Section 1
AI Agent Title – AI Crack hub
Use Cases - An AI Agent to Help Users Crack Interviews

Section 2

2.1 What problem does your AI Agent solve?

Many users struggle with structured interview preparation. They often spend hours searching across multiple platforms for reliable and relevant questions. Our AI Agent solves this common and valuable problem by providing complete, end-to-end interview preparation in one place — tailored to specific domains like Machine Learning, AI, or Full Stack Development. It focuses on delivering the most probable and high-impact questions that are likely to be asked in real interviews.

2.2 Why is this Agent useful?

This agent is valuable because it brings all essential interview preparation materials into one place. It provides well-structured, concise answers with examples — helping users deeply understand key concepts quickly. Within just a few hours, a user can be interview-ready without wasting time on unnecessary or irrelevant content. It avoids fluff and focuses only on what truly matters. As a bonus, it allows users to download the Q&A set for free. By using this agent, candidates get up to 30% better chances of cracking their interviews and landing their dream job.

2.3 Who is the target user?

Anyone preparing for a job interview — including college students, freshers, working professionals, or career switchers looking to enter tech domains.

2.4 What not to include?

- Do not provide overly long or overly short answers — keep it focused and precise.
- Avoid mixing unrelated domains (e.g., don’t mix Data Analytics with Data Annotation).
- Do not include irrelevant or unlikely-to-be-asked questions — only focus on high-priority topics.
- Avoid unnecessary UI elements or distracting behavior — the agent must stay focused on learning and content delivery only.

Section3

prompt: “The user is preparing for an interview as a {job_role}. Generate a comprehensive list of topics and subtopics that are crucial to master for this role. Cover theoretical concepts, tools, frameworks, real-world practices, and niche subjects where applicable. Present the topics in structured format with categories.”

what is this prompt responsible for?

It is responsible for identifying the user’s intent and domain, and generating a list of relevant topics the candidate should prepare for.

Example Input: “Help me prepare for a Machine Learning Engineer interview.”

Example Output:

1. Machine Learning Fundamentals

   - Supervised vs Unsupervised

   - Bias-Variance Tradeoff

   - Overfitting & Underfitting

2. Algorithms & Models

   - Linear Regression, Logistic Regression

   - Decision Trees, Random Forests, XGBoost

   - SVM, KNN, Naive Bayes

3. Model Evaluation

   - Accuracy, Precision, Recall

   - ROC-AUC, Confusion Matrix

   - Cross-validation

...

9. ML System Design & Deployment

   - ML Pipelines

   - CI/CD in ML

   - Model Drift & Retraining

3.2 STATE TRACKER

Prompt:

The agent stores the current session's information, including the user's domain, the list of generated topics, questions, answers, and any user comments or clarifications. This allows the agent to track the conversation context and refer back to previous questions or details when requested.

How does this help the agent "remember"?

This state tracker functions as a short-term memory for the AI agent, enabling it to maintain context throughout the user's current interview preparation session. For example, when the user asks, "Explain question 5 again," the agent can fetch that exact question and answer from the stored session data and provide detailed clarification or expansion.

Did you simulate memory with variables / system messages? If yes, how?

Yes. The memory is simulated using structured data storage (like JSON objects or database rows) that hold all session-relevant information. Each interaction appends or updates this data store. System messages or internal variables keep track of session context, so the agent can reference and build upon previous exchanges dynamically.

3.3 TASK PLANNER

Prompt:

The agent internally breaks down the interview preparation task into four sequential steps, each handled by a separate model:

1. **Model 1** fetches all relevant topics and subtopics for the user's specified job role.
2. **Model 2** filters these topics to identify the most important core topics likely to appear in interviews.
3. **Model 3** generates high-quality interview questions and detailed answers for each core topic.
4. **Model 4** reviews the question-answer set, removes redundant or less relevant questions, and prioritizes the most impactful ones.

Did you use chaining? Branching? How did you manage complexity?

We use a **chaining approach** where the output of each model serves as the input to the next, ensuring a smooth, linear workflow. This modular design helps manage complexity by isolating different subtasks and allowing focused processing at each stage. Additionally, we leverage **runnables** to organize and execute these chained steps efficiently, making the system scalable and easier to debug or enhance.

3.4 OUTPUT GENERATOR

Prompt:

The output is designed to be clear, concise, and highly structured to facilitate easy understanding and quick revision. The formatting emphasizes readability with logical sections, summaries, and numbered Q&A for smooth navigation.

Output formatting and phrasing goals:

- **Domain Overview:** Start with a brief summary of the domain, giving users context and highlighting the importance of the core skills.
- **Core Topics List:** Present the key topics the candidate should focus on, ensuring comprehensive coverage without overwhelming detail.
- **Interview Questions & Answers:** For each core topic, provide 1-2 well-crafted questions with detailed, clear answers.
- **Formatting:** Use headings and subheadings, numbered lists for questions (Q1, Q2) and answers (A1, A2), and keep language precise and professional.
- **Highlighting:** Important terms and techniques can be bolded to draw attention.
- **Export-ready:** The structure supports easy conversion to markdown, PDF, or plain text for user convenience.

Example Output:

Domain Overview:
Machine Learning is a rapidly evolving field focused on building models that learn from data to make predictions or decisions. Mastery of algorithms, model evaluation, and system design is crucial for interview success.

Core Topics to Focus On:

- Overfitting and Regularization
- Model Evaluation Metrics
- Feature Engineering

Interview Questions & Answers:

1. Overfitting and Regularization
Q1: What is overfitting in machine learning, and how can it be prevented?
A1: Overfitting occurs when a model learns noise in the training data instead of general patterns,

resulting in poor generalization to new data. Prevention techniques include **regularization (L1/L2)**, cross-validation, early stopping, and simplifying the model.

Q2: How do L1 and L2 regularization differ?
A2: L1 regularization (**Lasso**) penalizes the absolute value of coefficients and can zero out some features, leading to sparse models. L2 regularization (**Ridge**) penalizes the squared value of coefficients, shrinking them but rarely to zero.

Section 4

Attempt #	Prompt Variant	What Happened	What You Changed	Why You Changed It
1	“Generate journaling prompts”	Output was too generic	Added user context to prompt	To make prompts more personal and relevant
2	Data Scientist Q&A with simple answers	Answers were brief and sometimes redundant	Added more detailed explanations and examples	To improve answer quality and clarity
3	Feature Engineering Q&A	Answers missing depth in explanations	Expanded answers with examples and pros/cons	To provide clearer understanding and practical insights
4	Asked AI Crack Hub to enhance/prune Q&A	Needed to remove redundant questions	Asked AI Crack Hub to enrich content and prune simpler Qs	To improve focus and quality of interview content
5	Expanded categorical variables encoding answer	Detailed examples and pros/cons added	Provided thorough explanation and examples	To make the answer comprehensive and suitable for interviews
6	Machine Learning Algorithms roadmap & Q&A	Good structure but could add more questions	Added focused topics and suggested detailed answers	To cover key interview topics comprehensively

🧪 SECTION 5: OUTPUT TESTS

✅ Test 1: Normal Input

Input: "Give me interview questions on feature engineering"
Output:
Absolutely! Here are 10 high-quality interview questions on Feature Engineering, curated to reflect what top companies often ask in Data Science interviews:

1. What is feature engineering and why is it important in machine learning?

2. How do you handle missing values in a dataset during feature engineering?
3. What techniques can you use to encode categorical variables? Which one would you choose and why?
4. What is feature scaling, and when should you use normalization vs. standardization?
5. How do you deal with high-cardinality categorical variables?
6. Explain the concept of interaction features. Can you give an example where interaction terms improved a model?
7. What are binning and discretization? How can they help in improving model performance?
8. How do you select the most relevant features from your dataset? What methods do you use for feature selection?
9. What is target encoding? What are the risks involved in using it?
10. Can you explain polynomial features and when they might be useful in a model?

✅ If you'd like detailed answers with examples for any (or all) of these questions, just say the word — I'll deliver them in an easy-to-understand, interview-ready format!

🤔 **Test 2: Vague Input**

Input: "Help me prepare for interviews"
Output:
Absolutely! Welcome to AI Crack Hub – your personal interview preparation partner. Let's get started!

🧠 Step 1: Understanding Your Role
To tailor your interview preparation perfectly, I just need to know:

👉 What role are you preparing for?
(For example: Data Scientist, Machine Learning Engineer, NLP Engineer, Data Analyst, MLOps Engineer, etc.)

Once you tell me your target job role, I'll begin a 4-step process:

1. Identify all relevant topics
2. Extract the most important core concepts
3. Generate smart Q&A for each
4. Filter, format, and deliver an optimized prep guide

Please reply with your target role and (optionally) the company or level (e.g., fresher, 2 years exp, etc.), and we'll begin your tailored journey.

🚫 Test 3: Invalid Input

Input: ""
Output:
Thanks! Since the input is empty (""), I'll prompt you again just to confirm:

👋 Let's Start Your Interview Prep!
Please tell me:

- What role are you preparing for?
  (Example: Data Scientist, ML Engineer, NLP Specialist, MLOps, etc.)
- Your experience level or target (optional):
  (Example: Fresher, 1-2 years experience, applying to Google, etc.)

Once I have this info, I'll initiate your custom 4-stage preparation using our **AI Crack Hub** pipeline.

Just drop in your role and we'll begin! 💪

🔄 SECTION 6: REFLECTION

6.1. What was the hardest part of this assignment?
The hardest part of this assignment was generating outputs strictly within a specific domain without mixing in questions from similar or overlapping areas. Identifying and extracting only the most relevant questions for a particular topic was a bit challenging, especially since each domain contains a wide range of subtopics. Additionally, not having access to the premium version of ChatGPT caused occasional model switching between GPT-4.0 and 3.5, which led to inconsistent outputs—something that was quite frustrating at times.

6.2. What part did you enjoy the most?
I genuinely enjoyed every part of the assignment, but the testing phase stood out the most. Seeing everything come together and work exactly as planned was incredibly satisfying. It gave me a sense of accomplishment and motivation to continue improving.

6.3. If given more time, what would you improve or add?
If I had more time, I would fine-tune the model on my custom dataset to improve the accuracy and relevance of responses. I'd also focus on building a clean, classic UI and enhance the user experience by integrating educational videos and reference materials so users could dive deeper into each topic.

6.4. What did you learn about ChatGPT or prompt design?

I learned that asking the right question is key—one well-structured prompt can completely solve your problem. Prompt design plays a huge role in getting accurate, relevant responses. I also realized how much more powerful GPT-4.0 is compared to 3.5; it's clearly trained on a much larger dataset and offers more reliable results.

6.5. Did you ever feel stuck? How did you handle it?

Yes, I felt stuck several times, but I followed a personal 3-step strategy to overcome it:

1. **Reframe the Problem as a New Prompt**
   Instead of just asking for a variation, I explained the issue clearly and asked for solutions in bullet points. This usually helped unlock new perspectives.

2. **Cross-Check the Suggestions**
   I compared GPT's suggestions with what I had already tried. I discarded any repeated ideas and focused only on fresh, useful ones.

3. **Focus on One Direction**
   I selected one promising direction and communicated it clearly to GPT. I also highlighted any past mistakes—like using outdated tools or dummy data—and asked for a complete, corrected solution accordingly.

This approach helped me stay in control of the output and truly use GPT as a collaborative assistant rather than just a content generator.

🧠 SECTION 7: HACK VALUE

Did you go beyond the brief in any way?
Yes, we went beyond the brief in several meaningful ways:

🔹 I implemented a two-layer memory system in the AI Crack Hub agent — session memory (to track ongoing inputs like domain, questions, user comments) and long-term user profile memory (to personalize future responses based on past behavior).

🔹 I also designed the logic to simulate real user behavior, such as vague inputs, edge cases, and follow-ups.

🔹 Additionally, I explored agent modularity, preparing the system to be extended with other roles or tools like video recommendations or knowledge source linking.