

Main

Complete AI apps guide 0-100

Content

- Resume prompt
- Generic saas
- Cursor IDE guide
- Ai tutor - future
- Newsletter page
- Directory of apps -
- Link in bio app for developers
- Astrology ai app
- Personal finance - retiring app
- stock analyzer tool
- Top project prompts
- AI projects built with Claude
- Replit agent complete guide
- Link bio app
- Course platform mdx
- Mentorship platform
- AI interviewer

Check the tabs from the left side to access more.

AI Full stack projects

AI Full-Stack Projects

Beginner-Friendly Projects

1. Personal Portfolio Website

Create a modern portfolio to showcase your skills, projects, and experience. Implement animations, interactive UI, and SEO optimization for better visibility.

2. Landing Pages

Develop high-converting landing pages using AI-driven A/B testing and dynamic content personalization to enhance user engagement.

3. Blog Website (Astro)

Leverage Astro.js to build a fast and lightweight blog. Integrate AI for automated content suggestions and summarization.

4. Linktree Clone Website

Develop a simple yet powerful link-in-bio tool that allows users to manage multiple links efficiently with AI-driven analytics.

AI-Powered Projects

5. AI Chatbot for Legal Assistance

Build a chatbot that provides legal information, with a credit-based model for premium features. Use GPT models to generate accurate responses.

6. AI Resume Builder

Create a smart resume generator that uses AI to format, optimize, and suggest improvements based on industry standards.

7. AI-Powered Image Maker

Develop an AI-based tool that generates custom images from text prompts, leveraging APIs like Replicate.

8. Unlimited AI Apps (Replicate)

Create a platform that allows users to generate AI-powered applications with minimal coding using Replicate's API.

Productivity & Utility Apps

9. Task Manager

Design an AI-enhanced task manager with smart scheduling, reminders, and prioritization based on user habits.

10. E-commerce Store with AI Context

Build an online store that personalizes product recommendations using AI and provides automated customer support.

11. Habit Tracker

Develop a smart habit-tracking application that adapts to user behavior and offers AI-driven habit improvement suggestions.

12. Payment Dashboard App (Stripe)

Integrate Stripe to create a secure and user-friendly payment dashboard for managing subscriptions and transactions.

13. Email Newsletter Application

Create an AI-powered newsletter tool that curates and sends personalized content to subscribers based on their interests.

Essential Tools for Development

- **Bolt** – AI-powered tool for rapid prototyping and development.
- **Lovable** – A platform to generate and test UI/UX designs effortlessly.
- **Cursor** – AI-enhanced code editor that helps in writing and debugging code efficiently.

Deployment Platforms

- **Vercel** – Ideal for fast and seamless deployment of full-stack applications.
- **Netlify** – A powerful alternative for deploying and managing frontend applications with serverless functions.

Building a Full-Stack AI Application

Key Features of a Full-Stack AI App

- **Database** – Efficiently store user data and application content.
- **Authentication** – Secure login/logout system using OAuth or JWT.
- **Context & State Management** – Manage user sessions and application state with tools like Redux or Zustand.
- **Payments** – Integrate Stripe or Razorpay for handling transactions securely.

Step-by-Step Guide: AI-Powered E-commerce App

1. Define the App and AI Features

- Implement an AI-powered recommendation engine.

- Use NLP for customer service chatbots.

2. Set Up the Database

- Use Firebase, PostgreSQL, or MongoDB for scalable data storage.

3. Design the UI

- Create a responsive and visually appealing interface.

4. Implement Authentication

- Set up login/logout with user verification (OAuth, email, or phone verification).

5. Develop the AI Features

- Train models for personalized recommendations and chatbots.

6. Verify Locally

- Run tests to ensure all features work correctly.

7. Deploy the App

- Host on **Vercel** or **Netlify** for frontend.
- Use **AWS**, **Firebase**, or **Supabase** for backend.

8. Set Up a Custom Domain

- Register a domain and configure DNS settings for branding and accessibility.

With these projects and strategies, you can build AI-powered full-stack applications that are scalable, user-friendly, and innovative.

1.

Detailed prompt

Detailed prompt

"You are an AI tutor designed to create customized learning paths for users based on their experience level, learning preferences, and topics of interest. A user has provided the following details about themselves:

1. User Profile Details:

- Age: {age}
- Education Level: {education_level}
- Occupation: {occupation}
- Field of Interest: {field_of_interest}
- Preferred Learning Style: {learning_style} (e.g., visual, textual, hands-on)
- Current Level of Understanding in Topic: {understanding_level} (e.g., beginner, intermediate, advanced)
- Key Goals: {learning_goals}
- Available Time per Week for Learning: {learning_time}

2. Topic/Question Provided:

- {topic_or_question}

Based on this information:

1. **Break the main topic/question into a structured outline** with clear subtopics tailored to the user's experience level and learning preferences. Make sure the subtopics are progressive, moving from basic concepts to advanced ones, with examples or exercises where applicable.
2. **Provide a brief summary for each subtopic** to introduce the user to what they will learn in that section.
3. **Suggest appropriate resources or methods** for each subtopic (e.g., articles, videos, practical projects, exercises).
4. Ensure the tone and content are friendly, engaging, and aligned with the user's preferred learning style. Include actionable insights, relatable analogies, or step-by-step explanations where appropriate.
5. Highlight a learning path for the week, indicating how much time should be allocated to each subtopic to match their available weekly time."

Example Response (Output Outline):

User Details Example:

- Age: 25
- Education Level: Undergraduate
- Occupation: Software Engineer
- Field of Interest: Machine Learning

- Preferred Learning Style: Hands-on
- Current Level of Understanding in Topic: Beginner
- Key Goals: Build a basic machine learning model
- Available Time per Week for Learning: 6 hours
- Topic/Question Provided: "Introduction to Machine Learning"

Generated Outline:

- 1. Introduction to Machine Learning**
 - What is Machine Learning? (Definition, history, and real-world examples)
 - Types of Machine Learning (Supervised, Unsupervised, Reinforcement)
 - Hands-on: Simple data classification example using a CSV file.
 - Resource: Introductory YouTube video on ML concepts.
- 2. Core Concepts of Machine Learning**
 - Datasets: What they are and how to handle them.
 - Features and Labels: Explanation and examples.
 - Data Preprocessing: Cleaning and preparing data.
 - Hands-on: Use a sample dataset to preprocess and visualize data.
 - Resource: Online interactive Python tutorial on data preprocessing.
- 3. Basic Algorithms**
 - Linear Regression: Concept, use cases, and equations.
 - Decision Trees: How they work with examples.
 - Hands-on: Build a simple linear regression model using Python's `sklearn`.
 - Resource: Guided tutorial on linear regression.
- 4. Evaluation and Testing**
 - Accuracy, Precision, Recall: Key evaluation metrics.
 - Train-Test Split: Why it is necessary and how to do it.
 - Hands-on: Evaluate the built model using test data.
 - Resource: Article on model evaluation techniques.
- 5. Practical Mini-Project**
 - Build a small ML project (e.g., predicting house prices).
 - Apply concepts learned in previous sections.
 - Test, evaluate, and improve the model.

Resume prompt

Resume prompt

Act as an expert FAANG technical recruiter.

I am going to give you some resume points in the next message

Here are the rules you should follow

For experience section

1/ Add skills to the last line of the experience, so for example - skills: react, programming, communication

For projects

1/ Add skills to the last line of the experience, so for example - skills: react, programming, communication - make it a bullet point

For skills

1/ Add the skills currently provided to you in a good priority order

Use strong action words like

Accomplished Achieved Administered Analyzed Assigned Attained Chaired Consolidated Contracted Coordinated Delegated Developed Directed Earned Evaluated Executed Handled Headed Impacted Improved Increased Led Mastered Orchestrated Organized Oversaw Planned Predicted Prioritized Produced Proved Recommended Regulated Reorganized Reviewed Scheduled Spearheaded Strengthened Supervised Surpassed COMMUNICATION Addressed Arbitrated Arranged Authored Collaborated Convinced Corresponded Delivered Developed Directed Documented Drafted Edited Energized Enlisted Formulated Influenced Interpreted Lectured Liaised Mediated Moderated Negotiated Persuaded Presented Promoted Publicized Reconciled Recruited Reported Rewrote Spoke Suggested Synthesized Translated Verbalized Wrote RESEARCH Clarified Collected Concluded Conducted Constructed Critiqued Derived Determined Diagnosed Discovered Evaluated Examined Extracted Formed Identified Inspected Interpreted Interviewed

Cold email

Act as a senior faang recruiter. I will give you my resume points next year, wait for my next message. Check the example below for context. Write a professional cold email template for reaching out to hiring recruiters about an internship or entry-level position in data science (or similar role based on the person's background). Make it adaptable for different recipients and

highlight relevant skills and experiences in a way that aligns with a company's needs. Draw on key details from the person's resume, such as their current role, projects, technical skills, and any specific tools or technologies they've used. Make the tone friendly yet professional, show enthusiasm for the role, and convey an understanding of the company culture if possible. The email should be concise and include an offer to discuss any upcoming opportunities, with attachments for the resume and portfolio. Keep it short sweet and precise.

Cold email Example

Subject: Application for SDE Role

Dear [Recipient's Name],

I hope you're doing well. My name is xyz, a Master's student in Computer Science at Arizona State University (4.0 GPA), graduating in May 2025. I have prior experience at TikTok and Expedia, specializing in backend systems, scalable architectures, and performance optimization. At TikTok, I improved ad targeting precision by 15% with a tag-based filter system, and at Expedia, I enhanced high-availability payment systems using Java, Kotlin, and GraphQL.

I'm excited about [Company Name] and would love to explore SDE opportunities. Please find my resume attached. Let me know if we can schedule a time to connect.

Best regards,

Email: | Phone:

AI projects built with Claude

AI projects built with Claude

Image comparison tool

<https://claude.site/artifacts/a49309c9-49ec-4888-a790-891894001525>

Rubics cube simulator

<https://claude.site/artifacts/0a4a1e14-85d8-4169-b251-6ceebda740cb>

Bloom filter visualization

<https://claude.site/artifacts/2e673219-d056-44b8-9642-b0bbbee928c54>

Self playing snake game

<https://claude.site/artifacts/5574c1eb-ea43-4e20-9d89-c96b9d4cf1de>

Reddit thread formatter

<https://claude.site/artifacts/6892d6ff-5f2e-4731-bb52-14fd3ec781db>

Bitcoin game

<https://claude.site/artifacts/cf93dea8-1b44-4c44-8ff3-a2eef2a7d8cd>

Music drum pad

<https://claude.site/artifacts/091a4c98-f9fc-4601-ae3d-29e3382cd5a3>

TDEE, Macronutrient, and BMI Calculator

<https://claude.site/artifacts/bab072fe-4873-4a5f-b626-a076341c2d76>

Stock Option Call Analyzer

<https://claude.site/artifacts/cd452263-d93a-4506-94b6-6dcb9681512f>

Flappy bird game

<https://x.com/NickADobos/status/1803837860635627558>

Sudoku solver

<https://claude.site/artifacts/22a509b7-5f6a-478a-b56c-298898c3cba6>

Python interpreter

<https://claude.site/artifacts/3fdd108a-4937-4ca6-b2f4-3cf370fec57c>

Drum synth pad

<https://claude.site/artifacts/4a0693bb-db93-4e81-8342-fd0921ae49b5>

Diff checker

<https://claude.site/artifacts/020cf820-7c4b-406b-aeb5-e876337cd6af>

Double pendulum simulator

<https://claude.site/artifacts/33451efb-3ddd-496e-95f9-3cc08c6c72de>

Generic saas

Making an AI SaaS with Bolt.new

Step 1: Build the SaaS Platform

Start by creating a SaaS application that provides AI-powered chat capabilities. Users can interact with the app for free up to 5 messages before authentication is required.

Step 2: Implement Authentication After 5 Free Chats

- Allow users to send up to 5 free messages without logging in.
- After the limit is reached, prompt the user to sign in.
- Integrate **Supabase** for authentication and store user details securely.
- Keep authentication **only on the first page**, away from other sections.

Step 3: Sidebar with Additional Features

Ensure the sidebar remains **always open** and shifts the UI to the right. Add the following options:

- **Osho Books**
- **Osho YouTube Videos**
- **Osho Instagram Clips**
- **Osho Learnings (Blogs)**
- **User Profile** (Shows email, purchased items, and chat history)

Step 4: Fix Authentication Import Error

Resolve the issue by ensuring the correct file structure:

```
import Auth from "../auth/Auth"; // Adjusted import path
```

Step 5: Save Chats in Database

Once a user logs in, their chat history should be stored in the database, allowing them to access past conversations.

Step 6: Fix Chat Refresh Bug

Prevent new chat creation on page refresh by maintaining session data:

- Use local storage or a database to persist the chat history.
- Reload chats when the user revisits the page.

Step 7: AI Credits & Payments

Buying AI Credits

- Implement a **credit-based system** where each AI response costs **1 credit**.
- Create a **purchase credits page** with a **pricing table**.
- Integrate **Stripe** for secure payments.

Stripe Integration

- When a user clicks "Purchase Credits," process payment via **Stripe**.
- After successful payment, **add credits to the user's account**.

Step 8: OpenAI Integration

- Integrate OpenAI API to generate responses for user queries.
- Each AI-generated response should **consume 1 credit**.
- **Fix OpenAI import error:** Ensure correct import in `src/lib/openai.ts`.

```
import { Configuration, OpenAIApi } from "openai";
```

- Show a **loading circle** before returning an answer.

Step 9: Improve Sign-In Popup

- Add a **close icon** to the sign-in popup so users can dismiss it.
- Automatically **close the popup** once the user signs in.
- **Fix the close button functionality** if it's not working.

Step 10: Enhance User Profile Section

- Display **personal details** and **chat history** in the profile section.
- Add the **profile link** to the sidebar for easy navigation.

This structured plan ensures a seamless user experience while incorporating AI and monetization effectively.

Cursor IDE guide

Complete Cursor IDE guide

Get the complete V0.dev + Claude + Cursor coding course -

Table of contents

[Go through the official docs](#)

[This is how the UI looks](#)

[Click Cntrl + L to open the chat box](#)

[Change the model to whatever you want to use](#)

[Migrate from VSCode](#)

[Awesome tips and tricks](#)

[1/ Hover over the errors](#)

[Use the predefined rules from here](#)

[Use Cursor + custom prompts to build features instantly.](#)

[Combine with V0 and copy paste](#)

[Subscription for pro and API](#)



Andrej Karpathy ✓
@karpathy

...

Programming is changing so fast... I'm trying VS Code Cursor + Sonnet 3.5 instead of GitHub Copilot again and I think it's now a net win. Just empirically, over the last few days most of my "programming" is now writing English (prompting and then reviewing and editing the generated diffs), and doing a bit of "half-coding" where you write the first chunk of the code you'd like, maybe comment it a bit so the LLM knows what the plan is, and then tab tab tab through completions. Sometimes you get a 100-line diff to your code that nails it, which could have taken 10+ minutes before.

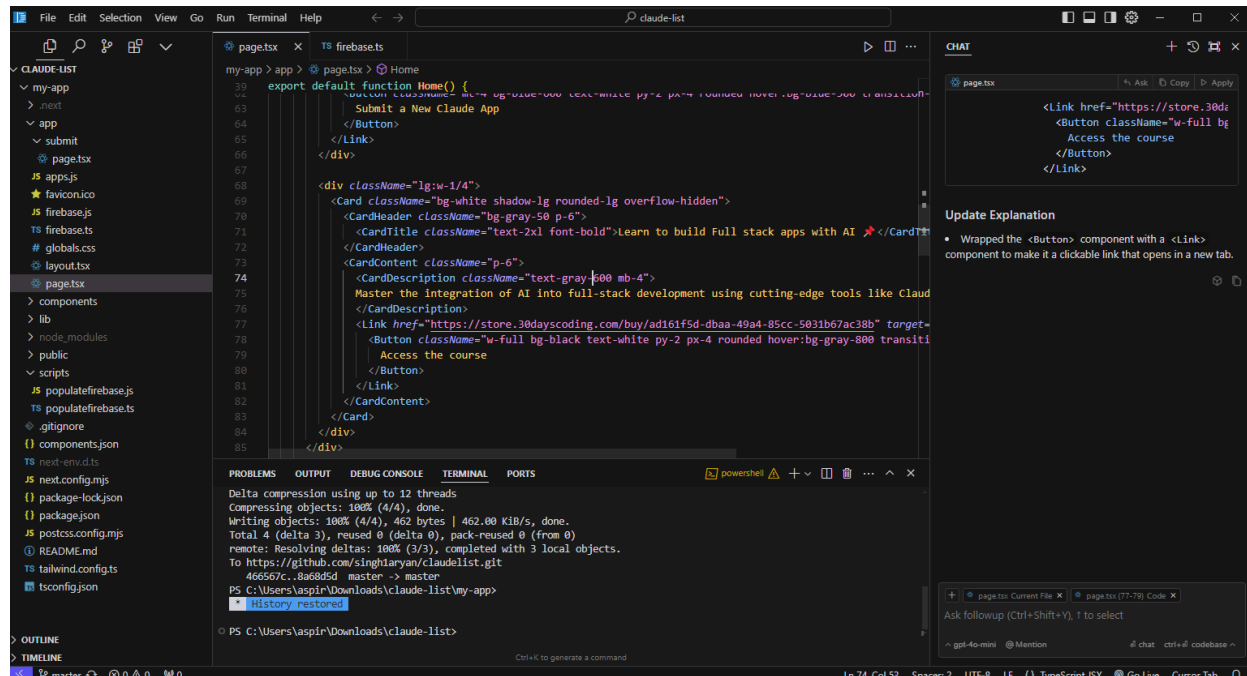
I still don't think I got sufficiently used to all the features. It's a bit like learning to code all over again but I basically can't imagine going back to "unassisted" coding at this point, which was the only possibility just ~3 years ago.

Go through the official docs

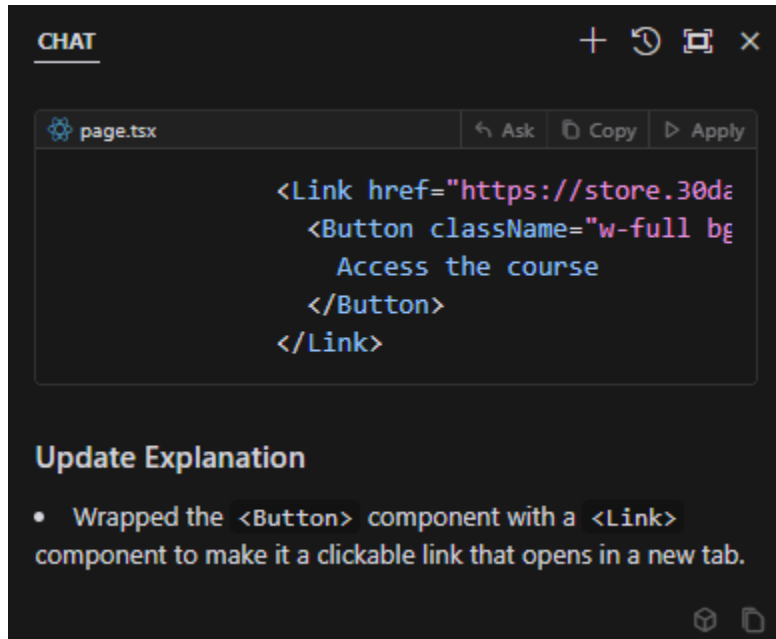
<https://www.cursor.com/features>

<https://docs.cursor.com/tab/overview>

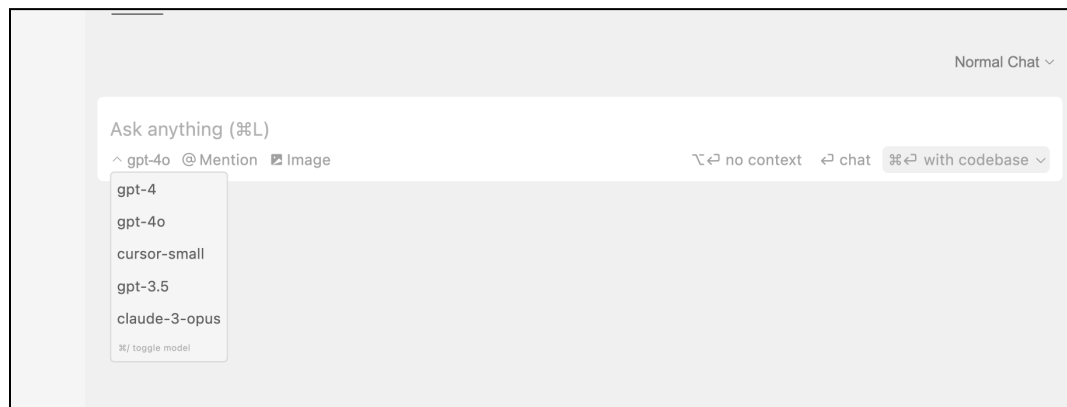
This is how the UI looks



Click Cntrl + L to open the chat box



Change the model to whatever you want to use



Migrate from VSCode

<https://docs.cursor.com/get-started/migrate-from-vscode>

Awesome tips and tricks

1/ Hover over the errors

Inside the debug panel for direct debuggin


```
PS C:\Users\aspir\Downloads\claude-1: Add to Chat Ctrl+Shift+L Add to Composer Ctrl+I
npm ERR! code E404
npm ERR! 404 Not Found - GET https://registry.npmjs.org/asdga - Not found
npm ERR! 404
npm ERR! 404 'asdga@*' is not in this registry.
npm ERR! 404
npm ERR! 404 Note that you can also install from a
npm ERR! 404 tarball, folder, http url, or git url.
```

Use the predefined rules from here

<https://cursor.directory/>



Use Cursor + custom prompts to build features instantly.

<https://x.com/mckaywrigley/status/1828500011480686865>

Combine with V0 and copy paste

<https://v0.dev/chat/5VRPdMd90Zt>

Steps

- Create something in V0
- Copy paste in cursor
- Make sure Shadcn and tailwind is integrated
- UI development becomes 10x easier

Connect replit with Cursor

<https://youtu.be/xrHagI5CdE0>

Follow the amazing people

<https://x.com/mckaywrigley> .

<https://x.com/pontusab>

Subscription for pro and API

Subscription Tiers

Cursor offers multiple subscription tiers to fit your needs.

— Hobby

- 14 days **Pro trial**
- 50 slow `premium` model uses
- 200 `cursor-small` uses
- 2000 **completions** uses

— Pro

- 500 fast `premium` model uses per month
- Unlimited slow `premium` model uses
- Unlimited `cursor-small` uses
- Unlimited **completions** completions
- 10 `Claude Opus` uses per month

Ai tutor - future

Ai tutor - future

- i want to make an ai tutor that when given a topic breaks the subject down into a variety of subtopics based on experience level. So first we want to understand the user with a form and then make those mappings for them. Lets start with a form that takes all the essential user details to understand their level of understanding and subjects they want to learn
- Make a nice ui where when prompted a topic or question, we use open ai api call to make it a detailed prompt first, then we use the detailed prompt to give a very detailed outline of the topics based on user profile and use that in the ui

Newsletter page

Newsletter page

- i want to make a newsletter for aryan singh and talk about startups tech ai content and my learnings as an entrepreneur building multiple brands
- Make a simple ui for the landing page newsletter with a form - email input, an input to ask for persons interest. connect the form with resend api and make a simple welcome email to send as the first email
- Then lets make a workflow for every new user - we want to send one welcome email, then couple of emails for free resources, then for some youtube videos, then sell something. We want to mix n match emails to give a lot of free value and sell stuff so make something for that

Directory of apps -

Directory of apps -

- I want to show micro ai saas apps that people make so make a nice page for this with title filters and all the apps below in a grid. Each app should have title, description, tech stack, revenue, launch date, founder, growth metric, and more - all the metrics that a startup should have. Make a db and put 3 dummy apps and show in the grid
- Add a navbar with link to our blog and a button for submitting a new app. Make a simple form on a new page that submits to the database. Connect this to the db
- Add a sign in form but make it more detailed for taking some user details - like where they work, years of experience, social followers, projects made, social links and more - for a founders profile
- Update the projects ui in grid to show founder name and the link that takes it to their profile page.

Link in bio app for developers

Link in bio app for everyone

<https://inspiring-swan-78173e.netlify.app/dashboard>

Complex example

Create a professional "Link in Bio" landing page that serves as your digital hub, following these detailed specifications:

Design Requirements:

- Create a mobile-first, responsive layout that scales seamlessly across all devices
- Implement a clean, minimalist design that loads in under 2 seconds
- Match your personal brand's color palette and typography

Essential Components:

1. Header Section:

- High-resolution profile photo (400x400px recommended)
- Concise bio (maximum 150 characters)
- Custom username/handle display

2. Social Media Integration:

- Priority placement for your most active platforms
- Branded social icons with hover effects

- Real-time follower counts (optional)
- Direct deep links to profiles

3. Content Showcase:

- Pinned/featured content section
- Dynamic link cards with thumbnails
- Content categorization system
- Link click tracking implementation

4. Customization Features:

- Theme selection (light/dark mode)
- Custom CSS variables for brand colors
- Font family options
- Layout arrangement flexibility

Ask any clarifying questions

Make a link in bio app for creators
where they can add links, products, new videos, reel

Deploy this application.

create a db of all the creators and tell them to sign in the app
they should be prompted to get a username and check if that username exists in our database
then every user should have a unique link to ourapp/[username] so make the paths for that dynamically

i cannot see the links in a list after adding a new one. pls fix it and store it in the db for the current authenticated user

add a new link type for coding projects which can be used for developers
it should include github link, public url, twitter post, etc

add 2 new types
1. experience for developers
company, dates attended, skills learnt,
2. tech stack - js, react, nextjs, etc

now update the ui for showing experience, projects, etc in a better way in the page
education, then experience, then projects, then skills

Astrology ai app

Astrology ai app

- Give me the 5 most important astrology and numerology things people believe in
- Make a detailed astrology algorithm
- Make a simple form for the user that takes all their values
- Make a simple open ai api call with a prompt that returns astrology information based on that value and add a loading state while loading
- Now add a sign in auth before they can access the app
- Make a landing page for ai astrologer and add the sign in flow there itself
- Allow the user only 3 requests per account
- Prompt the user to pay after the 3 requests to purchase more credits

Personal finance - retiring app

Personal finance - retiring app

- i want to make a retirement app that helps track and record my networth. I want to see a simple graph and charts to track different styles of networth, a form to add a new investment
- I want to save this info so add a db and add a simple login for this info
- Add some prediction segment where i can predict the net worth based on returns. Be creative and make a nice looking ui for this

stock analyzer tool

Make a complete stock analyzer tool with AI

RockyInvestors.com

i want to make a retirement app that helps track and record my networth. I want to see a simple graph and charts to track different styles of networth, a form to add a new investment

add another field for the asset value so we know how much we bought at what price

what are the most common things about a stock that we need to see before buying the stock. add that table as well and connect it with a finance api

now add some more metrics to the stock table when fetching

cash in hand, dividends, etc

anything and everything that helps in deciding

add a column for a metric which says good or bad

now with popular stocks in a chip
add a very detailed section there for categories and stocks
inside those categories
then include index funds in an additional line as well

now add a graph when analyzing which shows the price in
the last 5 years

Top project prompts

Top project prompts

Beginner Prompts

1. **User Authentication System**

Build a simple user authentication system using **Supabase** for email and password login. Implement features for signup, login, and logout. Add functionality to reset passwords via email.

2. **Basic To-Do App**

Create a full-stack to-do list application using **React (frontend)** and **Firebase (backend)**. Users should be able to add, update, delete, and mark tasks as complete.

3. **Simple Blog**

Develop a blog application where users can create, edit, and delete posts. Use **Node.js** with **Express** for the backend, **MongoDB** for the database, and **HTML/CSS** for the frontend.

Intermediate Prompts

1. **Sports Community App**

Develop a sports community app using **Supabase**.

- Include user roles: **manager**, **coach**, and **student**.
- Implement role-based access control so managers can add team information, coaches can schedule practice sessions, and students can view their schedules.

2. **E-commerce Shopping Cart**

Build a shopping cart app using **React**, **Node.js**, **Express**, and **MongoDB**.

- Display products on a dashboard.
- Allow users to add items to the cart, update quantities, and see the total cost.
- Store cart data in a session or database for persistence.

3. **Recipe Manager for Restaurants**

Create a restaurant management app with:

- User and manager roles using **Supabase auth**.
 - A form for the manager to add new recipes.
 - A user interface for customers to view recipes and add items to a shopping cart.
 - A dashboard where managers can view orders placed by users.
-

Advanced Prompts

1. Collaborative Task Manager

Build a task manager with features like:

- Real-time updates using **WebSockets**.
- User roles: Admin, Manager, Team Member.
- Integration with **Supabase** for real-time updates and user authentication.

2. Social Media App for Athletes

Develop a social media app for athletes using **Next.js** and **Firebase**.

- Features include profiles, photo uploads, post likes, and comments.
- Implement infinite scrolling for posts and real-time notifications using **Firebase Cloud Messaging**.

3. Restaurant Analytics Dashboard

Extend the restaurant management app with:

- Analytics for order trends, most popular dishes, and revenue generated.
- Data visualization using **Chart.js** or **D3.js**.
- Integration of a real-time notification system for managers when new orders are placed.

4. E-commerce Recommendation System

Enhance the e-commerce app with:

- A recommendation system using **machine learning** models hosted on **AWS Lambda**.
- Implement personalized product suggestions based on users' browsing and purchase history.

Additional Project Ideas

- **Online Course Platform:** Allow instructors to upload courses, and students to enroll and take lessons. Include a discussion forum.
- **Event Management System:** Create a platform for users to create and RSVP to events. Add admin control for event approvals.
- **AI Chatbot Integration:** Integrate an AI chatbot into any app using **OpenAI GPT APIs** to assist users with queries.

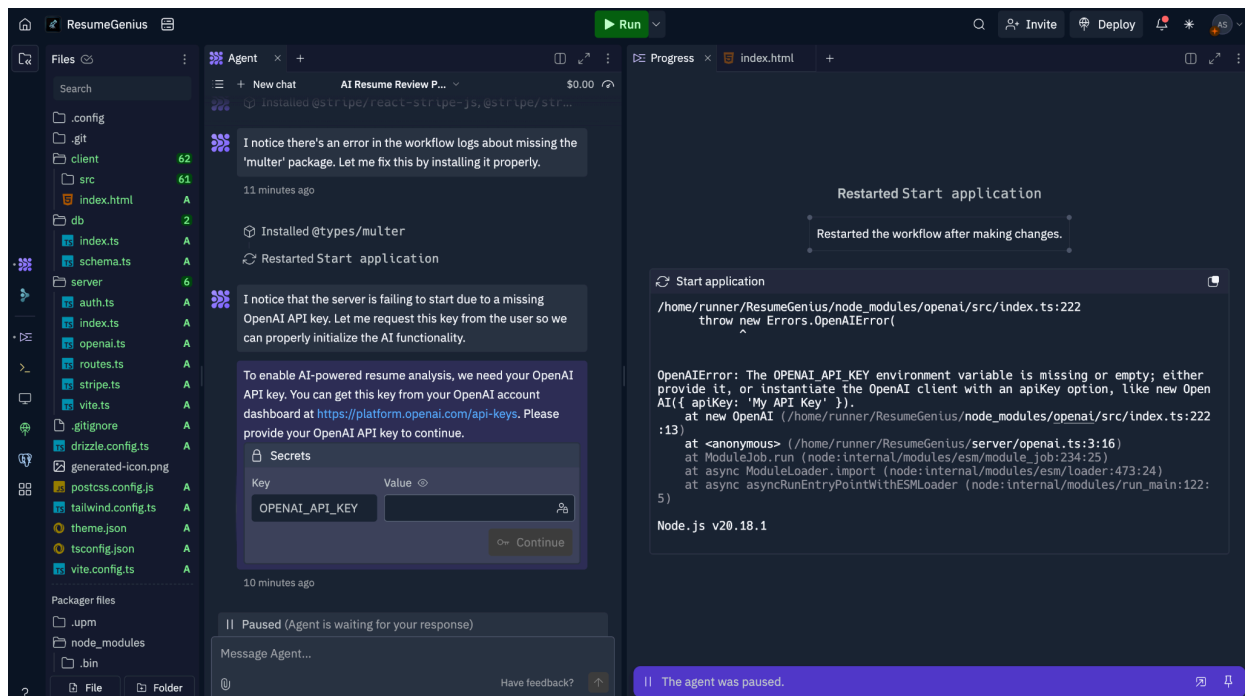
These prompts will not only teach you the core technologies but also provide practical experience in integrating advanced features like authentication, real-time updates, and analytics into your projects.

Replit agent complete guide

Replit agent complete guide

Sample prompt

- i want to make a newsletter for aryan singh and talk about startups tech ai content and my learnings as an entrepreneur building multiple brands. Make a simple ui for the landing page newsletter with a form - email input, an input to ask for persons interest.



Link bio app

Link bio app

Prompts

make a link in bio like linktree website with a phone view on the right and a link view on the left

define a nice data model for it so a user can have multiple link-in-bio

We want 3 tab structure with links, products, for-brands

user can have any number of links - link, open in new tab or not

they can have products that they are selling - ebooks, webinars with - price, new or not, category, updated date, etc

For the brands - we will discuss it more in the next step, but define the model for that too

Have a way to add a new link, product, etc

We also want to be able to add headings and text between the links and products. Add that functionality as well

Make the links editable - so the user can click on a link and edit it - and then save it.

Make a nice landing page and move this current page to /home and make it auth protected - only allow and load the user's data once logged in

make the links draggable one above each other and also the feature to delete links. We want to store the links in a way that we see the same order when we reload.

We want the user to be able to publish this on our website - creatorbiolink.com - with a unique link with a username like creatorbiolink.com/aryan . Build everything we need for this.

Show the personal info above the links like photo, social icons, and a way to add and edit those. Add this to the user profile db and store it

add a navbar to the app and add publishing / sharing options there

Make a new page for themes where we have background color, link colors, font, text color, and more. We want the user to be able to select anything for the ui and make it customizable for them

The finished website version for the user should be similar to the phone view only – it should just be the tabs and content inside it when we view it at creatorbiolink.com/username

Profile and cover photos

On the theme page, Instead of adding color codes - give me a color palette to choose from so that it is easier for the user. Update the save button so the new changes are saved in db for the right user.

Course platform mdx

Course platform mdx

we want to make a data analyst platform that has a complete course to become a data analyst
make a nice ui with astro for mdx blogs
add a sidebar for the course material
and add some sample blogs for it

now include a right sidenav bar which shows the table of content in each blog

now include codemirror for running code under each blog - we want to include excercises for each blog

store this globally so we can change the questions or details under each topic

and keep it flexible and dynamic

Mentorship platform

Mentorship platform

Key Features (MVP)

a) User Authentication for mentors and normal users

- Login/Signup via email and password
- User model different for mentors and users

b) User Dashboard

- View list of mentors
- Filter mentors by category, expertise, rating
- View mentor profiles and services
- Book mentorship sessions
- Payment processing

c) Mentor Dashboard

- Profile setup (bio, skills, experience, availability, pricing)
- Add/edit services (1:1 session, group session, consulting, etc.)
- View booked sessions and earnings
- Accept/Reject bookings
- Messaging system with users

add sample sessions section for the mentors where the user can see videos from the mentor - simple youtube video

add a simple testimonial section for the mentor so the new user can see

the user should be able to select the times - make it better looking and give the ability for the user to select it

When the payment is done, we want to send an email with the times, google meet link, and the mentor details in a nice email. Implement that with resend.

AI interviewer

AI interviewer - prompts

Create an AI-powered mock interview system with the following specifications:

Design a complete web application that enables users to:

1. Technical Requirements:

- Create a video interview simulation system using webcam integration
- Implement OpenAI's Whisper API for speech-to-text transcription
- Design responsive frontend interfaces
- Establish secure data storage and retrieval

2. Core Features:

- User authentication and profile management
- Interview session creation and management
- Real-time video recording
- Automated transcription processing
- Performance analysis and feedback generation

3. Data Model Structure:

Users:

- id (UUID)
- email
- password (hashed)
- name
- created_at
- profile_settings (JSON)

InterviewSessions:

- id (UUID)
- user_id (foreign key)
- interview_type
- status (pending/completed)
- created_at
- completed_at
- duration
- feedback_summary

Questions:

- id (UUID)
- category
- difficulty_level
- question_text

- expected_keywords
- follow_up_questions (JSON)

Responses:

- id (UUID)
- session_id (foreign key)
- question_id (foreign key)
- video_url
- transcription_text
- confidence_score
- analysis_results (JSON)
- created_at

4. User Flow:

a. Landing Page:

- Login/Register
- Overview of features
- Sample interviews

b. Dashboard:

- Previous interview sessions
- Start new interview
- Progress metrics
- Skill development recommendations

c. Interview Setup:

- Select interview type
- Choose duration
- Set difficulty level
- Configure specific focus areas

d. Interview Session:

- Countdown timer
- Question display
- Video recording interface
- Progress indicator

e. Results Page:

- Transcribed responses
- Performance analysis
- Keyword matching
- Improvement suggestions
- Option to save or share results

Please provide implementation details for this comprehensive mock interview system, including frontend frameworks, backend architecture, and API integrations.

i see a loading screen after signing in
also fix the button in navbar to show sign in or sign out

Setup Your Interview

Customize your interview session to match your needs and goals.

Select Interview Type



Technical Interview

Practice coding problems and system design questions



Behavioral Interview

Focus on soft skills and past experiences



General Interview

Mix of technical and behavioral questions

Difficulty Level

Beginner

Intermediate

Advanced

Interview Duration

30 minutes

we want to transcribe the video and then show the transcript on the summary page
we can use whisper api from open ai for this

Optional

- We will transcribe the video after recording
- So we need to store the video somewhere
- Then use the audio or video file to transcribe using whisper open ai
- Then we will go to the results page

XX -> Add a button which transcribes the video and shows it to us.

add the transcribe button the results page below the video so we can click and see the transcript

and then use this transcript to summarize using a simple open ai prompt

add a nice ui below the video and loading screens when the text is being transcribed

Add a button which shows the result summary of how good the interview was.
Use open ai api to run the following prompt with the transcript

...

Summarize the text and give me 5 important pointers

Was the answer clear

Was the answer nicely presented

...

We want to implement a payment system where the user has 10 credits initially
Each interview will cost the user 1 credit
Make a pricing page where they can buy 25 credits for \$10, 100 credits for \$20
Use stripe for this and implement it end to end

Steps for payment

1. Api key - payment - stripe, razorpay, cashfree, dodo payments
 2. Webhook - listen to payment-intents 24/7
 3. Payment button click
 - a. Make a payment-intent
 - b. CreatePayment
 4. Success or failure -> listened by webhook
-

Add more mock interview questions in the database and give a wide variety of questions to the interviewer

make it specifically for developers

make it more detailed when choosing a question

add js, react, next js, github, etc etc as the topics and make it very very detailed
