CARE INDIA PROBLEM STATEMENT 1

1. **Title:** Code Search Utility

2. **Background**

A common challenge for development teams is the retrieval and reuse of specific code snippets that comply with internal technical guidelines. This can be time-consuming and error-prone, leading to inconsistencies and deviations from best practices. New developers might struggle to find the right pieces of code amidst vast and frequently updated repositories, which can slow down their productivity and contribute to a steeper learning curve.

To address this issue, the need arises for an intelligent tool that simplifies the process of finding relevant code snippets while ensuring these snippets adhere to the company's predefined technical standards and practices. Leveraging Natural Language Processing (NLP) technologies, such a tool can interpret natural language queries and return precise, contextually appropriate code implementations.

3. **Problem Statement**

Develop a chatbot utility that aids new developers in retrieving code snippets adhering to the company's custom technical instructions and practices by querying in natural language. The bot should fetch relevant code implementations from various source code repositories stored in Azure DevOps.

4. **Scope and Expectations**

A. Natural Language Processing (NLP)

- Implement NLP capabilities to understand and process developer queries related to code implementation.

B. Integration with Azure DevOps

- Connect and interact with Azure DevOps to access different source code repositories.
- Efficiently search across repositories to locate code snippets that match the user's query.

C. Custom Code Practice Recognition

- Recognize and retrieve code examples that conform to the company's custom wrappers and standard practices.

D. User Interface

- Develop a user-friendly chat interface for developers to interact with the bot seamlessly.
- Ensure the UI is intuitive and accessible within the developer's working environment, potentially through integration with IDEs or standalone web applications.

E. Response Accuracy and Relevance

- Ensure high accuracy in search results, providing the most relevant and useful code snippets.
- Implement mechanisms to continuously evolve and refine search algorithms based on feedback and user interaction.

F. Documentation and Guides

- Provide clear instructions and guides on how to use the chatbot, including examples of queries it can handle.
- Ensure the system is well-documented for maintenance and future enhancements.

5. Deliverables

a. A fully functional chatbot capable of natural language query processing with source code.

b. Integration with Azure DevOps for sourcing code snippets.

c. A responsive and intuitive chat interface.

d. Documentation and user guides for both developers and maintainers.

6. Key Outcomes

a. Streamlined onboarding process for new developers by providing quick access to relevant code snippets.

b. Increased efficiency in code implementation by reducing time spent searching through scattered repositories.

c. Enhanced adherence to company's custom technical practices and standards.

d. Also, propose ways to document such type of custom technical practices.

If any aspect of the problem statement is unclear or if additional details are needed, please ask.

CARE INDIA PROBLEM STATEMENT 2

Here's a structured template for creating a clear and focused problem statement for a technical hackathon:

---

1. Title: Senior citizen help alert to nearest nurse

---

2. Background: Patients living alone (especially senior citizens), when encounter an emergency situation due to accident or health vital changes or for any other reason need immediate support from the nurse. There are several nurses on duty for home visit, but they will have to figure out who will attend the call depending on their location. A geo-location aware solution will help health care management to locate the nearest nurse and assign visit to the patient's home.

---

3. Problem Statement: Develop a solution that identifies geographically nearest nurse for the given patient. The system should be able to identify live location of the patient and track live location of the nurses on the duty at the time of call. System then should be able to figure out nurse who can reach to the patient fastest based on the geo-distance and travel path. If first nurse contacted first does not respond to the call in given time frame, system should continue to contact next nearest nurse.

---

4. Requirements: The solution must:

- For the given city/region/zip, get the list of nurses on duty
- Track live location of the nurses
- When call initiated by the patient, determine the location of the patient
- Calculate and identify nurse who is nearest and can reach fastest and notify
- In case of no response continue to notify next available nurse

---

5. Constraints and Considerations:

Constraints:

- The solution should be developed within 48 hours.
- Notification must be in continuous manner until nurse accept/reject
- Ensure data privacy and security

Considerations:

- In addition to Geo-location, vehicle type and shortest path to figure out who can reach fastest.

---

6. Data and Resources: Participants can use the following resources:

- Map services

---

7. Deliverables: Participants must submit:

- A working prototype or MVP of prescription reading, scheduling and generating audio reminder
- Source code in a GitHub repository.
- A presentation or demo video (5 minutes max) explaining the solution.
- Documentation detailing how to run and test the solution.

---

8. Bonus Features (Optional): Bonus points will be awarded for:

CARE INDIA PROBLEM STATEMENT 3

- Here's a structured template for creating a clear and focused problem statement for a technical hackathon:

---

1. Title: Voice enabled medicines reminder

---

2. Background: Senior citizens, often have some regular medicines to take at specific time and it is very common that they forget due to aging and poor ability to remember. This is more prominent when they are living alone. A voice enabled assistant reminding each occurrence will help them taking medicines at specified time making treatment effective and resulting into better health condition

---

3. Problem Statement: Develop a solution that reads given prescription, identify different medicines along with its dosage, method and frequency and schedule for reminder. The system should be able to read standard prescription, identify method of taking the medicine and create schedule according to prescribe frequency and period. System then should integrate with voice enabled AI assistant such as Alexa and Google home or with equivalent audio device. System should use such device to remind each occurrence when a person needs to take medicine. The reminder should be in the form of audio that will describe which medicine to take and in what dosage.

---

4. Requirements: The solution must:

- Process standard prescription and extract details from it
- Schedule reminder for the prescribe frequency and period
- Integrate with audio enabled AI assistant or equivalent audio device
- Remind in audio format whenever there is a time to take medicine
- Capture person's meal time so that it remind well in advance for medicines that are to be taken before food
- Manage more than one person and include name while reminding

---

5. Constraints and Considerations:

Constraints:

- The solution should be developed within 48 hours.
- Reminder must be in audio format considering reaching out to senior citizens with limited capability
- Authentication to be able to use Alex (or any other) device if used

- Ensure data privacy and security

Considerations:

- Clarity for name of medicine, dosage and intake method
- Bluetooth as an alternative to voice enabled AI assistant

---

6. Data and Resources: Participants can use the following resources:

- Cloud platform like AWS or Azure
- Audio enabled device
- Sample prescriptions
- STT service

---

7. Deliverables: Participants must submit:

- A working prototype or MVP of prescription reading, scheduling and generating audio reminder
- Source code in a GitHub repository.
- A presentation or demo video (5 minutes max) explaining the solution.
- Documentation detailing how to run and test the solution.

---

8. Bonus Features (Optional): Bonus points will be awarded for:

- Get input from user in audio format and make it interactive

CARE INDIA PROBLEM STATEMENT 4

- Here's a structured template for creating a clear and focused problem statement for a technical hackathon:

---

1. Title: AI-Based Classification of Errors.

---

2. Background: In many industries, especially in software development and operations, identifying and classifying errors quickly is crucial to maintaining system performance and user satisfaction. Currently, many organizations rely on manual error tracking and analysis, which is time-consuming and prone to human error. By leveraging AI, we can automate the classification of errors, enabling faster resolutions and reducing downtime. This will not only improve operational efficiency but also enhance the user experience by minimizing disruptions.

---

3. Problem Statement: Develop an AI-based system that can automatically classify errors based on their type, severity, and impact. The system should analyse log files, error messages, or system behaviour to categorize errors and suggest possible resolutions. In addition, the system should be able to filter out non-critical logs, such as development or debug-level logs, and focus on identifying the actual problem-causing errors. The solution should be intuitive, providing clear classifications and actionable recommendations to help engineers resolve issues efficiently.

---

4. Requirements: The solution must:

- Analyze system log files or error reports to identify patterns.
- Classify errors by type (e.g., critical, non-critical, warnings).
- Suggest potential resolutions or next steps for error handling.
- Provide a user-friendly interface for viewing error classifications.
- Ensure scalability to handle large datasets of errors in real-time.

---

5. Constraints and Considerations:

Constraints:

- The solution should be developed within 48 hours.
- Use existing AI/ML models for error classification or develop custom models using available datasets.
- Ensure data privacy and security, particularly when working with sensitive system logs.

Considerations:

- Accuracy in classification and relevance of suggestions.
- Handling diverse types of errors from various domains (e.g., software, hardware).

---

6. Data and Resources: Participants can use the following resources:

- Open-source log datasets (e.g., Apache logs, Linux system logs).
- Error tracking APIs (e.g., Sentry, Datadog).
- Cloud platforms like AWS, Google Cloud, or Azure for model deployment.(To Do)
- Pre-trained models available via libraries like TensorFlow or PyTorch.

---

7. Deliverables: Participants must submit:

- A working prototype or MVP of the AI-based classification system.
- Source code in a GitHub repository.
- A presentation or demo video (5 minutes max) explaining the solution.
- Documentation detailing how to run and test the solution.

---

8. Bonus Features (Optional): Bonus points will be awarded for:

- Implementing real-time error classification and reporting.
- Providing an alert system that notifies engineers of critical errors.
- Integrating with popular error tracking tools (e.g., JIRA, Slack).
- Offering detailed analytics on error trends over time.

CARE INDIA PROBLEM STATEMENT 5

Title: DataGeN - AI Powered system to generate safe, secure, compliant and usable datasets

Background: Test Data is required for:

- **Software Development and Testing:**
    - Realistic data - To develop and Test application
    - Data Spectrum – Wide range of data needs for better coverage
    - Atomic test - Robust atomic tests
    - Data volume – To test application behavior under heavy data volume
    - Debug – To debug and reproduce production issues
- **Data Privacy and Compliance:** To comply with regulations such as General Data Protection Regulation (GDPR) by replacing personally identifiable information (PII) and Protected Health Information (PHI) with synthetic data or other non-sensitive data representations.
- **Secure Data Sharing:** Sharing data for the purpose of collaborations or outsourcing and other initiatives while reducing the risk of exposing sensitive information.

Problem Statement*:*

Various rudimentary approaches can be taken for test data generation and management like calling a post API's, running an API or Gui automation suite, calling a databases query or stored procedures etc. While in some simple application and areas these approaches works but in todays fast paced application development these can introduce inefficiencies in the development.

We need a much better data generation and management solutions which can generate data, which is safe, secure and can be scaled to a large volume while keeping integrity across tables and feature distribution.

Create an intelligent system which can create synthetic data which is scalable, secure, correct while maintaining the feature distribution.

Requirements for the minimum viable product (MVP) and additional requirements.

- The solution must provide features like:
    - De-identification of PII and PHI
    - Sub-setting
    - Scaling
    - System must be lightweight, performant, efficient and scalable

Constraints and Considerations:

- The solution should be built within 24 hours.

        - Use pre-trained AI models or create new models.

Data and Resources: Northwind Database

Deliverables:

- Participants must submit:
    - A working prototype or MVP.
    - Source code in a GitHub repository.
    - A brief presentation or demo video (5 minutes max) explaining the solution.
    - Documentation outlining how to run and test the solution.

CARE INDIA PROBLEM STATEMENT 6

Title: Test- Sense - AI Powered/Intelligent system to pick tests that matters

Background: Currently we are running entire test suite in CI Pipeline and sometimes otherwise. Which was a good approach to start with. As our system are becoming mature, we are expected to improve on test coverage meaning our test cases are increasing everyday and our execution times are increasing.

Problem Statement*: Design and develop an intelligent system that can be plugged with CI Pipelines (Azure-devops/Jenkins) which can pick and execute tests only for the areas which are directly or indirectly impacted. The tool should be intuitive, offer accurate selection, and provide explanation when sought on the selections it made for executions.*

Requirements for the minimum viable product (MVP) and additional requirements.

- The solution must:
  - Pick direct and indirectly impacted test cased
  - Provide clear explanation about its choices
  - Should be easy to integrate with CI systems
  - System must be lightweight, performant, efficient and scalable

Constraints and Considerations:

- The solution should be built within 24 hours.
- Use pre-trained AI models or create new models using open datasets.

Data and Resources: TBD

Deliverables:

- Participants must submit:
  - A working prototype or MVP.
  - Source code in a GitHub repository.
  - A brief presentation or demo video (5 minutes max) explaining the solution.
  - Documentation outlining how to run and test the solution.

Bonus Features (Optional):

Bonus points will be awarded for:

- GUI to understand code change and test traceability

FINTECH PROBLEM STATEMENT 1

Title: Mobile Game for Personal Finance Education and Investment

Challenge: Children often lack engagement with personal finance education, which limits their understanding of saving, investing, and financial management from an early age. Mobile games present a unique opportunity to make personal finance fun and interactive. The challenge is to create a free-to-play mobile game that not only educates children about personal finance but also integrates real-world financial tools to help them start saving and investing.

Objective: Develop a mobile game that engages children in learning about personal finance through gameplay. The game should:

- Be free-to-play, with optional in-app purchases.
- Channel proceeds from in-app purchases into an investment vehicle or personal savings account.
- Once a certain amount of funds is accumulated, provide players (or their guardians) with tools to reinvest and manage these funds, fostering real-life financial responsibility.

Key Considerations:

- How will the game make financial concepts accessible and engaging for children?
- How can the proceeds of in-app purchases be securely managed and allocated to investment or savings accounts?
- What financial tools and features can be provided to help users reinvest and manage their funds once they reach certain thresholds?
- How will the platform ensure security, transparency, and legal compliance when handling real-world money and investments?

Expected Outcome:

- Engaging Financial Education: The game will teach children personal finance concepts like saving, investing, and budgeting through interactive and fun gameplay, making learning enjoyable.
- Real-World Financial Integration: Players can manage proceeds from in-app purchases within secure savings or investment accounts, under the supervision of their guardians.
- Gamified Learning: The game will include rewards, achievements, and levels to motivate players and enhance their understanding of financial responsibility.
- Reinvestment Tools: Once certain financial thresholds are met, players will unlock advanced tools to reinvest or manage their funds, promoting long-term financial habits.
- Parental Controls: Guardians will have oversight of real-world fund management and the ability to control reinvestment decisions and spending limits.
- Technical Infrastructure: The game will be built on a scalable platform, ensuring secure transactions, regulatory compliance, and integration with financial services APIs.

FINTECH PROBLEM STATEMENT 2

Challenge : You work for a consumer finance company which specializes in lending various types of loans to customers. When the company receives a loan application the company has to make a decision for loan approval based on the applicant's profile.

There are 2 types of risks are associated with this decision-

1. If applicant is likely to repay the loan, then not approving the loan results in a loss of business to the company

2. If applicant is not likely to repay the loan i.e. the applicant defaults, which means /may lead to financial loss for the company.

Objective: Create an algorithm using ML/AI for analyzing loan applicant data to identify patterns that indicate the likelihood of default. The consumer finance company wants to minimize risks and losses from loans by better understanding how customer and loan attributes influence the chance of defaults. The analysis should use EDA techniques to determine which variables are strong predictors of default so that high-risk applicants can be denied loans or given higher interest rates to reduce credit losses from defaults.

Expected Outcome:

- Explain the dataset used for the problem statement. Summarize the dataset in detail.
- Identify attributes and factors which will help your model to decide the loan disbursement to an applicant
- Create a ML AI algorithm for analyzing loan applicant data to identify patterns that indicate the likelihood of default and explain why according to you that is correct algorithm. What are the considerations behind it.

FINTECH PROBLEM STATEMENT 3

Challenge: Detection of Fraud transactions when coming from sensitive countries etc.

Financial institutions need to optimise the monitoring of fraudulent transactions and identify potential suspicious activity during transaction processing.

Objective: Develop a machine learning application that would identify behavioural patterns.

For instance, transactions originating from geographically distant locations, multiple transactions are already understood to be highly correlated to suspicious activity. There are certainly many more tell-tale signs that could be used to detect potentially fraudulent behavior,

Expected Outcome:

- Develop a machine learning application that would identify behavioural patterns. For instance, transactions originating from geographically distant locations, multiple transactions are already understood to be highly correlated to suspicious activity.
- There are certainly many more tell-tale signs that could be used to detect potentially fraudulent behaviour. Explain these attributes based on which the algorithm decides potentially fraudulent behaviour.

FINTECH PROBLEM STATEMENT 4

Title: Sentiment analysis for a stock

Challenge: Retail investors often rely on news headlines for making stock market decisions. However, gauging sentiment from large volumes of news can be challenging

Objective: Build an AI-based sentiment analysis tool that analyzes stock market news and provides sentiment scores (positive, negative, neutral) to help investors make informed decisions.

- Sentiment analysis for a particular stock by doing text mining of capital markets portals/sites like money control, ET, twitter etc.
- Sentiments can be categorized as bullish, bearish, and neutral
- Time horizon segregation of sentiments on quarterly basis

Key Considerations:

- Real-time analysis of financial news.
- Sentiment scoring for specific stocks or sectors.
- Keyword-based search for news trends.
- Visualization of sentiment trends over time.

Expected Outcome: When particular stock is searched in out web page/ app it should present the sentiment analysis in tabular format or in graphical format with its sentiments (positive, negative, neutral) against the time(quarter).