

Test Technique - Ingénieur IA Générative Juillet 2025

1 Instructions générales

- **Durée estimée** : 2-3 heures
- **Livrable** : Repository GitHub (public ou privé) contenant votre code. Si privé, accordez-nous l'accès ou partagez le lien avec les permissions appropriées.
- **Format** : Code source + README détaillé

2 Partie 1 : Développement d'un Chatbot avec accès Internet

2.1 Contexte

Dans le cadre d'une mission hypothétique pour un grand groupe, vous devez construire un chatbot capable d'accéder à Internet pour rechercher des informations et les intégrer dans ses réponses.

2.2 Spécifications techniques

1. Modèle LLM :

- Créez un compte gratuit sur OpenRouter
- Utilisez le modèle `qwen/qwen-2.5-32b-instruct:free` (gratuit)
- Points bonus si vous utilisez un LLM en local

2. Stack technique :

- **Backend** : Python
- **Frontend** : React
- Interface de chat simple

3. Fonctionnalités requises :

- Capacité de recherche web (vous pouvez utiliser des APIs gratuites comme DuckDuckGo, SerpAPI free tier, ou web scraping)
- Intégration des résultats de recherche dans les réponses du LLM
- Gestion des erreurs et des limites de rate

4. Scénario de test obligatoire :

Votre chatbot doit pouvoir répondre à ce type de requête :

“Quels sont les derniers développements en IA générative annoncés cette semaine ? Donne-moi 3 exemples concrets avec leurs sources.”

3 Partie 2 : Questions théoriques sur le chatbot avec accès internet

Important : Cette partie théorique porte sur le système développé dans la Partie 1. Incluez vos réponses dans votre README ou dans un document séparé dans votre repository.

3.1 Architecture et Déploiement en Production

1. Plan de déploiement Azure (TRÈS IMPORTANT) :

- Décrivez étape par étape comment déployer votre solution sur Azure
- Identifiez TOUS les prérequis nécessaires
- Estimez les coûts mensuels
- Proposez une architecture scalable
- **Points d'attention** : Quels accès sont nécessaires ? Quelles permissions ? Accès au backend ? Autres ?
- Listez tous les prérequis techniques et organisationnels
- Quels services Azure utiliser et pourquoi ?
- Quelles sont les considérations de sécurité ?
- Comment gérer les secrets et API keys ?

2. Stratégie de mise en production :

- CI/CD pipeline
- Monitoring et logs
- Gestion des erreurs
- Stratégie de backup

4 Partie 3 : Challenge optionnel - Vision Language Models

ATTENTION : Uniquement si vous avez une configuration avec GPU (min 8GB VRAM)

4.1 Exercice pratique

Déployez un petit VLM en local (ex : SmolVLM, Qwen 2.5VL, ou autre modèle <3B paramètres) et créez un script Python simple qui :

- Prend une image en entrée
- Utilise le VLM pour extraire le texte ou décrire le contenu
- Affiche le résultat

4.2 Question théorique ouverte

Comment architectureriez-vous un système de vectorisation de documents qui intègre des VLMs ? Décrivez votre approche pour traiter des documents contenant à la fois du texte et des éléments visuels (images, graphiques, schémas) dans un pipeline de RAG.

5 Notes importantes

1. **Budget** : Toutes les solutions doivent utiliser des APIs gratuites ou des free tiers. Ne dépensez pas d'argent personnel.
2. **Sécurité** : Ne committez JAMAIS vos API keys. Utilisez des variables d'environnement.
3. **Questions** : Si vous avez des questions, documentez vos assumptions dans le README.

6 Ressources utiles

- [OpenRouter Documentation](#)
- [Azure OpenAI Service](#)
- [LangChain Documentation](#)
- [Hugging Face Hub](#)

7 Grille de notation

Critère	Pondération	Points d'évaluation	Score max
Code	40%	• Qualité et lisibilité du code	10
		• Gestion robuste des erreurs	10
		• Architecture modulaire	10
		• Documentation du code	10
		Sous-total	40 points
Compréhension technique	30%	• Pertinence des réponses théoriques	8
		• Profondeur de l'analyse	8
		• Connaissance écosystème Azure	7
		• Vision produit	7
		Sous-total	30 points
Plan de déploiement	20%	• Exhaustivité des prérequis	5
		• Réalisme du budget	5
		• Considérations de production	5
		• Scalabilité de la solution	5
		Sous-total	20 points
Innovation et initiative	10%	• Solutions créatives	3
		• Optimisations proposées	3
		• Anticipation des problèmes	2
		• Clarté de la documentation	2
		Sous-total	10 points
Bonus (optionnel)	+15%	• Script VLM fonctionnel	+5
		• Réflexion architecture multimodale	+5
		• Utilisation d'un LLM local	+5
		Sous-total	+15 points
TOTAL			100 points
TOTAL avec bonus			115 points

7.1 Barème de notation

- **90-100 points** : Excellent - Maîtrise complète des concepts
- **75-89 points** : Très bien - Solides compétences avec quelques améliorations possibles
- **60-74 points** : Bien - Compétences de base acquises, potentiel de progression
- **45-59 points** : Passable - Lacunes importantes à combler
- **< 45 points** : Insuffisant - Ne répond pas aux attentes du poste

Bonne chance ! Nous avons hâte de voir votre créativité et vos compétences techniques en action.