**STANDARD INTERVIEW QUESTIONS**

**SECTION 1: STAR BEHAVIORAL QUESTIONS**

1.1 Tell me about a time you faced a really hard technical problem that wasn't straightforward.

1.2 Tell me about a time you failed. What happened, and what did you learn from it?

1.3 Tell me about a time you disagreed with a team member or manager. How did you handle it?

1.4 Tell me about a time you took initiative and did something without being asked.

1.5 Tell me about a time you got stuck and had to ask for help. How did you handle it?

1.6 Tell me about a time something unexpected happened during a project. How did you adapt?

1.7 Tell me about a time you worked in a team that was really great. What made it great?

1.8 Tell me about a time working in a team was difficult. What happened and how did you resolve it?

1.9 Tell me about a time you went above and beyond to deliver something excellent.

1.10 Tell me about a time a client or manager was wrong about something. How did you handle it?

1.11 Tell me about a time you set a goal but couldn't meet it. How did you handle it?

1.12 What's your greatest achievement? Why is it meaningful to you?

**SECTION 2: LIBRARIES & FRAMEWORKS**

**Flask**

2.1 Why Flask over Django?

2.2 How do you structure a Flask app at scale?

2.3 How do you handle errors and exceptions in Flask?

2.4 What's the difference between development and production deployment?

2.5 Explain request context vs app context.

2.6 How would you deploy Flask to production?

**python-telegram-bot**

2.7 Explain polling vs webhooks for Telegram bots.

2.8 How do you handle long-running operations in a bot?

2.9 How do you secure your bot token?

2.10 How do you store user state across interactions?

2.11 What are handlers, filters, and context in telegram-bot?

**Twilio**

2.12 What are rate limits for Twilio API?

2.13 How would you handle Twilio API failures gracefully?

2.14 What's the cost model for Twilio?

2.15 Explain sync vs async Twilio API calls.

2.16 What can you do with Twilio besides phone lookups?

**OpenAI API**

2.17 How do you manage costs with OpenAI API?

2.18 What's the difference between temperature and max\_tokens?

2.19 How do you prevent hallucinations from GPT?

2.20 Explain few-shot learning with GPT.

2.21 What's token counting and why does it matter?

**Oracle Database**

2.22 How does indexing work in Oracle?

2.23 What's the difference between primary key and foreign key?

2.24 Explain transactions and ACID properties.

2.25 How do you optimize slow SQL queries?

2.26 What's partitioning vs sharding?

2.27 How do you handle concurrent writes?

**Pandas & NumPy**

2.28 What's the difference between loc and iloc?

2.29 How do you handle missing data in a DataFrame?

2.30 Explain groupby and aggregation.

2.31 What's the difference between merge and join?

2.32 When would you use apply()?

2.33 Explain broadcasting in NumPy.

**Scikit-learn**

2.34 Explain train/test split. Why 80-20?

2.35 What's cross-validation and why use it?

2.36 How do you tune hyperparameters?

2.37 Random Forest vs Decision Tree—when use each?

2.38 What's overfitting and how do you detect it?

2.39 Explain precision, recall, F1 score, and accuracy.

**Keras/TensorFlow**

2.40 What's the difference between Keras and TensorFlow?

2.41 What's dropout and why use it?

2.42 Explain backpropagation briefly.

2.43 What's activation function? Why ReLU?

2.44 How do you prevent overfitting in neural networks?

**PyTorch**

2.45 What's autograd in PyTorch?

2.46 Difference between PyTorch and TensorFlow?

2.47 What's a computational graph?

2.48 How do you train a model in PyTorch?

**NLTK**

2.49 Difference between tokenization, stemming, and lemmatization?

2.50 What are stopwords? When do you remove them?

2.51 Explain POS tagging.

2.52 How would you build an NLP pipeline?

**Hugging Face Transformers**

2.53 What's BERT and how is it different from traditional ML?

2.54 Explain RoBERTa vs BERT. Which is better and why?

2.55 What's transfer learning in NLP?

2.56 How do you fine-tune a transformer model?

2.57 Explain what a tokenizer does.

2.58 What's the attention mechanism?

**Gensim**

2.59 What's Word2Vec?

2.60 What's LDA?

2.61 When would you use Gensim over other NLP libraries?

**SECTION 3: PROJECTS**

**Phone Checker Bot**

3.1 Walk me through the Phone Checker Bot from start to finish.

3.2 What's the problem it solves?

3.3 What's the architecture?

3.4 Why integrate Twilio, DuckDuckGo, and OpenAI specifically?

3.5 You mention 40% improvement with caching. What was the baseline?

3.6 How did you measure the 40% improvement?

3.7 Why Oracle Database for caching instead of Redis?

3.8 How would you scale this to 1 million concurrent users?

3.9 What breaks at scale? 3.10 Database strategy at scale?

3.11 How would you handle API rate limits?

3.12 What if Twilio API goes down?

3.13 What if DuckDuckGo Search fails?

3.14 How do you handle false positives?

3.15 What's your fallback strategy?

3.16 What's the monthly cost of running this bot?

3.17 How is it deployed?

3.18 What's the latency per check?

3.19 Is this deployed to real users? How many?

3.20 What would you do differently if rebuilding it today?

**Anti-Money Laundering System**

3.21 Walk me through the AML system.

3.22 What was the business problem?

3.23 Why 94% accuracy? What does that mean?

3.24 What about precision and recall?

3.25 Which matters more—precision or recall for fraud detection?

3.26 How did you handle class imbalance?

3.27 What features did you engineer?

3.28 What algorithm did you use and why?

3.29 How did you prevent data leakage?

3.30 How did you validate your model?

3.31 What was YOUR specific contribution vs others?

3.32 What was the hardest part for you?

3.33 Was this deployed in production?

3.34 How do you monitor for model drift?

3.35 What happens when accuracy degrades?

**AI Plagiarism Detection**

3.36 Tell me about the plagiarism detection project.

3.37 What problem does it solve?

3.38 Why build it?

3.39 Why RoBERTa instead of BERT or GPT?

3.40 How did you benchmark?

3.41 What were the trade-offs?

3.42 92% accuracy—what about false positives?

3.43 Can it detect paraphrased AI text?

3.44 Does it work on different languages?

3.45 Edge cases: code, technical writing?

3.46 Did you fine-tune RoBERTa or use pre-trained?

3.47 How long does inference take per essay?

3.48 What's your data pipeline?

3.49 Is it deployed? Real users?

3.50 ChatGPT keeps evolving. How do you handle that?

**Inverse Cooking**

3.51 Walk me through the Inverse Cooking project.

3.52 How does image-to-recipe generation work?

3.53 What's the architecture?

3.54 What CNN architecture did you use?

3.55 Did you use transfer learning?

3.56 How does T5 generate recipes?

3.57 Why combine vision + NLP?

3.58 You mention 85% accuracy. How did you measure it?

3.59 What does 85% accuracy mean for this use case?

3.60 Inference time?

3.61 What if image has multiple dishes?

3.62 What if image quality is poor?

3.63 Does it work for all cuisines?

**SECTION 4: CLOSING QUESTIONS**

4.1 Why are you interested in ANZ specifically?

4.2 What space in banking interests you most?

4.3 Why now? 4.4 Where do you see yourself in 5 years?

4.5 What do you want to become expert in?

4.6 Do you see yourself more as specialist or generalist?

4.7 What's one skill you want to develop in the next 2 years?

4.8 How do you stay current with technology?

4.9 Tell me about something new you learned in the last 3 months.

4.10 What do you find most challenging about ML/Data Science?

4.11 What's NOT your strength?

4.12 Why would you succeed in a fast-paced, high-pressure environment?

4.13 What's your salary expectation?

4.14 Are you willing to relocate?

4.15 When would you be available to start?

4.16 What questions do YOU have for us?