Part 1: Short Answer Questions (30 points)

AI Problem: Detecting Fake Job Postings Online

1. Problem Definition (6 points)

AI Problem:
Develop an AI system to detect fake job advertisements posted on online job platforms.

Objectives:

1. Flag suspicious job ads before they are published.
2. Reduce the number of job seekers scammed through fake ads.
3. Help job platforms maintain trustworthy listings.

Stakeholders:

- Job seekers
- Online job platform administrators

Key Performance Indicator (KPI):
Accuracy of the model in correctly classifying real vs. fake job ads.

2. Data Collection & Preprocessing (8 points)

Two Data Sources:

1. Labeled datasets of past job ads from online job platforms (e.g., Kaggle's Fake Job Postings dataset).
2. Web-scraped data from real-time job boards like LinkedIn, Indeed, or Glassdoor.

One Potential Bias:
If most training examples of fake jobs are from specific countries or sectors, the model may generalize poorly across regions or industries.

Preprocessing Steps:

1. Clean the job description text (remove HTML, symbols, and formatting).
2. Tokenize and normalize the text (lowercasing, removing stopwords, punctuation).
3. Encode categorical features like job type, location, and salary range.

3. Model Development (8 points)

Model Choice:
Logistic Regression for a baseline, or an XGBoost classifier for better performance on tabular + text data. XGBoost handles mixed feature types and missing data well.

Train/Validation/Test Split:

- 70% Training
- 15% Validation
- 15% Testing
  Split randomly while maintaining the ratio of real vs. fake labels (stratified split).

Two Hyperparameters to Tune:

1. max_depth – controls the complexity of the model (avoid overfitting).
2. learning_rate – affects how fast the model learns (balance speed and accuracy).

4. Evaluation & Deployment (8 points)

Evaluation Metrics:

1. **Precision** – important to avoid false positives (don't wrongly label legit jobs as scams).
2. **Recall** – important to catch as many fake job ads as possible.

What is Concept Drift?
Concept drift occurs when the patterns in the data change over time, for example, scammers might start writing more realistic job descriptions.

How to Monitor Concept Drift:

- Track accuracy and F1-score over time.
- Re-evaluate the model monthly with fresh data.
- Use data versioning and alert systems when performance drops.

One Technical Challenge During Deployment:
Handling high traffic and real-time predictions efficiently, the model should process thousands of job posts daily without lag, requiring a scalable API (e.g., Flask + Docker or cloud deployment via Azure).