

علیرضا ارومند

نام دوره: دوره آنلاین از Prompt تا Production مهندسی

نرم افزار در عصر هوش مصنوعی

طول دوره: ۱۸ ساعت

نوع ارائه: آنلاین

زمان برگزاری دوره: از ۲۹ تیر - دوشنبه ها ۱۷ - ۱۹

پیش نیاز شرکت در دوره: ۵ سال سابقه برنامه نویسی

سطح دوره: پیشرفته

مخاطبین: توسعه دهندگان نرم افزار با حداقل چند سال تجربه عملی

در برنامه نویسی مهندسان نرم افزار و برنامه نویسان Backend،

Full-Stack و Frontend



مدرس دوره

علیرضا ارومند

مدرس، معمار و مشاور ارشد نرم افزار Product Manager در شرکت داتین (وابسته به فناپ) در حوزه پروژه های بانکی
مدرس و Technical Manager پروژه های نیک آموز
متخصص در تولید فریمورک برنامه نویسی فوق العاده حرفه ای با قابلیت مدیریت بیش از ۱ میلیون تراکنش در ثانیه
همکاری با تیم توسعه شرکت ارتباط فردا (بانک آینده)
مشاور فنی شرکت توسعه رفاه پردیس (بانک رفاه)
سخنران تنها همایش مورد تأیید میکروسافت در خاورمیانه در حوزه ASP.NET Core
مدیر فنی خبرگزاری بین المللی پیام کوتاه نسیم (برنده جشنواره وب ایران)
مدرس دوره های Dot Net و ASP.NET در نیک آموز



جلسه ۱: مبانی GenAI و تغییر نقش توسعه دهنده نرم افزار

هدف جلسه

ایجاد درک مشترک از هوش مصنوعی مولد، مدل های زبانی، نقش آن ها در توسعه نرم افزار و تغییرات ایجاد شده در کار توسعه دهندگان.

- هوش مصنوعی مولد و مدل های زبانی بزرگ چیستند؟
- مفهوم Prompt، Context و Response
- تفاوت استفاده سنتی از ابزارها با استفاده از AI در توسعه نرم افزار
- مفهوم Agent و Agentic Application
- تغییر نقش توسعه دهنده از تولیدکننده صرف کد به بازیبن، اعتبارسنج و تصمیم گیرنده فنی
- فرصت های AI در افزایش سرعت توسعه
- محدودیت ها و ریسک ها: Hallucination، دانش قدیمی، پاسخ نادرست و خروجی غیرقابل اتکا
- خطر پذیرش بی بررسی خروجی AI و مفهوم Vibe Coding

تمرین پیشنهادی

- حل یک مسئله ساده کدنویسی با کمک AI و سپس بررسی خروجی از نظر صحت، کیفیت، خوانایی، امنیت و قابلیت نگهداری.

خروجی جلسه

- چک لیست اولیه برای استفاده مسئولانه از AI در کدنویسی.



علیرضا ارومند

جلسه ۲: طراحی Prompt و کنترل خروجی مدل

هدف جلسه

آموزش روش‌های کنترل خروجی مدل از نظر ساختار، قالب، سبک پاسخ، دقت و تناسب با نیاز فنی.

- چرا خروجی مدل‌های زبانی متغیر و غیرقطعی است؟
- اصول طراحی Prompt مؤثر
- تفاوت Prompt آزاد با Prompt ساختاریافته
- کنترل قالب خروجی با ساختارهای مشخص
- استفاده از مثال‌ها برای هدایت مدل
- طراحی Prompt برای تولید کد
- طراحی Prompt برای توضیح کد
- طراحی Prompt برای بازنویسی، Refactoring و مستندسازی
- الگوهای استدلالی برای حل مسائل پیچیده‌تر
- بهینه‌سازی Prompt بر اساس کیفیت خروجی

تمرین پیشنهادی

- برای یک مسئله مشخص، سه نوع Prompt طراحی شود: Prompt ساده، Prompt ساختاریافته و Prompt با خروجی قالب‌مند. سپس خروجی‌ها با یکدیگر مقایسه شوند.

خروجی جلسه

- قالب استاندارد Prompt برای تولید کد، توضیح کد و بازبینی خروجی.



علیرضا ارومند

جلسه ۳: افزودن دانش به مدل با RAG و تولید خروجی قابل اعتماد

هدف جلسه

آموزش روش‌هایی برای اتصال مدل به دانش بیرونی، مستندات، کدها و منابع سازمانی به منظور کاهش خطا و افزایش دقت پاسخ.

- مشکل محدودیت دانش مدل‌ها و Knowledge Cutoff.
- چرا مدل به تنهایی دانش کافی از سیستم، مستندات و کدبیس سازمان ندارد؟
- مفهوم Retrieval-Augmented Generation یا RAG.
- اجزای اصلی RAG: منبع داده، Chunking، Embedding، Vector Store و Retrieval.
- Semantic Indexing و اهمیت جستجوی معنایی.
- اهمیت Metadata در بازیابی دقیق‌تر.
- Trustworthy Generation و تولید پاسخ مبتنی بر منبع.
- Deep Search و بازیابی چندمرحله‌ای.
- کاربرد RAG در مستندسازی فنی، پاسخ‌گویی به پرسش‌های توسعه‌دهندگان و توضیح کد.
- اهمیت ارجاع به منبع، Grounding و جلوگیری از پاسخ بی‌پشتوانه.

تمرین پیشنهادی

- یک مجموعه کوچک از مستندات یا کد فرضی در اختیار تیم قرار می‌گیرد. شرکت‌کنندگان مشخص می‌کنند چه بخش‌هایی باید ایندکس شوند، چه Metadata‌هایی لازم است و چه Contextی باید در اختیار مدل قرار گیرد.

خروجی جلسه

- طرح ساده RAG برای یک سناریوی فنی، مستندسازی یا پاسخ‌گویی داخلی.



علیرضا ارومند

جلسه ۴: افزایش قابلیت اطمینان؛ ارزیابی، AI Review و Reflection

هدف جلسه

آموزش روش‌های کاهش خطا، اعتبارسنجی خروجی مدل و استفاده از AI در بازبینی کد و تحلیل کیفیت.

- مفهوم Reliability در سیستم‌های مبتنی بر GenAI
- چرا خروجی مدل باید ارزیابی شود؟
- استفاده از مدل به عنوان داور یا LLM-as-a-Judge
- الگوی Reflection و خوداصلاحی مدل
- Self-Check و بررسی پاسخ توسط خود سیستم
- Guardrails و محدودسازی خروجی‌های پرریسک
- استفاده از AI در Code Review
- تشخیص Bug، Code Smell، آسیب‌پذیری و Regression
- نقش بازبین انسانی در کنار AI Review
- محدودیت‌های AI در بازبینی کد و تصمیم‌گیری نهایی

تمرین پیشنهادی

- یک قطعه کد دارای مشکل به AI داده می‌شود تا آن را بازبینی کند. سپس تیم خروجی بازبینی AI را بررسی کرده و موارد صحیح، موارد اشتباه و موارد جاافتاده را مشخص می‌کند.

خروجی جلسه

- چک‌لیست. AI-assisted Code Review.



علیرضا ارومند

جلسه ۵: کاربردهای عملی GenAI در چرخه توسعه نرم افزار

هدف جلسه

آشنایی با کاربردهای عملی GenAI در مراحل مختلف توسعه نرم افزار و طراحی یک جریان کاری روزمره برای تیم توسعه.

- استفاده از AI در Autocompletion و Code Generation
- ابزارهای مبتنی بر مرورگر و ابزارهای یکپارچه با IDE
- استفاده از AI در طراحی UI/UX و Frontend Development
- استفاده از AI در Bug Detection و Code Review
- استفاده از AI در Automated Testing و QA
- تولید Unit Test و Integration Test با کمک AI
- استفاده از AI در تحلیل داده، گزارش گیری و بهینه سازی عملکرد
- تولید مستندات فنی، API Docs، Changelog و راهنماهای کاربری
- طراحی Chatbot و Virtual Assistant برای کاربردهای فنی و سازمانی
- Code Execution و Tool Calling
- همکاری چندعاملی و تقسیم وظایف بین چند Agent

تمرین پیشنهادی

- یک Feature کوچک انتخاب می شود و تیم از AI در چهار فعالیت استفاده می کند: تولید تست، تولید مستندات و بازبینی خروجی.

خروجی جلسه

- یک Workflow عملی برای استفاده از AI در توسعه روزمره نرم افزار.



علیرضا ارومند

جلسه ۶: بهینه‌سازی، ایمنی و طراحی Workflow های Agentic

هدف جلسه

جمع‌بندی مفاهیم دوره و آموزش نحوه ترکیب الگوها برای طراحی یک Workflow قابل استفاده در تیم‌های نرم‌افزاری.

- محدودیت‌های عملیاتی مدل‌ها در هزینه، سرعت، حافظه و پایداری
- استفاده از مدل‌های کوچک‌تر در سناریوهای مناسب
- Prompt Caching و کاهش هزینه پردازش
- Inference Optimization در سطح مفهومی
- Degradation Testing و بررسی رفتار سیستم در شرایط نامطلوب
- Long-Term Memory و حفظ زمینه در تعاملات طولانی‌تر
- Template Generation و تولید خروجی‌های کنترل‌شده
- Assembled Reformat و بازآرایی خروجی مدل به قالب قابل مصرف
- Guardrails در سطح Workflow
- طراحی Composable Agentic Workflows
- روش ارزیابی ابزارها و Workflow های جدید
- مرزبندی مسئولیت AI و انسان در توسعه نرم‌افزار

تمرین نهایی

- طراحی یک Workflow کامل برای یک سناریوی واقعی توسعه نرم‌افزار:

علیرضا ارومند

نیازمندی

- Prompt ساختاریافته →
- بازیابی دانش یا مستندات →
- تولید کد →
- تولید تست →
- Reflection →
- AI Review →
- Human Review →
- Documentation →



خروجی نهایی دوره

یک نمونه Workflow مستند برای استفاده از GenAI در توسعه نرم‌افزار. در پایان دوره، شرکت‌کنندگان یک Workflow مستند و قابل‌استفاده برای به‌کارگیری GenAI در فرآیند توسعه نرم‌افزار طراحی می‌کنند. منظور از Workflow، یک جریان کاری مرحله‌به‌مرحله است که مشخص می‌کند در هر بخش از فرآیند توسعه نرم‌افزار، از هوش مصنوعی چگونه، با چه هدفی، با چه ورودی‌ای و با چه سطحی از کنترل و بازبینی استفاده می‌شود.

این Workflow نشان می‌دهد که چگونه می‌توان از GenAI برای حرکت از یک ایده یا نیازمندی اولیه تا رسیدن به خروجی قابل‌استفاده نرم‌افزاری بهره‌گرفت؛ به‌گونه‌ای که استفاده از AI صرفاً محدود به تولید کد نباشد، بلکه در کل چرخه توسعه نقش داشته باشد.

در این خروجی، شرکت کنندگان مشخص می کنند:

- چگونه یک ایده یا نیازمندی محصول با کمک AI تحلیل و شفاف سازی می شود.
- با چه ابزار یا مدل هوش مصنوعی می توان نیازمندی را به User Story، سناریوهای کاربری یا تسک های فنی تبدیل کرد.
- برای هر مرحله، چه نوع Prompt یا دستورالعملی باید طراحی شود.
- ساختار Prompt ها چگونه باشد تا خروجی مدل قابل کنترل، دقیق و قابل ارزیابی باشد.
- در چه مراحل باید از مستندات، کدبیس یا دانش سازمانی به عنوان Context استفاده شود.
- چگونه خروجی AI به طراحی فنی، تولید کد، تولید تست و مستندسازی تبدیل می شود.
- خروجی تولید شده توسط AI چگونه با Reflection، AI Review و Human Review بررسی و اصلاح می شود.
- در نهایت چگونه این جریان کاری به تولید یک خروجی نرم افزاری قابل اعتماد و قابل استفاده منتهی می شود.

به بیان ساده، خروجی نهایی دوره یک نقشه عملیاتی است که نشان می دهد:

از تعریف محصول و نیازمندی، تا طراحی User Story، تولید Prompt، تولید کد، تست، بازبینی و مستندسازی، چگونه می توان از GenAI به شکل حرفه ای، کنترل شده و قابل اعتماد در توسعه نرم افزار استفاده کرد.



سوالات متداول دوره آنلاین از Prompt تا Production مهندسی نرم افزار در عصر هوش مصنوعی

۱. مدل زبانی بزرگ یا LLM در توسعه نرم افزار چه نقشی دارد؟

مدل زبانی بزرگ یا LLM می تواند متن و کد را تحلیل، تولید، بازنویسی و توضیح دهد. در توسعه نرم افزار، LLM در فعالیت هایی مانند تولید کد، توضیح کدهای پیچیده، تولید Unit Test، پیشنهاد Refactoring، مستندسازی API، تحلیل خطا، بررسی Pull Request و حتی طراحی اولیه معماری کاربرد دارد.

البته LLM جایگزین کامل توسعه دهنده نیست؛ زیرا ممکن است دچار خطا، Hallucination، برداشت نادرست از نیازمندی یا تولید کد ناامن شود. به همین دلیل در این دوره تأکید می شود که AI باید در قالب یک فرآیند کنترل شده، قابل بررسی و قابل ارزیابی استفاده شود.

۲. Prompt، Context و Response چه تفاوتی با هم دارند؟

Prompt همان دستور، سوال یا مسئله ای است که به مدل داده می شود. **Context** اطلاعات زمینه ای، محدودیت ها، مستندات، کدها یا توضیحاتی است که به مدل کمک می کند پاسخ دقیق تری تولید کند.

Response خروجی ای است که مدل در پاسخ به Prompt و Context تولید می کند. برای مثال، اگر از مدل بخواهید یک تابع احراز هویت بنویسد، خود درخواست شما Prompt است. اگر معماری پروژه، نسخه فریم ورک، استانداردهای امنیتی و نمونه کدهای موجود را هم به مدل بدهید، این اطلاعات Context محسوب می شوند. پاسخی که مدل تولید می کند نیز Response است. در کار حرفه ای با GenAI، کیفیت Context معمولاً به اندازه کیفیت Prompt اهمیت دارد.

۳. چرا خروجی مدل های زبانی همیشه قطعی و یکسان نیست؟

مدل های زبانی معمولاً بر اساس احتمال، توکن بعدی را پیش بینی می کنند. به همین دلیل ممکن است برای یک Prompt یکسان، پاسخ های متفاوتی تولید کنند؛ مخصوصاً اگر تنظیماتی مانند Temperature، Top-p یا نمونه برداری تصادفی فعال باشد.

این ویژگی می تواند برای ایده پردازی و خلاقیت مفید باشد، اما در توسعه نرم افزار یک ریسک محسوب می شود؛ چون برای تولید کد، تست یا تصمیم های فنی، به خروجی قابل اعتماد، قابل کنترل و قابل تکرار نیاز داریم. در این دوره یاد می گیرید چگونه با Prompt ساختاریافته، قالب خروجی مشخص، مثال ها، Guardrails و فرآیندهای بازبینی، این عدم قطعیت را مدیریت کنید.

۴. Prompt ساختاریافته چه تفاوتی با Prompt آزاد دارد؟

در Prompt آزاد، درخواست معمولاً به شکل ساده و غیررسمی مطرح می‌شود؛ مثلاً:

این کد را بهتر کن.

اما در Prompt ساختاریافته، هدف، نقش مدل، ورودی‌ها، محدودیت‌ها، قالب خروجی، معیارهای کیفیت و موارد ممنوع به صورت شفاف مشخص می‌شوند.

برای مثال، می‌توان از مدل خواست ابتدا مشکل کد را تحلیل کند، سپس پیشنهادهای Refactoring بدهد، بعد نسخه اصلاح‌شده کد را تولید کند و در انتها ریسک‌های احتمالی را فهرست کند. Prompt ساختاریافته باعث می‌شود خروجی مدل قابل کنترل‌تر، دقیق‌تر، قابل مقایسه‌تر و مناسب‌تر برای استفاده حرفه‌ای در توسعه نرم‌افزار باشد.

۵. آیا در دوره یاد می‌گیریم چطور برای تولید کد Prompt بنویسیم؟

بله. یکی از محورهای اصلی دوره، طراحی Prompt برای تولید کد است. اما هدف فقط این نیست که مدل برای ما کد بنویسد؛ بلکه یاد می‌گیریم چگونه درخواست تولید کد را طوری طراحی کنیم که خروجی آن قابل استفاده، قابل بررسی و همسو با نیاز فنی پروژه باشد.

در دوره بررسی می‌شود که یک Prompt مناسب برای تولید کد باید شامل مواردی مثل زبان برنامه‌نویسی، فریم‌ورک، محدودیت‌های معماری، سبک کدنویسی، الزامات امنیتی، تست‌پذیری، الگوی طراحی و قالب خروجی باشد.

همچنین تأکید می‌شود که خروجی تولیدشده توسط AI نباید بدون بررسی پذیرفته شود و باید از نظر کیفیت، صحت، امنیت و قابلیت نگهداری ارزیابی گردد.

۶. آیا می‌توان از GenAI برای Refactoring و بهبود کیفیت کد استفاده کرد؟

بله. یکی از کاربردهای مهم GenAI در توسعه نرم‌افزار، کمک به Refactoring، ساده‌سازی کد، بهبود خوانایی، حذف تکرار، پیشنهاد الگوهای بهتر و مستندسازی بخش‌های پیچیده است.

با این حال، Refactoring با کمک AI باید با احتیاط انجام شود؛ چون ممکن است مدل به صورت ناخواسته رفتار کد را تغییر دهد یا بخشی از منطق اصلی برنامه را نادیده بگیرد.

در این دوره یاد می‌گیرید چگونه Prompt‌هایی طراحی کنید که مدل تفاوت بین بازنویسی ظاهری، Refactoring واقعی و تغییر رفتار برنامه را رعایت کند. همچنین نقش تست، بازبینی انسانی و بررسی Regression در این فرآیند توضیح داده می‌شود.

۷. Knowledge Cutoff چه مشکلی در استفاده از AI برای برنامه نویسی ایجاد می کند؟

مدل های زبانی معمولاً تا یک تاریخ مشخص آموزش دیده اند و ممکن است از نسخه های جدید کتابخانه ها، فریم ورک ها، API ها یا تغییرات معماری پروژه شما اطلاع نداشته باشند. به این محدودیت، Knowledge Cutoff گفته می شود.

در توسعه نرم افزار، این موضوع می تواند باعث تولید کد قدیمی، استفاده از API منسوخ شده، پیشنهاد روش های ناامن یا ارائه راه حل ناسازگار با نسخه فعلی پروژه شود. برای مدیریت این مسئله، در دوره به استفاده از Context مناسب، مستندات به روز، RAG و ارجاع به منبع پرداخته می شود تا پاسخ های مدل دقیق تر و قابل اعتمادتر باشند.

۸. Hallucination در کدنویسی با AI چه شکلی پیدا می کند؟

Hallucination زمانی رخ می دهد که مدل پاسخی تولید می کند که ظاهراً منطقی و مطمئن به نظر می رسد، اما در واقع نادرست، بی پشتوانه یا غیرقابل اجراست.

در کدنویسی، Hallucination می تواند به شکل های مختلفی ظاهر شود؛ از جمله: استفاده از تابعی که در کتابخانه وجود ندارد

پیشنهاد API منسوخ شده

تولید کدی که کامپایل یا اجرا نمی شود

نادیده گرفتن نیازمندی های امنیتی

ارائه توضیح اشتباه درباره رفتار کد

تولید تست هایی که سناریوهای مهم را واقعاً پوشش نمی دهند

پیشنهاد راه حل هایی که با معماری پروژه سازگار نیستند

یکی از اهداف این دوره، آموزش روش هایی برای کاهش این ریسک از طریق Prompt دقیق، Context مناسب، بازبینی، تست، Reflection و Human Review است.

۹. AI-assisted Code Review چه تفاوتی با Code Review انسانی دارد؟

AI-assisted Code Review یعنی استفاده از AI برای کمک به بازبینی کد. AI می تواند مواردی مثل Bug احتمالی، Code Smell، پیچیدگی بیش از حد، تکرار کد، نبود تست، مشکلات خوانایی، ناسازگاری با استانداردهای کدنویسی و برخی آسیب پذیری ها را تشخیص دهد.

اما Code Review انسانی همچنان ضروری است؛ چون تصمیم های معماری، درک عمیق از دامنه کسب و کار، اولویت های محصول، ملاحظات تیمی، Trade-off های فنی و ریسک های پنهان معمولاً نیاز به قضاوت انسانی دارند. در این دوره یاد می گیرید AI را به عنوان دستیار بازبینی کد استفاده کنید، نه جایگزین کامل Reviewer انسانی.

۱۰. Code Execution و Tool Calling در توسعه نرم افزار چه کاربردی دارند؟

Tool Calling به مدل اجازه می دهد برای انجام یک کار، از ابزارهای بیرونی استفاده کند؛ مثلاً جست و جو در مستندات، اجرای کوئری، خواندن فایل، فراخوانی API یا اجرای تست.

Code Execution یعنی مدل بتواند کد تولید شده یا بخشی از آن را در یک محیط کنترل شده اجرا کند و نتیجه را بررسی کند. این قابلیت برای Debugging، تولید تست، تحلیل داده و اعتبارسنجی خروجی مفید است.

اما این قابلیت ها باید با محدودیت های امنیتی و سطح دسترسی کنترل شده استفاده شوند؛ چون اجرای کد بدون نظارت می تواند خطرناک باشد.

۱۱. آیا خرید اقساطی امکان پذیر است؟

بله، امکان خرید اقساطی با اسنپ پی فراهم شده است. برای اطلاعات بیشتر می توانید با مشاورین مجموعه در تماس باشید یا راهنمای [خرید اقساطی دوره آموزشی با اسنپ پی](#) را مطالعه بفرمایید.

۱۲. آیا فیلم دوره رکورد می گردد؟

بله، دسکتاپ و صدای مدرس رکورد خواهد شد و در پلیر اختصاصی اسپات پلیر به همراه کلید لایسنس ارائه خواهد شد. شما در سیستم عامل های ویندوز، اندروید، آیفون (سیب، اناردون)، مک بوک می توانید فیلم را مشاهده کنید.

۱۳. افرادی که بصورت لایو کلاس را مشاهده می کنند، آیا امکان پرسش و پاسخ دارند؟

شما بصورت چت آنلاین می توانید سوالات خود را بپرسید و مدرس هم سوالات شما را پاسخ خواهد داد. البته توجه داشته باشید که این پروسه پاسخگویی هر ۴۰ دقیقه یکبار خواهد بود تا مدرس رشته کلام از دستش خارج نشود. البته که گروه تلگرامی دوره در اختیار شما است و می توانید سوالات خود را آنجا هم مطرح کنید.

۱۴. آیا امکان برگزاری دوره های سازمانی وجود دارد؟

بله؛ در نیک آموز امکان برگزاری دوره های سازمانی به صورت تخصصی فراهم شده است. به منظور ثبت درخواست، کافی است اطلاعات خود و دوره سازمانی مدنظر را در [فرم درخواست آموزش سازمانی](#) ثبت کنید تا ما با شما تماس بگیریم.

۱۵. می‌خواهم مشاوره / تدریس خصوصی بگیرم؟

برای اینکه بتوانید در ارتباط با برنامه‌نویسی، مشاوره / تدریس خصوصی بگیرید، لطفاً [فرم درخواست مشاوره مدرسین](#) را تکمیل نمایید تا کارشناسان ما با شما تماس بگیرند.

۱۶. آیا می‌توانم مشاوره سازمانی برای پروژه دریافت کنم؟

شما می‌توانید با مراجعه به [فرم درخواست مشاوره تخصصی](#)، از متخصصان نیک آموز مشاوره دریافت کنید و با به‌کارگیری مهارت‌های تجربی تیم ما، در ارتباط با پروژه‌های تخصصی خود، راهنمایی دریافت کنید.

۱۷. من باز هم سؤال دارم؛ امکان ارتباط با مشاوران نیک آموز وجود دارد؟

بله؛ شما می‌توانید از مشاوره‌های [نیک آموز](#) به‌عنوان راهنما در مسیر خود استفاده کنید. برای این منظور، لطفاً شماره خود را در فرم مشاوره صفحه دوره وارد کنید تا مشاوران نیک آموز با شما تماس بگیرند.

شماره تماس فروش: [02191070017](tel:02191070017) – داخلی ۱

شماره تماس پشتیبانی: [02191070017](tel:02191070017) – داخلی ۲

۱۸. نحوه پشتیبانی دوره به چه صورت است؟

رضایت شما از دوره آموزشی و کمک به رفع مشکلات احتمالی برای ما اهمیت زیادی دارد. به همین دلیل، یک گروه پشتیبانی در تلگرام ایجاد شده است تا شما بتوانید در صورت نیاز، مسائل خود را در این بستر مطرح کنید. تا حداکثر ۴۸ ساعت کاری پس از ثبت نام در دوره، با شما تماس گرفته می‌شود و فرآیند عضویت شما در گروه تلگرام نهایی خواهد شد. توجه شود که در آینده، سیستم تیکتینگ راه‌اندازی می‌شود و فرآیند پشتیبانی از گروه تلگرامی به آن جا منتقل خواهد شد.

[لینک ثبت نام: از Prompt تا Production — مهندسی نرم افزار در عصر هوش مصنوعی](#)