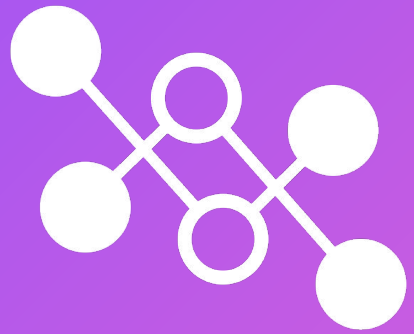
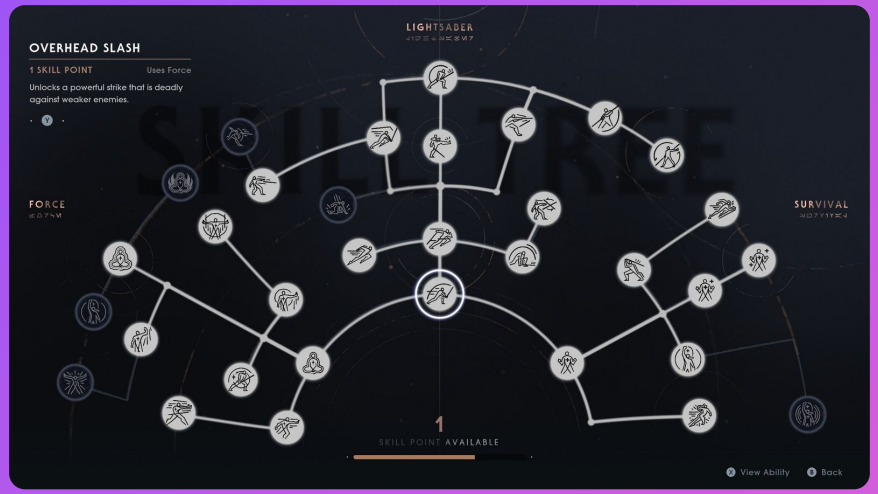


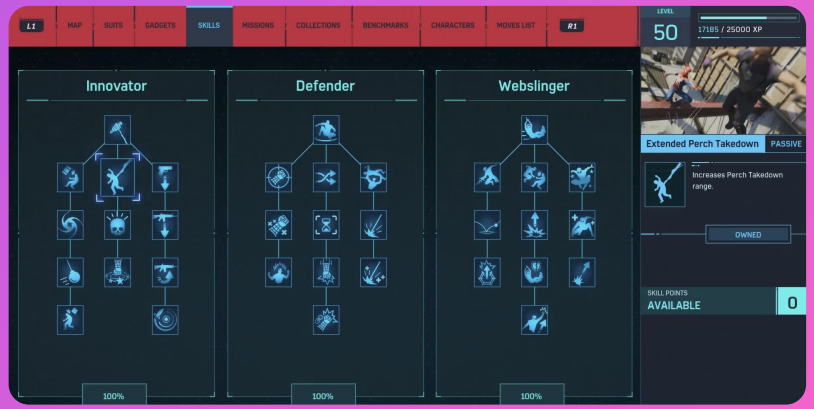
branchly

Inspiration

Inspired by skill trees in video games, which illustrate progression and mastery, we aimed to make self-learning more **engaging and structured**.

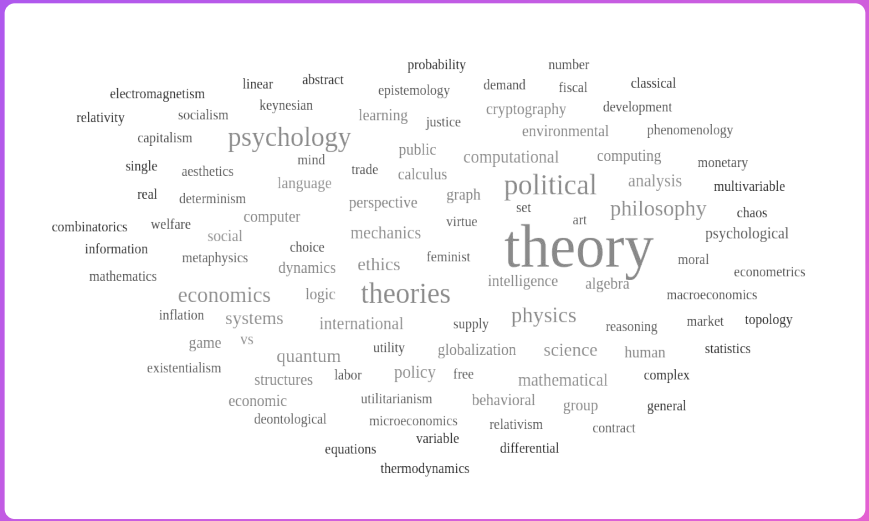


Skill tree in Jedi Fallen Order (source: Rock Paper Shotgun)



Skill tree in Marvel's Spider Man (source: Game Rant)

We recognized there is a lack of tools that offer organized pathways for self-directed learning, especially for **diverse interests and skills.**



Users can learn an infinite number of topics, in diverse fields

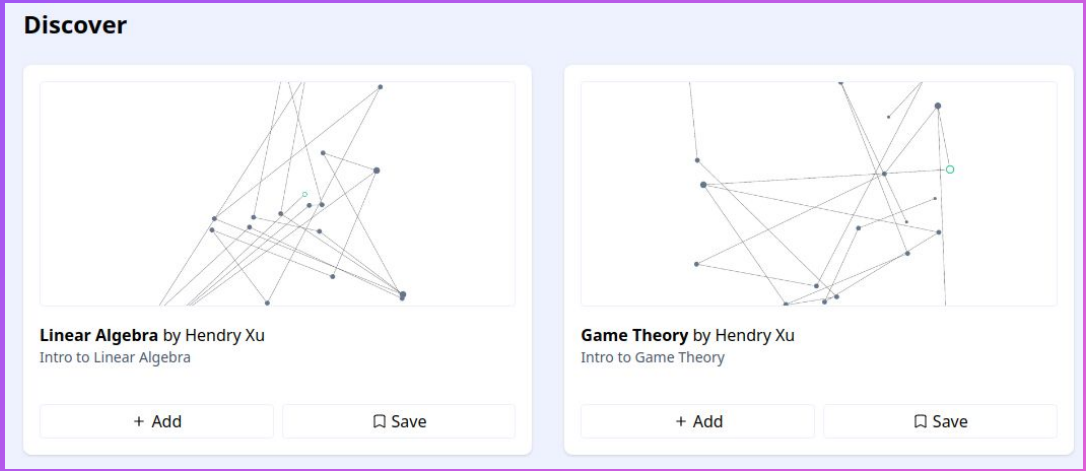
We propose an AI-driven tool that can **guide** users on a pathway to **learning anything**

Motivated by the belief that collective wisdom can drive change, we wanted to create a collaborative platform where **users can share and learn together.**

An example of a "branch" for one specific topic in a user's skill tree

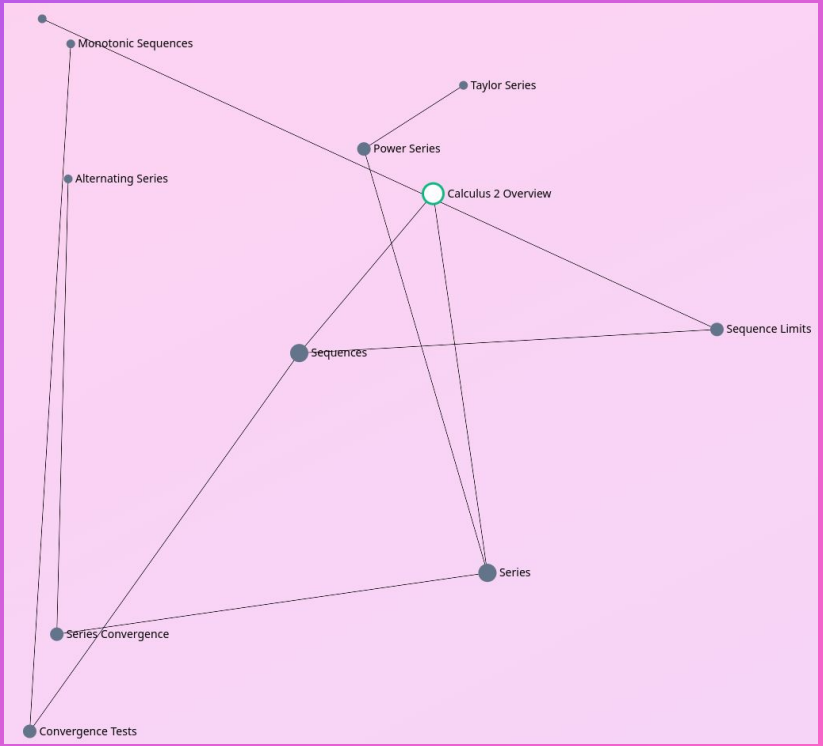
What it does

Custom skill trees: Users develop a custom comprehensive skill tree over time, adding *branches*, or topics that they want to master.



Each *branch* consists of multiple nodes, or lessons.

LLM-generated content: We utilize artificial intelligence to automatically generate skill branches for any topic, allowing users to learn whatever they want.



Content recommendations: Each node uses an AI-driven recommendation engine to offer multiple free online sources to aid in mastering the given node's lesson.

Vector Operations

Vector operations include addition, scalar multiplication, and dot product, which are used to manipulate vectors in a vector space.

Total number of points possible: 142

Related Wikipedia Articles: (Total Gunning Fog Index: 82.90)

Vectorization (86 words, Gunning Fog Index: 46.03)

Vector (mathematics and physics) (1309 words, Gunning Fog Index: 20.87)

Euclidean vector (8733 words, Gunning Fog Index: 16.00)

Extra textbook:

MATLAB Programming/Arrays/Basic vector operations

YouTube Videos: (Total time: 18 minutes)

Vector Operations - Adding and Subtracting Vectors by The Organic Chemistry Tutor (5:31)

Operations on Vectors by Math and Stats Help (3:07)

Vector Operations (2D) by Mathispower4u (9:05)

Introduction to Linear Algebra

Linear algebra is a branch of mathematics that deals with the study of linear equations, vector spaces, linear transformations, and matrices.

Total number of points possible: 95

Related Wikipedia Articles: (Total Gunning Fog Index: 44.92)

Linear algebra (7181 words, Gunning Fog Index: 15.55)

System of linear equations (5156 words, Gunning Fog Index: 15.88)

Rank (linear algebra) (4221 words, Gunning Fog Index: 13.49)

Extra textbook:

Linear Algebra

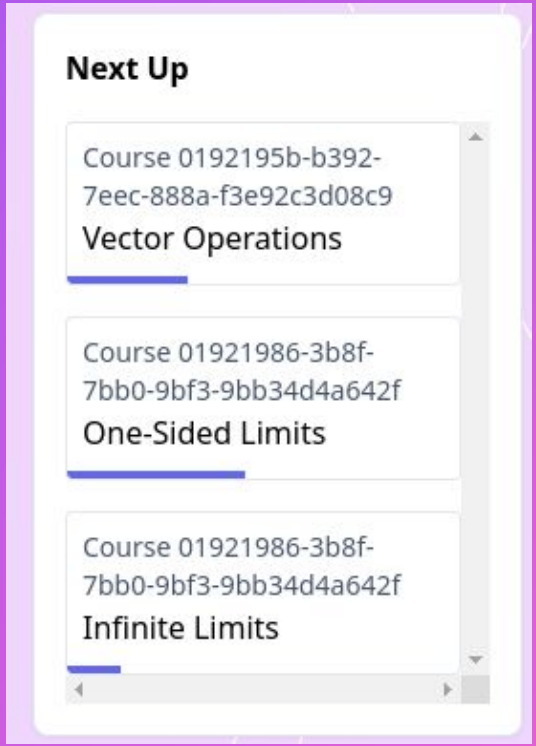
YouTube Videos: (Total time: 28 minutes)

Introduction to Linear Algebra: Systems of Linear Equations by Professor Dave Explains (10:45)

Essence of linear algebra preview by 3Blue1Brown (5:09)

Intro to Matrices by The Organic Chemistry Tutor (11:23)

The skill tree allows users to visually track their mastery of topics in smaller segments.



Tech Stack

Frontend

Tailwind: *Responsive styling for modern user interface*

Next + React: *Efficient frontend framework for fast reloading and smooth rendering*

Tree Logic + Visualization

Graphology: *Easy to use graph data models and algorithms*

Sigma.js: *Graph visualization, rendering, and interactions*

LLM Branch Generation + AI Recommendations

Llama 3.1 + Groq: *LLM prompt engineering for automated branch generation from given topic*

Transformers.js
Vector embedding and recommendation engine

Cloud and Authentication

MongoDB: *Database used to store user skill trees and lesson progress*

PropelAuth:
User Authentication

Challenges

Developing a **from-scratch physics simulation** for enhanced skill tree visualizations.

Making sure UI remained **responsive and efficient** even under high loads (Next JS hydration).

Engineering a consistent and reliable prompt for Llama 3.1 skill branch generation.

Accomplishments

Successfully built a **dynamic, interactive skill tree** that is intuitive for users.

Using AI-driven branch generation allows for a **diverse and highly personalized experience.**

Created a **visually appealing modular interface** with Tailwind CSS, which makes it easy to add new features in the future.

Built a functional **recommendation engine** that can pull content from multiple free sources.

Established a fully working backend with MongoDB Atlas, allowing our webapp to be **hosted entirely on the cloud.**

What we learned

Learnt how to make responsive UI, even under high computational loads.

Experimented for the first time with prompt engineering with LLaMa and machine-driven semantic analysis with our recommender model.

Strengthened understanding of backend architecture, particularly in terms of managing user data with MongoDB.

Future improvements

Implement a more advanced recommendation system - takes into account user preferences over time.

Incorporate assessments to validate user’s understanding of content.

Introduce more social features for the community (expand on the discovery tab).

Add a branch editor (so that users can create custom branches both with and without the help of AI).

Partner with educators (professors and teachers), to develop verified skill branches that users can benefit from.