Sr No	Category	XAIQB	Customized XAIQB Questions for Code Generation using Customization Prompt	Concern Category
1	Data	What kind of data was the system trained on?	What kinds of code artifacts (e.g., functions, scripts, projects) were used to train the model?	Training Data Characteristics
2	Data	What is the source of the training data?	Were the training code examples sourced from open-source repositories (e.g., GitHub, Stack Overflow), textbooks, or pr	Training Data Characteristics
3	Data	How were the labels/ground-truth produced?	For supervised training tasks (e.g., code summarization, translation), how were ground-truth labels or outputs defined a	Training Data Quality & Labeling
4	Data	What is the sample size of the training data?	How many code files or code snippets were used to train the model?	Training Data Characteristics
5	Data	What dataset(s) is the system NOT using?	Are there any known or commonly used code datasets (e.g., HumanEval, CodeSearchNet) that were explicitly excluded f	Training Data Characteristics
6	Data	What are the potential limitations/biases of the data?	Does the training data favor specific programming languages, styles, domains (e.g., web, systems), or coding conventio	Training Data Bias
7	Data	What is the size, proportion, or distribution of the training data with given feature(s)/feature-value(s)?	What is the distribution of code samples by language, code length, project type, or complexity in the training dataset?	Training Data Characteristics
8	Output	What kind of output does the system give?	What kind of code outputs can the model generate—functions, full programs, configuration files, documentation, or test cases?	Model Output Capabilities
9	Output	What does the system output mean?	How should I interpret the generated code—does it represent a complete, executable solution or just a code snippet meant to be extended or integrated?	Output Interpretation & Usage Context
10	Output	What is the scope of the system's capability? Can it do...?	What code-related tasks can the model handle (e.g., code translation, completion, debugging, refactoring), and where are its known limitations?	Model Capability & Limitations
11	Output	How is the output used for other system component(s)?	How can the generated code be integrated into my existing development pipeline or tools (e.g., compilers, test frameworks, CI/CD)?	Integration & Workflow
12	Output	How should I best utilize the output of the system?	What are the best practices for reviewing, testing, or modifying the generated code before deploying it in production or submitting it to version control?	Code Review & Validation Practices
13	Output	How should the output fit in my workflow?	How can this model-generated code be used efficiently within my software development workflow (e.g., prototyping, learning, accelerating repetitive tasks)?	Integration & Workflow
14	Performance	How accurate/precise/reliable are the predictions?	How accurate, syntactically correct, and semantically valid is the generated code? How reliable is it across different code tasks (e.g., translation, completion, test generation)?	Output Quality & Reliability
15	Performance	How often does the system make mistakes?	How frequently does the model produce incorrect, incomplete, insecure, or non-compiling code?	Output Quality & Reliability
16	Performance	In what situations is the system likely to be correct/ incorrect?	Under what input conditions or code tasks (e.g., simple utility functions vs. complex algorithms) is the model more or less likely to produce correct outputs?	Input Condition Sensitivity
17	Performance	What are the limitations of the system?	What are the known weaknesses of the model—e.g., poor handling of rare languages, advanced design patterns, or domain-specific APIs?	Model Capability & Limitations
18	Performance	What kind of mistakes is the system likely to make?	Does the model tend to introduce logic bugs, misuse libraries, ignore edge cases, or produce non-idiomatic code?	Output Quality & Reliability
19	Performance	Is the system's performance good enough for...?	Is the model suitable for use in production-level code, educational purposes, rapid prototyping, or legacy code migration?	Output Interpretation & Usage Context
20	How	• How does the system make predictions?	How does the neural network process source code or natural language input to generate corresponding code output?	Model Understanding Capabilities
21	How	• What features does the system consider?	What syntactic or semantic code features (e.g., keywords, function signatures, code structure) are important in guiding p	Model Understanding Capabilities
22	How	• Is [feature X] used or not used for the predictions?	Does the model consider [feature X], such as a specific variable, comment, or function call, in generating the output?	Model Understanding Capabilities
23	How	• What is the system's overall logic?	What are the high-level decision patterns or learned associations the model uses to predict code completions or transfo	Model Understanding Capabilities
24	How	• How does it weigh different features?	How much influence do different code tokens, AST nodes, or input types (e.g., comments vs. code) have on the model’s	Model Understanding Capabilities
25	How	• What kind of rules does it follow?	Are there implicit patterns or rules the model tends to replicate, such as standard library usage or code idioms?	Model Understanding Capabilities
26	How	• How does [feature X] impact its predictions?	If we modify or remove [feature X], how is the generated code affected?	Input Sensitivity & Feature Impact
27	How	• What are the top rules/features that determine its predictions?	What are the most influential input patterns, syntax structures, or tokens that affect generation behavior?	Model Understanding Capabilities
28	How	• What kind of algorithm is used?	What neural architecture underlies the model (e.g., Transformer, LSTM)? How does this affect its capacity to understand	Model Architecture & Design
29	How	• How were the parameters set?	What training strategies (e.g., fine-tuning, pretraining) and hyperparameter settings were used? What data was the mod	Training Methodology & Architecture
30	Why	Why/how is this instance given this prediction?	Why did the model generate this specific code snippet for the given prompt or partial code? How did it arrive at this particular output among possible alternatives?	Output Explanation & Reasoning
31	Why	What feature(s) of this instance determine the system's prediction of it?	What aspects of the input (e.g., function name, comments, code context, natural language intent) influenced the model’s code generation most?	Input Influence Explanation
32	Why	Why are [instance A and B] given the same prediction?	Why did the model generate the same or similar code for two different prompts or inputs (e.g., slightly different phrasing or variable names)? What common patterns led to this convergence?	Output Explanation & Reasoning
33	Why Not	Why is this instance NOT predicted to be [a different outcome Q]?	Why didn’t the model generate a more efficient (or idiomatic, or expected) version of the code? Why was a particular API or construct not used in the generated code?	Output Explanation & Reasoning
34	Why Not	Why is this instance predicted [P instead of a different outcome Q]?	Why did the model choose this code structure or algorithm over another possible one (e.g., iteration vs recursion, quicksort vs mergesort)?	Output Explanation & Reasoning
35	Why Not	Why are [instance A and B] given different predictions?	Why did similar input prompts/code snippets result in different outputs? What differences in input caused the model to generate distinct code?	Input Sensitivity & Output Variation
36	How to be that	How should this instance change to get a different prediction Q?	How should I rephrase the prompt or modify the code snippet to generate a different kind of implementation (e.g., use a different library, algorithm, or coding style)?	Prompt Refinement Control
37	How to be that	What is the minimum change required for this instance to get a different prediction Q?	What is the smallest modification to the input (e.g., parameter, comment, or keyword) needed to generate a significantly different version of code (e.g., with optimization, better readability, or fewer dependencies)?	Input Sensitivity & Variation Control
38	How to be that	How should a given feature change for this instance to get a different prediction Q?	How should I change a specific part of the input—like the function name, data type, or comment—for the model to generate code using a different method or abstraction?	Prompt Refinement Control
39	How to be that	What kind of instance is predicted of [a different outcome Q]?	What kind of prompt or code context typically leads the model to output more robust, idiomatic, or secure code compared to the one it gave now?	Input Specificity Requirements
40	How to still be this	What is the scope of change permitted for this instance to still get the same prediction?	What changes can I make to the prompt or input code (e.g., reordering comments, using synonyms, changing formatting) that still result in the same generated code or logic?	Input Specificity Requirements
41	How to still be this	What is the range of value permitted for a given feature for this prediction to stay the same?	How much can I vary elements like variable names, input types, or function structure before the model starts generating different code logic or structure?	Input Sensitivity & Variation Control
42	How to still be this	What is the necessary feature(s)/feature-value(s) present or absent to guarantee this prediction?	What specific parts of the input (e.g., algorithm name, data structure, comment cues) are critical for ensuring that the generated code performs the same operation or uses the same technique?	Input Sensitivity & Critical Input Parts
43	How to still be this	What kind of instance gets the same prediction?	What types of prompts or code snippets typically result in the same or similar generated code (e.g., same logic with different wording)?	Input Specificity Requirements
44	What if	What would the system predict if this instance changes to...?	What kind of code would be generated if I changed the input prompt slightly—such as modifying the problem description, function name, or input constraints?	Input Sensitivity & Variation Control
45	What if	What would the system predict if a given feature changes to...?	How would the generated code differ if I specified a different data structure (e.g., use a list instead of a dictionary) or requested an iterative solution instead of a recursive one?	Input Sensitivity & Variation Control
46	What if	What would the system predict for [a different instance]?	What kind of code will be generated if I input a completely different coding problem or ask for a solution in a different programming language?	Input Sensitivity & Variation Control
47	Others	How/why will the system change/adapt/improve/drift over time? *(change)*	How will the code generation model evolve over time—e.g., with updates, new training data, or fine-tuning? Will it improve in generating idiomatic code or adapting to newer libraries and language features?	Model Evolution & Updates
48	Others	Can I, and if so, how do I, improve the system? *(improvement)*	Can users or developers fine-tune or customize the model for a specific codebase, domain, or team coding standards? If yes, how can I do that safely and effectively?	Model Customization & Safety
49	Others	Why is the system using or not using a given algorithm/feature/rule/dataset? *(follow-up)*	Why did the model choose a particular algorithm (e.g., BFS vs DFS) or avoid using a specific feature or library in the generated code? What data or training influenced that choice?	Output Explanation & Reasoning
50	Others	What does [a machine learning terminology] mean? *(terminological)*	What does it mean when the model output explanation says “attention weight,” “activation,” “temperature,” or “beam search”? How does that relate to the generated code?	Model Explanation Terminology
51	Others	What are the results of other people using the system? *(social)*	What kind of experiences or typical outcomes have other developers had when using this model for code generation? Are there known issues, best practices, or communities sharing insights?	User Experience & Community Insights