List of problem statements for NIT-K Hackathon (Oct-2024)

General Instructions:

Below are general instructions applicable to all problem statements.

- Create a repo under the **nitkhackathon2024** GitHub Organisation. Use the template repository provided which contains the expected folder structure of the submission
    - Mandatory: Demo video (5-7 minutes) is mandatory for all submissions (screen capture + voice over). In the video explain your problem statement, your solution approach, demo your working solution & walkthrough the code)
    - Mandatory: Include a README.md file at the root of your submission. An example README will be provided when creating the repo using the template. The README should include at the minimum a basic drawing of the solution architecture & instructions on how to get the project working in new machine. Make sure to mention any pre-requisites or dependencies for your app.
    - Commit all your code into “code” folder of your repo
- Make sure that you do not commit any secrets/passwords into the git repository
- Explicitly mention the usage of Generative AI tools in your solution
- No commits allowed after cut-off time (disqualify team if any commit is made after cut-off time)

1. Theme: Improving Software Development

1.1 Problem Statement: Software Design Pattern Detector

Introduction:

Design patterns are fundamental to creating scalable and maintainable software. Your challenge is to develop an IDE plugin that can detect specific design patterns in a piece of code. This tool will assist developers in identifying and understanding the patterns used in existing codebases, improving code readability and maintenance.

The plugin must analyse code in real-time within the IDE, identifying predefined design patterns like for example; Singleton, Factory, or Observer. For the purpose of the hackathon, the number of patterns to be detected will be limited to a manageable set.

Participants must implement algorithms that can parse and analyse the structure of the code, recognizing patterns based on their specific characteristics. The plugin should work efficiently without slowing down the IDE, providing instant feedback to developers.

The solution should include a user interface integrated into the IDE, where detected patterns are highlighted in the code. The plugin should also provide explanations of the detected patterns, helping developers understand their implementation and usage.

The effectiveness of the plugin will be evaluated based on accuracy, the number of patterns correctly detected, and the user experience within the IDE. Extra points will be given for detecting patterns in complex code structures and providing insightful pattern summaries.

Expected Solution:

- Participants must submit their solutions as a fully functional IDE plugin, with detailed documentation on the implementation and usage of the tool within the IDE. If there are time constraints with developing a fully functional IDE plugin, a proper web UI that mimics the plugin must be developed.

Bonus Features(Nice-to-haves):

1. **Multiple Language Support:** Extend the plugin to detect design patterns in multiple programming languages, enhancing its versatility and applicability.
2. **Custom Pattern Detection:** Allow users to define and add their own patterns to be detected by the plugin, making it adaptable to various coding styles and frameworks.
3. **Pattern Refactoring Suggestions:** Implement a feature that not only detects patterns but also suggests refactoring or improvements based on best practices.

Notes:

- Preferred programming languages – Java/Python/JavaScript
- You are free to use any publicly available code for your demo to detect design patterns

- VS Code / IntelliJ IDEA IDE is preferred

2. Theme: Improving Customer Experience in Banking

2.1 Problem Statement: Sign Language Translator for Virtual Meetings

Introduction:

Enhancing accessibility in virtual meetings is essential to fostering inclusive communication. Your challenge is to develop a plugin for popular meeting platforms like Microsoft Teams or Zoom that translates spoken conversations into sign language in real-time. This plugin will enable participants who are deaf or hard of hearing to engage fully in conversations by providing a seamless translation of spoken words into sign language.

User Interaction: The plugin should integrate smoothly with the existing meeting apps, allowing users to activate the sign language translation feature with a simple click. The plugin will display a sign language interpreter alongside the video feed or within a designated panel, translating the conversation into sign language for users who prefer or require it.

Real-Time Translation: Participants must implement real-time speech recognition and sign language translation, converting spoken words into text and then into sign language animations or video representations. The translation should be accurate and synchronized with the conversation to ensure clarity and comprehension.

Expected Solution:

- Seamlessly integrate with Teams or Zoom, providing an easy-to-use interface for users to enable or disable the sign language translation feature.
- Utilize open-source speech-to-text tools to capture and transcribe spoken language in real-time.
- Convert the transcribed text into sign language using animated avatars or pre-recorded videos.
- Present the sign language translation in a non-intrusive manner, such as a sidebar or overlay, ensuring it does not obstruct the main video feed.
- If you cannot build a Teams/Zoom plugin due to time constraints, a fully functional web UI can be submitted

Bonus Features(Nice-to-haves):

1. Multiple Sign Language Support: Extend the plugin to support various sign languages (e.g., American Sign Language, British Sign Language) to cater to a diverse user base.
2. Adaptive Learning: Implement machine learning algorithms that allow the plugin to improve translation accuracy based on user feedback and usage patterns.
3. Interactive Features: Add functionalities such as clickable signs for definitions, the ability to pause and replay translations, or integrating gesture recognition for more interactive communication.

Notes:

- You are free to use any audio / video file of your choice for input
- Assume only English language will be used for translation

2.2 Problem Statement: Financial Literacy App for Young People

Introduction:

In today's fast-paced financial world, young people need the right tools and knowledge to manage their finances effectively. Your challenge is to develop a financial literacy app tailored to young people in the USA, reflecting Wells Fargo's way of providing banking services. The app will introduce users to fundamental financial concepts such as budgeting, saving, managing credit, and investing, while aligning with the services and tone that Wells Fargo offers.

The app must cater to young users by offering engaging, bite-sized lessons and interactive content on personal finance. These lessons should reflect Wells Fargo's trusted approach to financial management, integrating educational tools that mirror real-world banking scenarios like checking accounts, credit cards, and loans.

To make the app more appealing to younger users, the content should be highly interactive, featuring quizzes, real-world simulations, and goal-setting tools to encourage users to build healthy financial habits. The app should also provide customized advice, suggesting relevant Wells Fargo services such as savings accounts, student loans, or investment products based on the user's profile and progress.

Expected Solution:

Participants must submit a fully functional app, including:

- Clear navigation through the financial literacy lessons and interactions with simulated Wells Fargo services
- At least one implementation of gamification/quiz
- Maintain a daily/weekly/all-time leaderboard

Bonus Features(Nice-to-haves):

1. **Gamification:** Incorporate gaming elements like rewards, badges, or leaderboards to motivate users to complete financial lessons and tasks.
2. **Parental Controls:** Allow parents to monitor their child's financial education progress and assist with goal setting and savings plans.
3. **Real-Time Financial Tips:** Implement a feature that provides real-time tips or updates on banking options based on changes in the user's financial behaviour or spending patterns.

Notes:

The participants are allowed to create an app with features they prefer. However, an example implementation can be as below

Your team of 3, can create an interactive game that teaches financial literacy by guiding players through different life stages, incorporating Wells Fargo's financial products and services. The game will follow a character from age 12 to 45, presenting them with financial decisions involving savings, loans, credit management, and investments. Wells Fargo products like savings accounts, credit cards, home loans, and investment tools will play an essential role in helping the player achieve financial success.

Life Phases & Financial Integration: The game will include five life phases, with each stage introducing new financial challenges and corresponding Wells Fargo products:

1. Phase 1: Age 12 - 18 (Teenage Years)
    - Earnings: Pocket money, part-time jobs.
    - Wells Fargo Product: Way2Save® Savings Account—Players will learn how small savings and automatic transfers from their checking account can build a financial safety net.
    - Pitfalls: Spending on unnecessary items and failing to start saving early.
2. Phase 2: Age 18 - 22 (College/Young Adult)
    - Earnings: Part-time jobs, internships.
    - Wells Fargo Product: Student Loan—Introduce the concept of financing education through student loans. The game will simulate paying off student debt while balancing living expenses.
    - Pitfalls: Taking on too much student debt or mismanaging credit.
3. Phase 3: Age 22 - 30 (Early Career)
    - Earnings: Full-time job salary.
    - Wells Fargo Product: Everyday Checking & Cash Back Credit Cards—Teach players to manage credit responsibly, avoid overdraft fees, and benefit from cash back on everyday purchases.
    - Pitfalls: Misusing credit cards and failing to manage monthly expenses.
4. Phase 4: Age 30 - 40 (Mid-Career)
    - Earnings: Higher salary, bonuses.
    - Wells Fargo Product: Home Mortgage—Players will learn to apply for and manage a mortgage, understanding down payments, interest rates, and the importance of good credit.
    - Pitfalls: Overextending with a large mortgage or neglecting long-term investments.
5. Phase 5: Age 40 - 45 (Established Career)
    - Earnings: Salary, passive income from investments.
    - Wells Fargo Product: Retirement Planning & Investments (IRAs, CDs)—Teach players about investment options for retirement, balancing between Wells Trade® accounts and Platinum Savings to maximize long-term wealth.
    - Pitfalls: Not saving enough for retirement or making risky investments late in life.

Example Game Mechanics:

1. Financial Decisions: Players will make financial decisions in each phase of life, from choosing savings accounts to managing loans and credit. The game should simulate real-life scenarios using Wells Fargo products to achieve financial goals.
2. Educational Value: Players will receive brief tips on financial literacy, such as the importance of savings accounts, avoiding debt, and investing wisely.
3. Level Progression: As players progress through the levels, they will need to use lessons learned in earlier phases to make better financial decisions in later stages, all while utilizing Wells Fargo's product offerings.

Example Add-ons:

1. Dynamic Market Conditions: Introduce fluctuating interest rates or market downturns that impact loans and investments, adding real-world complexity to financial planning.
2. Family & Career Planning: Add a feature where players can simulate major life decisions such as having children or changing jobs and see how it affects their financial status.
3. Financial Advice Mode: Include a financial advisor character (modelled after Wells Fargo's advisory services) who gives hints on how to best manage money, investments, and retirement planning.

2.3 Problem Statement: AI-Based User Behaviour Prediction for Mobile App Features

Introduction:
In the banking industry, understanding how users will react to new mobile app features is crucial for successful product launches. Your challenge is to develop an AI-based tool that predicts how different user personas of a banking mobile app will respond to new feature introductions. By analysing historical user behaviour, preferences, and interactions, the model will simulate persona-specific reactions, providing insights into potential adoption, engagement, and friction points without needing to contact real customers. The key requirements are

1. Feature Input:
    - The tool should allow product managers to input the details of a new feature (e.g., functionality, usability, UI changes, notifications).
2. Behaviour Prediction:
    - Based on user personas (which are created from historical user data), the model should predict how each persona would interact with the new feature. This includes adoption rate, engagement frequency, and likely challenges.
3. Persona Reactions:
    - Simulate various user persona reactions, such as early adopters, infrequent users, or tech-savvy users, predicting how each group will respond to the new feature.
4. Output Insights:
    - Provide clear, actionable insights to the product manager, including:
        - Engagement Scores: Predict how likely each persona is to engage with the new feature.
        - Potential Risks: Highlight areas where user frustration or dissatisfaction may arise.
        - Adoption Forecast: Estimate the percentage of users within each persona that will adopt the feature.

Expected Solution:
Participants must build a machine learning model capable of predicting user reactions based on predefined user personas and feature inputs. The solution should simulate how different personas will interact with the feature, focusing on:

- **Feature Adoption**: Whether users will adopt the new feature.
- **Engagement Levels**: How frequently users will engage with the feature.
- **Potential Pain Points**: Predicting areas where users might experience friction or dissatisfaction with the feature.

Bonus Features(Nice-to-haves):

1. **Persona Customization**: Allow product managers to create custom personas based on specific traits (e.g., income, age, transaction frequency) to see how these customized personas might react to the feature.

2. **Market Trends Integration**: Incorporate external factors such as market trends, customer sentiment analysis, or economic conditions to enhance the accuracy of the persona reactions.
3. **Real-time Updates**: Enable continuous learning where the model refines predictions based on real-time data and actual customer interactions with previous feature launches.

Notes:

- Any of the below features for a net banking app can be picked for solutioning and predicting the persona behaviour.

1. AI-Powered Smart Financial Assistant

This feature would act as a **personal financial advisor** that helps users manage their finances more effectively by providing **customized insights, recommendations, and automation**. The assistant would analyze the user's spending habits, income patterns, and savings goals to offer personalized suggestions, such as budgeting tips, investment opportunities, and bill payment reminders.

2. Card Management Hub with Virtual Cards and Spend Limits

This feature would give users more control over their debit and credit cards through an integrated **Card Management Hub** that allows for the creation of **virtual cards**, setting **spend limits and** offering **real-time transaction control**

Personas: Below are the examples of a net banking app user personas. This is just for reference, more personas can be added as required.

Summary of Net Banking Personas:

1. **Tech-Savvy Millennial**: Focused on seamless mobile banking and smart financial tools

2. **Busy Professional**: Requires efficient financial management and time-saving automation

3. **Small Business Owner**: Needs specialized business banking services and cash flow management

4. **Retiree/Conservative Investor**: Prefers simple, secure banking with a focus on safety and income

5. **Frequent Traveler/Expat**: Needs global banking features and easy international access

6. **New-to-Banking/Rural User**: Requires basic banking with easy accessibility and regional language support

7. **High-Net-Worth Individual (HNI)**: Focused on premium banking and wealth management services.

3. Theme: Improving Work Efficiency

3.1 Problem Statement: Knowledge Distiller App

Introduction:

In the fast-paced world of academia, students often struggle with organizing and retrieving study materials. Your challenge is to develop an app that allows users to load their study notes—whether text, images, or audio—and automatically generates a personal knowledge graph/ highlight. This graph will help students quickly find and connect key concepts, making study sessions more efficient and tailored to their learning style.

The app must support various input types, including text documents, handwritten notes, images, and audio recordings. The app should effectively parse these different formats and extract meaningful information to construct the knowledge graph.

Expected Solution:

The app should analyse the inputted study materials, identify key concepts, and create a dynamic, expanding knowledge graph that visually represents the relationships between different topics. This graph should be continuously updated as new notes are added.

The solution should include an intuitive user interface where students can easily upload their study materials and view their evolving knowledge graph. The app should allow users to explore the graph, search for specific concepts, and retrieve related study content quickly.

Bonus Features(Nice-to-haves):

1. **Collaboration Features:** Allow users to share their knowledge graphs with peers, enabling collaborative study and knowledge exchange.
2. **Adaptive Learning Suggestions:** Implement a feature that suggests additional study resources or concepts based on the gaps or weaknesses identified in the user's knowledge graph.
3. **Integration with Learning Platforms:** Enable the app to integrate with existing educational platforms (like LMS) to automatically pull in course materials and update the knowledge graph.
4. **Tagging & Summarization:** Bonus points may be awarded for innovative features like automatic tagging, summarization, or cross-referencing between different subjects.

3.2 Problem Statement: Intelligent Peer Learning Platform

Introduction:
In today's rapidly evolving educational landscape, personalized learning and peer collaboration have become increasingly important. Your challenge is to develop an innovative platform that facilitates peer-to-peer learning by matching students based on their complementary skills and learning needs.

This platform will empower students to both teach and learn from each other, creating a dynamic and engaging educational ecosystem. The core platform must implement a matching algorithm that considers factors such as academic strengths and weaknesses and learning preferences to create peer connections. It should provide a virtual collaborative workspace where matched peers can share resources and work together on projects. Additionally, the platform should incorporate a skill tracking system to monitor learners' progress over time.

Expected Solution:

The expected solution should be a functional web-based platform that enables peer-to-peer learning. Users should be able to register and create profiles, inputting their skills, strengths, and areas for improvement. One of the key aspects the solution is the matching algorithm, the algorithm should efficiently pair users based on complementary skills and learning needs, its effectiveness will be a key evaluation criterion. The platform should feature a basic chat system for matched peers to communicate in real-time and share files. Ideally, users should also have the ability to track their skills and improvements over time through a simple self-reporting system, visualized through charts or progress bars.

The user interface should be intuitive and user-friendly, ensuring smooth navigation between profile viewing, peer matching, and chat functionalities. Additional features could include resume parsing for profile auto-creation or integration of screen sharing capabilities.

Bonus Features(Nice-to-haves):

1. **Gamification Elements**: Implement game-like features such as challenges, achievements, and leaderboards to encourage continuous engagement and motivation.

2. **Reputation System:** Develop a basic reputation and endorsement system that recognizes top contributors and tutors within the platform.

3. **Skill Recommendation:** Create a simple recommendation system that suggests new skills for users to learn or teach based on their current proficiencies.

4. **Visual Skill Progression:** Design a clear visual representation of users' skill progression over time.

4. Theme: Improving Banking, Integrated apps

4.1 Problem Statement: Build an app using Beckn Protocol

Introduction:

The Beckn Protocol is an open-source standard enabling seamless interoperability between decentralized digital commerce networks. Your challenge is to develop a Beckn-compliant application that leverages this protocol to connect buyers and sellers in a decentralized marketplace, ensuring seamless transactions across multiple platforms. Introduction - Beckn (becknprotocol.io). The key requirements are

- **Discovery and Matching:** Enable buyers to discover products or services based on defined criteria (e.g., location, pricing).
- **Order Management:** Allow users to place orders, track them, and facilitate payment using the Beckn protocol.
- **Interoperability:** Ensure that the app can interact with multiple networks and service providers while adhering to the Beckn protocol specifications.

Expected Solution:

Participants must create an application that:

1. Implements the Beckn protocol for buyer and seller interactions.
2. Allows real-time discovery, order placement, fulfilment, and tracking of services or goods.
3. Demonstrates interoperability between two or more networks (e.g., multiple vendors offering different services).

Bonus Features(Nice-to-haves):

1. **Multi-Language Support:** Extend the application to support multiple languages, ensuring wider accessibility.
2. **Service Recommendations:** Add personalized service recommendations based on user preferences and past transactions.
3. **Cross-Network Payment Integration:** Enable seamless payment handling across different service providers and platforms.

Notes:

- https://blog.becknprotocol.io/a-comprehensive-tech-guide-to-beckn-by-tarka-labs

4.2 Problem Statement: Real-time equity analysis

Problem Statement:
As global investors increasingly prioritize sustainability and responsible investing, access to reliable, real-time data has become crucial in making informed financial decisions. With the growing importance of environmental, social, and governance (ESG) factors, alongside the rapid changes in market sentiment driven by social media and news platforms, there is a need for an innovative tool that consolidates these insights into a single, easy-to-use platform. The key requirements are:

- A **user-friendly application** that delivers real-time sustainability and market sentiment data from credible sources, helping investors stay informed about the equities they follow.
- **Enhanced decision-making capabilities** for investors, made possible by an interactive dashboard that integrates diverse data analytics, giving them a holistic view of stock performance.
- **Scalability for future enhancements**, ensuring that the application can easily integrate additional data sources, features, or asset classes as investment needs evolve.

Expected Solution:
Develop a fully functional mobile/web real-time equity analysis application that equips investors with critical data on specific equities, focusing on sustainability scores, market sentiment Indicators like Beta and Relative strength Index, and other key financial indicators like Price-to-Earnings(P/E) Ratio, Price-to-Book(P/B) Ratio. The application should source and aggregate data from authorized platforms and APIs, offering users actionable insights to enhance their investment decision-making process.

Notes:

1. This application should cater to the US Stock Market only

4.3 Problem Statement: Smart document processing and form filling system

Introduction:
Develop a versatile system that can efficiently verify, process, and auto-fill a wide variety of documents and forms across multiple industries. The core system should be able to:

- Analyze and extract information from common document types (e.g., ID cards, utility bills).
- Verify the authenticity and validity of submitted documents.
- Auto-fill forms based on extracted information and user input.
- Provide a user-friendly interface for document submission and form filling.

Expected Solution:

- Build a fully functional web/mobile application that can allow users to capture and upload documents
- The application must be able to identify and extract all information from the uploaded documents
- The application must be able to support various types and formats of documents irrespective of quality / alignment
- Implement a feedback mechanism e.g., intimate user if document is of very poor quality and cannot be read

Bonus Features(Nice-to-haves):

- Adapt to new document types and form structures with minimal reconfiguration.
- Generate confidence scores for the accuracy of filled forms and verified documents.
- Implement robust data privacy and security measures.
- Create an API for easy integration with existing systems across different sectors.

Notes:

- Avoid using sensitive documents for solutioning
- Images and PDFs can be used for demo to begin with

5. Theme: Quantum Computing

5.1 Problem Statement: Quantum Detective: Cracking Financial Anomalies

Introduction

Financial transactions generate massive and complex datasets where unusual patterns or anomalies, such as fraudulent activities or errors, can easily remain hidden. Traditional machine learning models often face challenges with these datasets due to their size and complexity, resulting in slower processing times and less accurate results. Quantum computing offers a promising approach to address these challenges by leveraging quantum properties to analyze large datasets more effectively.

You are tasked with building a quantum machine learning model to detect anomalies in financial transactions using the "**Credit Card Transaction Anomalies**" dataset from Kaggle. Your solution should explore the use of quantum algorithms and tools like Qiskit (Ref. IBM Quantum Learning) or PennyLane (Ref. PennyLane) to develop an innovative approach. The goal is to design a quantum model that demonstrates improvements in speed, scalability, and accuracy compared to classical methods or existing quantum algorithms. Participants are encouraged to experiment with various encoding techniques, model architecture and to provide a thorough analysis demonstrating the advantages of their approach.

Expected Solution:

- Development of a quantum-based model for anomaly detection in financial datasets using a unique approach or algorithm to provide a measurable advantage over classical methods.
- Comparative analysis showcasing improvements in speed, scalability, and accuracy.
- A clear demonstration of how quantum computing techniques outperform classical models in the context of financial anomaly detection.
- Jupyter Notebook or Python code is expected to be submitted

Dataset to be used:

- Dataset: Start with the "Credit Card Transaction Anomalies" dataset from Kaggle (Credit Card Fraud Dataset). https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud
- However, participants are encouraged to explore and incorporate more complex datasets to demonstrate the potential quantum advantage.