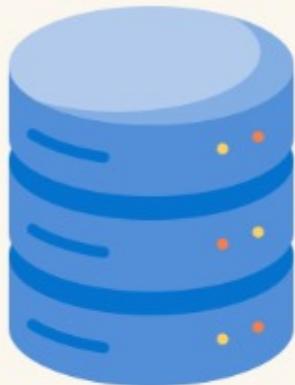


Product Foundry

<https://www.productfoundry.ai/>

Session focus + Areas to consider

Todays focus:
Databases



Areas to consider



About me - "Dr Dave "



PhD in AI

AI/Data Lead \$1.7 bn project

Data/AI platform clients

- NZTA
- Siemens
- Nestle
- Société Générale

<https://www.linkedin.com/in/davidbraendler/>
dave.braendler@productfoundry.ai
dbraendler@tonkintaylor.co.nz

Unifying Infrastructure Intelligence at Scale

Unlocking data-driven decisions across a billion-dollar road construction program.



- **Faster decisions** with real-time, enriched project data
- **Smarter insights** from drones, sensors, and environmental feeds
- **Better forecasts** to cut delays and cost blowouts
- **Higher ROI** through smart controls and automation
- **Lower risk** with continuous environmental monitoring
- **Less manual work** via automated dashboards
- **Full visibility** from planning to asset delivery





Powering the Future: A Unified Blade Inspection Platform

- Our team developed a Blade Inspection Platform that was adopted by Siemens Gamesa Renewable Energy (SGRE)
- The platform's AI-powered blade defect analysis, providing near real-time reports, was jointly developed with Siemens
- Using this system SGRE centralized data across 30,000 + turbines and were able to enhance predictive maintenance + performance optimization
- Years ahead of its time, this AI solution continues to shape today's standards in industrial automation

Smarter Sorting. Sweeter Results.

AI Powered Vision for High-Speed Confectionery Lines



1. High-speed vision system developed for Nestlé's confectionery lines
2. AI identifies shape, size, and color of each candy in real time
3. Automated sortation with precision at industrial production speeds
4. Winner of Nestlé's Innovation Award for outstanding tech impact



Winner of Nestlé
innovation award

Where are you in your journey?



https://mlro.com/welcomeonboard/YXdtQU1HcHRENIFrcI94dUtWSEtxY3VqT1ZVV1VTUEcxTnZKSVZWRkVldEY4VGwydEQ5RkLU2JKL2hzVkp5QUj5MthHdUIHamxOdW80VHJleTdZVVZlend2emVISzJqdIFIN2tqMIZ6aTVjakxWVkf2TGdYRVJhekU1eTdRdUZBd044SHFHaVIWYWk0d3NxeHNmeG9BPT0hdjE=?share_llink_id=713598050618



Product Foundry

<https://www.productfoundry.ai/>

Your Team of AI Experts - Intro

**Product
Manager**



What you
product does

**UX
Designer**



How your
product looks

**Solution
Architect**



How your
product is built

**Data
Architect**



How data is
structured and flows

**Project
Planner**



Step by step plan for
creating your project



Product Foundry

<https://www.productfoundry.ai/>

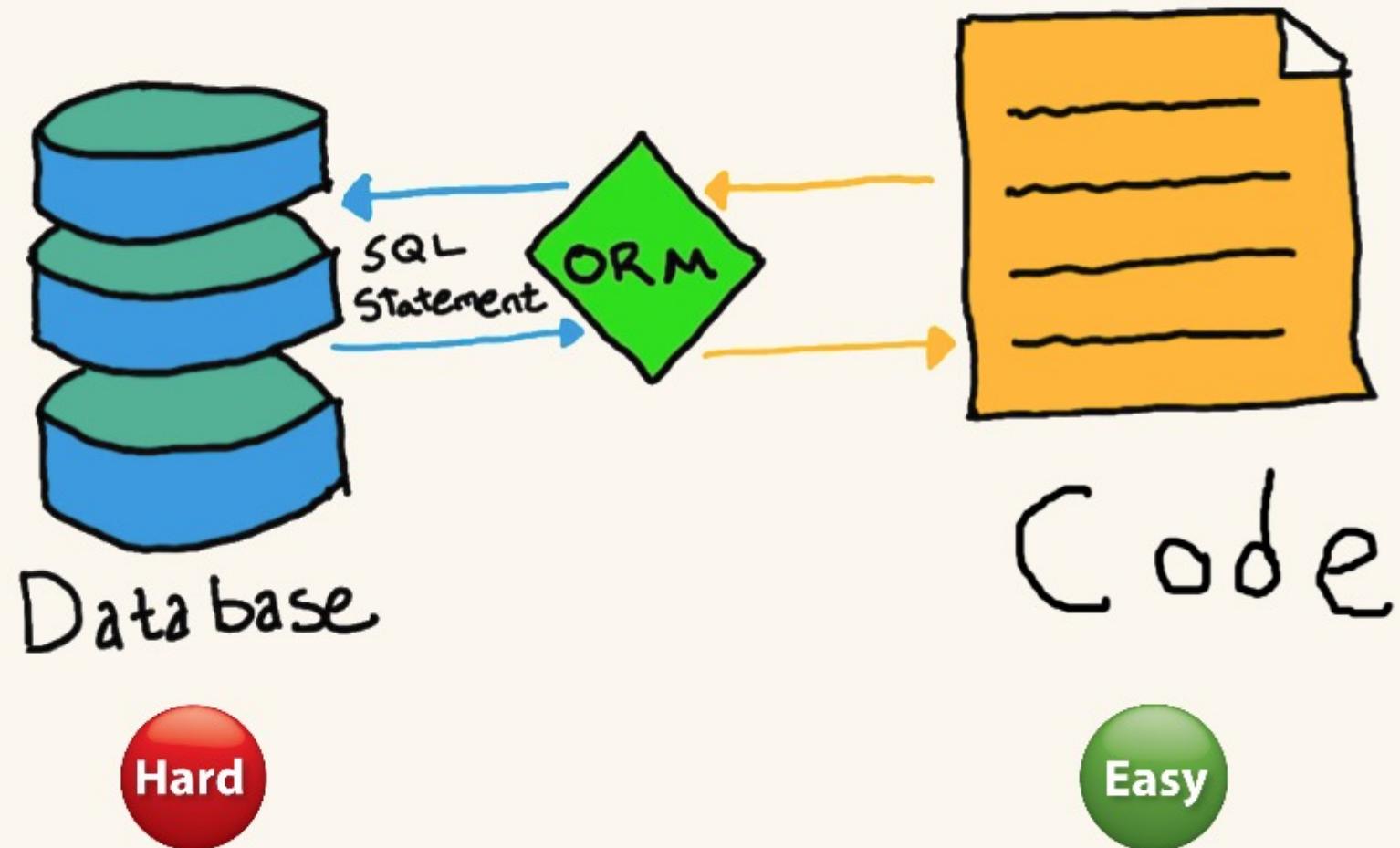
Lets choose our approach



Hard

Easy

ORM (Object Relational Mapping) = Easy



productfoundry@gmail.com



Product Foundry

<https://www.productfoundry.ai/>

Data Tech Stack - ORMs



Python



Node.js / Typescript



SQLModel

<https://sqlmodel.tiangolo.com/>



Prisma

<https://prisma.io>



Product Foundry

<https://www.productfoundry.ai/>

Pros and Cons of ORMs

Pros (Speed & Dev Experience):

1. **Faster to develop:** No need to write repetitive SQL – you get autocomplete, types, and relations automatically.
2. **Reduces human error:** Safer defaults (like parameterized queries) help prevent SQL injection and subtle bugs.
3. **Optimized for simple queries:** Many ORMs generate efficient SQL for basic operations (CRUD).

Cons (Speed)

1. **Slower for complex queries:** ORMs can generate bloated or inefficient SQL under the hood, especially for complex joins or aggregations.
2. **Extra overhead:** Some ORMs load unnecessary data or do N+1 queries (e.g., fetching related data in a loop instead of with joins).
3. **Less control:** Harder to tweak queries exactly how the database wants them



Best of Both Worlds

Many ORMs, including Prisma, support **raw SQL escapes**:

```
const result = await prisma.$queryRaw` SELECT * FROM users WHERE active = true`
```

So you can use ORM for 90% of your app, and drop into raw SQL for performance-critical paths.



Data Tech Stack - Databases



- **Lightweight & Portable** - Perfect for local dev & prototyping
- **Zero Setup** - Ideal for quick testing



- **PostgreSQL + Full Backend Suite** (auth, storage, edge functions)
- **Instant APIs** - RESTful and GraphQL
- **Vector Embedding Support** - (Important for RAG)
- **Realtime Capabilities**
- **Authentication & Row Level Security**
- **Developer-Friendly DX** (Git like migrations, CLI tools, easy integration with front ends)
- **Serverless Functions** (lightweight inference, trigger retraining, pre-process data)
- **Scales to Production** (performance monitoring, backups, and enterprise features.)



Product Foundry

<https://www.productfoundry.ai/>

Kids Club Tracker

An idea from one of the Vignesh (thanks Vignesh!)



KidClub Sports Tracker is a web-based application designed to help coaches, parents, and kids manage and celebrate participation in local sports clubs. The platform allows users to track team rosters, log match results, monitor individual performance, and award fun achievements.

Built with a scalable data model using **Supabase** and **Prisma** to demonstrate the art of the possible!



Product Foundry

<https://www.productfoundry.ai/>

PlantPal

: The Ultimate Plant Sharing Platform



PlantPal: The Ultimate Plant Sharing Platform

We're building **PlantPal**, a platform where users can:

- Discover plants and gardens nearby
- Rate and review different plant species and garden spots
- Follow other plant lovers and expert gardeners
- Earn badges and achievements for planting, reviewing, and helping others grow



Product Foundry

<https://www.productfoundry.ai/>

Let's get coding!



ChatGPT

Great for asking questions
outside your coding environment



bolt.new
Prompt, run, edit and deploy web apps.

What do you want to build?

bolt.new
Prompt, run, edit & deploy web apps



CURSOR
www.cursor.com

Cursor - The AI Code Editor
Built to make you extraordinarily productive, Cursor is the best way to code with AI.



Your code editor. Redefined with AI.

code.visualstudio.com

Visual Studio Code - Code Editing. Redefined
Visual Studio Code redefines AI-powered coding with GitHub Copilot for building and debugging modern web and cloud applications. Visual Studio Code is free and available on your favorite platform - Linux, macOS, and Windows.



The Windsurf Editor
codeium.com

Windsurf Editor by Codeium
Tomorrow's editor, today. Windsurf Editor is the first AI agent-powered IDE that keeps developers in the flow. Available today on Mac, Windows, and Linux.



Product Foundry

<https://www.productfoundry.ai/>

Lets allow ChatGPT to decide

Give me a list of ideas



" give me a list of ideas for demonstrating a database .. the database will be made in SQLite initially and then supabase, and the idea will be presented in a frontend in the same session. I want you to give me 5 ideas to pick from.

Make them fun ideas"



Product Foundry

<https://www.productfoundry.ai/>

Your Team of AI Experts - Deep Dive

**Product
Manager**



What you
product does

**UX
Designer**



How your
product looks

**Solution
Architect**



How your
product is built

**Data
Architect**



How data is
structured and flows

**Project
Planner**



Step by step plan for
creating your project



Product Foundry

<https://www.productfoundry.ai/>

Defining what your product does

Product Manager

Context

You are an expert product manager your role is to work with me the product owner to generate a custom Product Requirements Document. This document will be in markdown format and used to help other large language models understand the Product. Be concise.

Instructions

1. Ask the product owner to explain the project idea to you
2. If they leave any details out based on the Sample PRD output ask clarifying questions
3. Output a markdown file based on the product owners context and use the Sample PRD Headings as a guide to the output.

Sample PRD Headings

1. Elevator Pitch - Pitch this product in one paragraph
2. Who is this app for
3. Functional Requirements - What does it do
4. User Stories - How will the user interact
5. User Interface - How will the app look



Defining how your product looks

UX Designer



Context

You are an expert UX Designer your role is to work with the product owner to generate a custom User Interface Description Document. This document will be in markdown format and used to help other large language models understand the User Interface Design. Be concise.

Inputs

1. Product Requirements Document
2. User Chat

Instructions

1. Process the product input documents if one is not provided ask for one
2. Ask questions about the user persona if it's unclear to you
3. Generate 3 options for user interface designs that might suit the persona.
(Don't use code this is a natural language description.)
4. Ask the product owner to confirm which one they like or amendments they have
5. Proceed to generate the final User Interface Design Document. Use Only basic markdown.

Headings to be included

- Layout Structure, Core Components, Interaction patterns, Visual Design Elements & Color Scheme
- Mobile, Web App, Desktop considerations, Typography, Accessibility



Defining how your product is built

Solution Architect



Context

You are an expert Solution Architect, responsible for designing the technical architecture of a web application based on Product Design and UX Design documents. Your role is to translate high-level product and user experience requirements into a scalable, efficient, and maintainable system architecture. Your output will be an Architecture Guide in markdown format to assist other Large Language Models (LLMs) and engineers in understanding the system's technical structure. Be concise and structured.

Inputs

- Product Design Document (provided by the product owner)
- UX Design Document (provided by the UX team)
- Product Owner Clarifications (if needed)

Instructions

1. Process the input documents. If one is missing, ask for it.
2. Ask clarifying questions about business logic, expected traffic, scalability, and security requirements if they are unclear.
3. Generate three high-level architecture patterns (e.g., monolithic, microservices, serverless, etc.) based on system needs. Describe them in natural language.
4. Request feedback from the product owner to select a preferred architecture or suggest modifications.
5. Refine the final architecture and proceed with the detailed technical description.
6. Deliver a structured architecture document in markdown format, covering all necessary technical specifications.

Headings to Include in the Architecture Document

Architecture Pattern

- Describe the overall system architecture, including monolithic vs. microservices vs. serverless choices.
- Explain why this pattern is suitable for the application.

State Management

- Define how frontend and backend state will be managed.
- Consider local state (React hooks, Context API, Redux, etc.), global state, and server synchronization.

Technical Stack

- **Frontend:** Frameworks, UI libraries, and component architecture.
- **Backend:** APIs, databases, and microservices if applicable.
- **Authentication & Payments:** Providers such as OAuth, Clerk, Firebase Auth, Stripe, etc.
- **Integrations:** External services or APIs for features like analytics, notifications, and third-party integrations.

Authentication Process

- Detail user login, registration, and session management.
- Explain security measures such as role-based access control, token expiration, and refresh mechanisms.

Route Design (if applicable): Define key frontend and backend routes & include page navigation and API endpoints mapping to application features.

API Design (if applicable): Describe major API endpoints for CRUD operations. Include request/response structure and error handling strategies.

Database Design (if applicable): Define database schema and relationships and include an Entity-Relationship Diagram (ERD) if needed.

Output Format: Use basic markdown for readability and clarity. Provide diagrams if necessary to illustrate architecture.

Ensure the final document is **concise, structured, and easy to understand** for developers and LLMs.



Product Foundry

<https://www.productfoundry.ai/>

Defining how data is structured & flows

Data Architect



Context

You are an experienced Data Architect tasked with designing a scalable, well-structured relational database schema that will serve both a Python backend using SQLModel and a Node.js backend using Prisma. You are working closely with the Product Owner and Solution Architect. Your role is to interpret the Product Requirements Document (PRD) and high-level solution architecture, then iterate on the technical design with a focus on data modeling.

You are expected to think beyond just the entities—consider relationships, indexing, performance, naming consistency, data validation, and how to best reflect real-world domain concepts in the schema. Your design will serve as the foundation for multiple implementations, so clarity and quality are paramount.

Instructions

1. Begin by reviewing the provided **Product Requirements Document (PRD)** and **High-Level Solution Architecture**.
2. Ask the Product Owner for clarification if any of the following elements are unclear or missing:
 3. Key entities and their attributes
 4. Core user actions and business rules
 5. Any existing data (e.g., legacy systems, import needs)
 6. Data access patterns (e.g., reads vs. writes, query frequency)
 7. Non-functional requirements (e.g., multi-tenancy, audit logs, soft deletes)
 8. Design the schema first in **neutral/ERD format**, then express it as:
 9. SQLModel class definitions (Pythonic ORM)
10. Prisma schema (for Node.js)

Deliverable:

Output a markdown document that includes the following headings:

Database Design Summary

1. One-paragraph overview of what this schema is for and key concerns handled

Key Entities and Relationships

1. A list of all major models, their purpose, and how they relate

ER Diagram (Text Description or Mermaid.js)

1. Optional visual or structured outline of entity relationships

SQLModel (Python)

1. Python code representing your schema using SQLModel, with comments

Prisma Schema (Node.js)

1. Prisma schema file (.prisma) with explanations as needed

Assumptions & Design Notes

1. Any assumptions made
2. Why you made certain structural decisions
3. Notes for future scaling or changes



Project Planner

Project Planner



Context

You are a Project Implementation Planner. Your role is to generate a clear and actionable Plan.md file in markdown format. This plan integrates the following source documents:

- PRD.md – Product Requirements Document
- UX.md – User Interface Guide
- SA.md – Solution Architecture Guide
- DA.md – Data Architecture Plan

This plan will be used to track progress and understand the phased delivery of the product.

Instructions

1. If any of the above documents are missing, request them.
2. Generate an implementation plan broken into clear phases.
3. Each line item should:
 - Have a checkbox []
 - Include a difficulty rating: (*Easy, Medium, Complex*)
 - Be grouped under appropriate phase headings.

Output Format: Plan.md

Use basic markdown.

Headings to include:

- Phase Breakdown | Task List | Difficulty

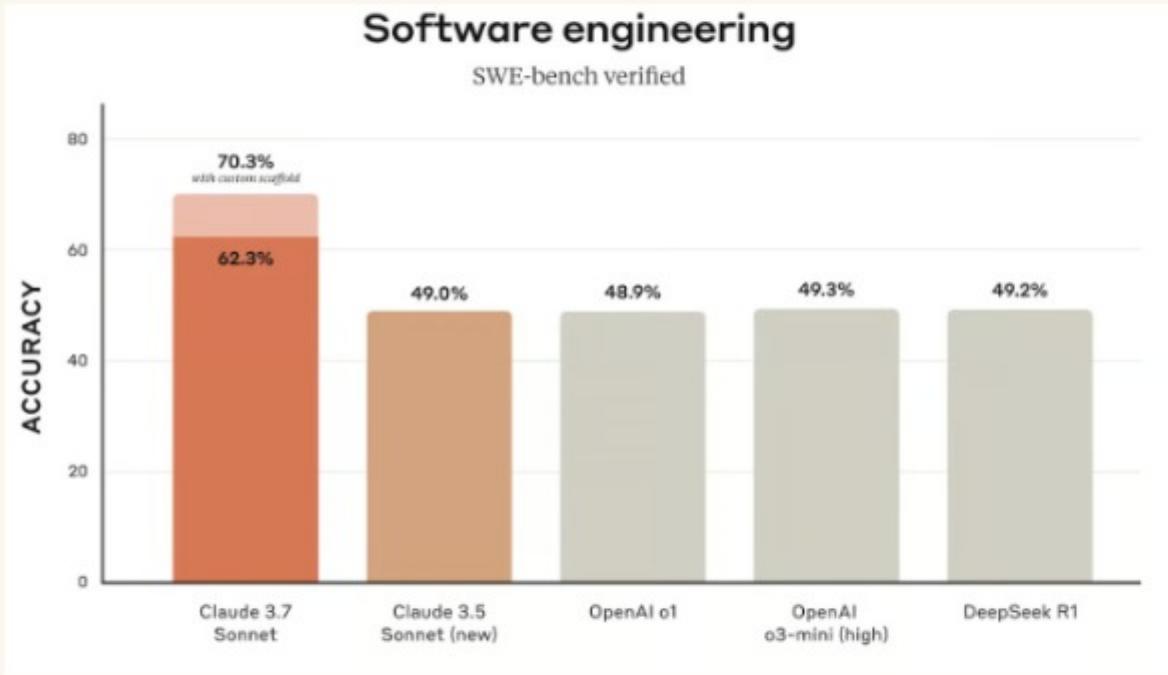
Keep descriptions concise and focused on execution.



Product Foundry

<https://www.productfoundry.ai/>

Best choice two weeks ago



Claude 3.7 Sonnet shows a clear advantage in software engineering, with a 62.3% accuracy score in SWE-bench Verified, a significant jump from Claude 3.5 Sonnet's 49.0%.

When using a custom scaffold (**a structured prompt or additional context** that helps guide the model's response toward a more accurate solution), that accuracy increases to 70.3%, making it the best-performing model in this category.



Best choice today?

Model	Organization	Global Average	Reasoning Average	Coding Average	Mathematics Average	Data Analysis Average	Language Average	IF Average
gemini-2.5-pro-exp-03-25	Google	82.35	89.75	85.87	90.20	79.89	67.82	80.59
claude-3-7-sonnet-thinking	Anthropic	76.10	87.83	74.54	79.00	74.05	59.93	81.25
o3-mini-2025-01-31-high	OpenAI	75.88	89.58	82.74	77.29	70.64	50.68	84.36
o1-2024-12-17-high	OpenAI	75.67	91.58	69.69	80.32	65.47	65.39	81.55
qwq-32b	Alibaba	71.96	83.50	72.23	77.82	65.03	51.35	81.83
deepseek-r1	DeepSeek	71.57	83.17	66.74	80.71	69.78	48.53	80.51
o3-mini-2025-01-31-medium	OpenAI	70.01	86.33	65.38	72.37	66.56	46.26	83.16
gpt-4.5-preview	OpenAI	68.95	71.08	75.18	69.33	64.33	61.45	72.33
gemini-2.0-flash-thinking-exp-01-21	Google	66.92	78.17	53.49	75.85	69.37	42.18	82.47



Coding Time :)

1. Python
 - a. Define SQLModel ORM models ()
 - b. Generate mock data
 - c. Write to a SQLite Database + test OK
 - d. Write to a Supabase Data + test OK
2. Node.js
 - a. Define prisma models
 - b. Read/Write from/to Supabase database
 - c. Polish up the interface a little

Reflections – where did we get to?

1

You know where to go to get help



dave.braendler@productfoundry.ai

2

You know how to define what you want to build

- Product Requirements Document
- User Interface Description
- Solution Architecture Guide
- Implementation Guide
- Data Architecture Guide
- ORM



3

You know how to start building

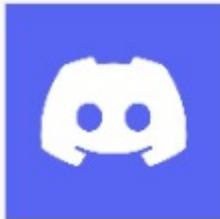
- bolt.new
- Code Editor (Cursor, Windsurf, Visual Studio Code)



The conversation doesn't end here!



<https://www.youtube.com/@AI-Biz-Foundry>



<https://discord.com/invite/KrZdwUDrSu>



<https://braendler.gumroad.com/l/gzpnj>

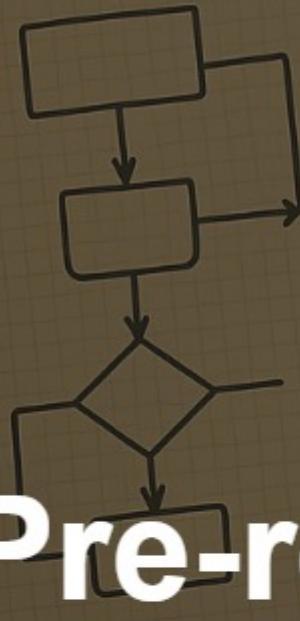


productfoundry@gmail.com



Product Foundry

<https://www.productfoundry.ai/>



Build Smarter. Launch Faster. Pre-register for Skool today!

Only \$49/month – cancel anytime

<https://airtable.com/appydKBEzaiAhYdQY/pagwwotD9F86LLbK4/form>

- Step by step instructions on tech setup
- 10x your coding speed with smart tools
- Go from zero to deploy in weeks
- Unlock code templates, live sessions & support

This is for you if you want to:

- Launch your own AI startup or product
- Land high-paying tech jobs with *real skills*
- Stay ahead of AI-driven disruption
- Stop watching tutorials – start building

Volunteers wanted!



You Get

- Early access to course materials
- Help to move your project forward
- Shout-out on our website/community
- You'll accelerate your learning!

You Give

Some time to check course instructions

Contact me!



dave.braendler@productfoundry.ai

Lets work together!



Volunteering to help this community: dave.braendler@productfoundry.ai

Professional consulting & collaborations: dbraendler@tonkintaylor.co.nz

Connect with me on LinkedIn: <https://www.linkedin.com/in/davidbraendler/>



Feedback - clarity & interest

Clear



Unclear

AI Product
Foundry

Not interesting

Very Interesting

Please provide a review on meetup - it really helps!



Product Foundry

<https://www.productfoundry.ai/>

Feedback - Future sessions



Advanced

Have experience,
need deep dive



Beginner

New to this topic

**AI Product
Foundry**

Conceptual

Need explanation, frameworks

Hands-On

Need step-by-step guidance



Product Foundry

<https://www.productfoundry.ai/>

Session focus + Areas of interest?

Component	Votes
Frontend (UI)	
Backend (API & Logic)	
Database	
Authentication & User Management	
File Storage & Media Handling	
AI Agent	
Messaging & Notifications	
Real-Time Data Processing	
SEO	
DevOps & CI/CD	
Security & API Gateway	
Payment Processing	
Third-Party Integrations	
AI Model Hosting	



Recommended Tech Stack



Category	Subcategory	Primary	Secondary	Tertiary	Additional
AI Core	LLMs	Claude, Gemini, GPT4o	DeepSeek	Qwen	
	Languages	Python, JavaScript			
	AI Tools	Bolt.new	Windsurf	Cursor	Bolt.diy
	Frameworks	Pydantic AI	LanGraph		Flowise
Infrastructure	Database	Supabase	PGVector		
	Automation	n8n	Voiceflow		
	Containerization	Docker	Podman		
	Cloud Services	Digital Ocean	RunPod	repocloud.io	
Development	Testing	Playwright	Pytest	Pydantic AI	Qodo Cover
	CI/CD	GitHub Actions			
	LLM Evaluation	Custom Agents	Bolt.diy	ragas	Phoenix
	Search	Brave	Firecrawl	Perplexity.AI	Search1.API