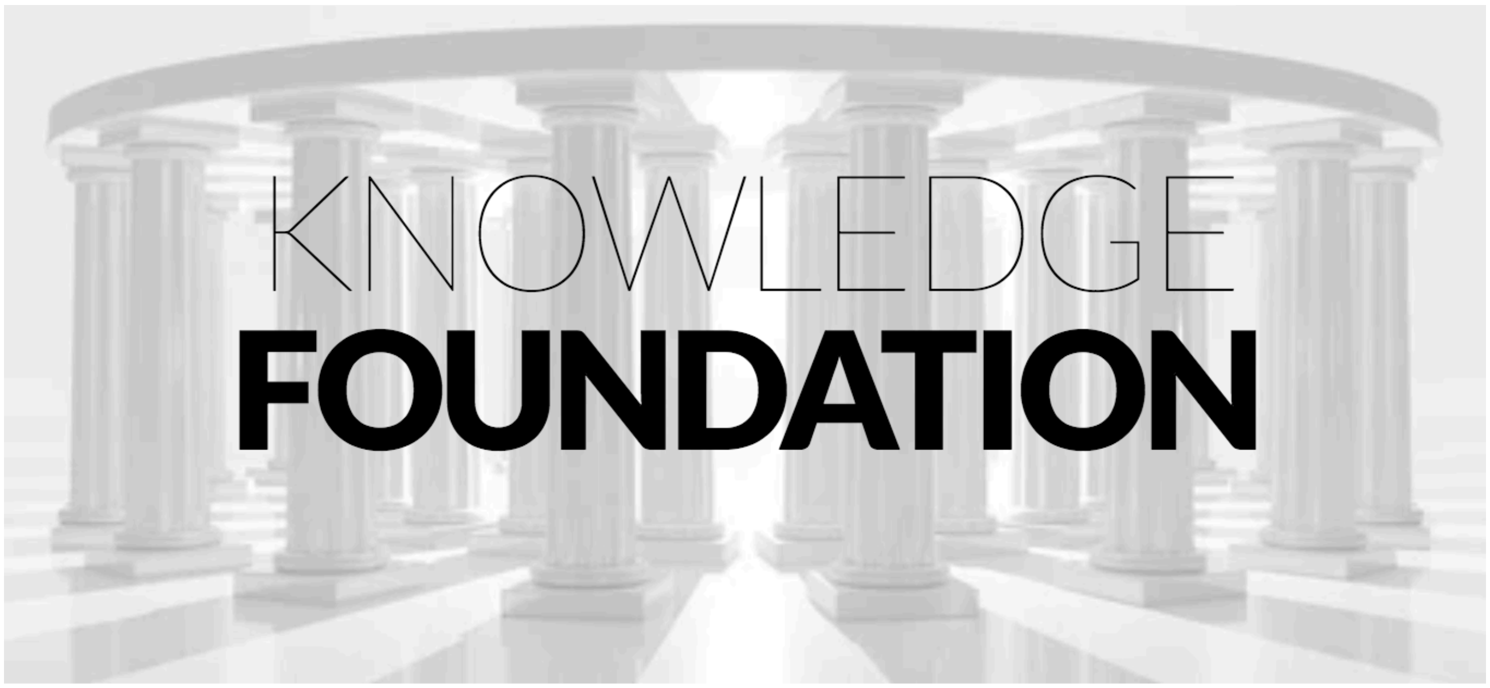


# KNOWLEDGE FOUNDATION

Knowledge Foundation : https://KnowledgeFound.org | Knowledge Token® : https://KnowledgeFound.org/token

## COLLABORATORS

- Immersive Education Initiative (USA)
- Lucerne University of Applied Sciences and Arts (Switzerland)
- Massachusetts Institute of Technology (MIT) Bitcoin Club (USA)
- University of Oxford Blockchain Research Centre (UK)
- University of Zurich Blockchain Center (Switzerland)
- Yale University Blockchain Club (USA)
- Brown University Blockchain Club (USA)

## INTERNATIONAL PARTNERS

- United Nations Global Resource for Anti-Corruption Education and Youth Empowerment (GRACE) initiative

- United Nations Office on Drugs and Crime (UNODC)

# 2024 OXFORD, UK

## Coding4Integrity HACKATHON

**Oxford University | April 09 - 11**

The 2024 Blockchain in Education Summit and Digital Civilisation conference come together for a special joint event at Oriel College, Oxford University, from 09 to 11 April in collaboration with the **Knowledge Foundation** and **University College Oxford Blockchain Research Centre**.

# Table of Contents

## UNITED NATIONS Coding4Integrity HACKATHON SERIES

The 2024 Oxford hackathon is the first in a new 3-year series of United Nations Office on Drugs and Crime (UNODC) Coding4Integrity hackathons made possible through the partnership of UNODC, the UNODC Global Resource for Anti-Corruption Education and Youth Empowerment (GRACE) Initiative, Knowledge Foundation and DFINITY Foundation.

# 1. AWARDS / PRIZES

Join us in person at Oxford University or online from April 09 to 11 to compete in the 2024 Blockchain in Education Summit Hackathon for prizes and prestige.

Developer grant prizes are awarded by the DFINITY Foundation in collaboration with the Knowledge Foundation. The grants are used to fund participation (after the hackathon) in developing Knowledge Token® on the Internet Computer platform and will be distributed after the hackathon has ended.

The Knowledge Token® prizes will be awarded to the winning teams and solo participants and are redeemable towards registration fees at all future Summits and Symposia worldwide.

*Award (prize) details in on the Hackathon area of the official event website*

# 2. REQUIREMENTS

## 2.1 OPEN SOURCE

All submissions (entries) for this hackathon will be made available to the public in a free and open manner, and are therefore automatically considered Open Source upon submission. Only Open Source submissions will be considered.

## 2.2 SUBMITTING YOUR SOLUTION (ENTRY)

At the conclusion of the hackathon you will submit your solution (hackathon entry) using an online form, at which time you will be required to provide:

1. A link to your source code. For this you can use a public GitHub repository, a public website folder, or any other public online code repository.

2. A link to your solution running on the Internet Computer main network ("mainnet"). In order to do this you must deploy your solution to an Internet Computer canister on the mainnet and "fuel" your canister using "cycles" (see "FREE CYCLES" below).

3. A link to your design documents, if you created such materials. Note that design documents are not required, but are recommended.

4. The name and email address of each member of your team (or yourself, if you are competing Solo as an individual and not on a team).

*See SUBMITTING YOUR SOLUTION below for more details.*

## 2.3 BLOCKCHAIN PLATFORM

Only hackathon entries (submissions) created/programmed using the open source Internet Computer will be considered.

Internet Computer is a relatively new blockchain platform for which you will find the following resources important to become familiar with this platform in time to compete:

1. INTERNET COMPUTER: https://internetcomputer.org
3. IC DASHBOARD: https://dashboard.internetcomputer.org

*See DEVELOPER RESOURCES below for Internet Computer developer documentation, videos and support resources.*

## 2.4 PROGRAMMING LANGUAGES

Although **Motoko** is preferred, your hackathon entry (submission) may be programmed in any language supported by the Internet Computer.

*Note that Motoko is preferred but not required (see JUDGING PREFERENCES).*

## 2.5 ANONYMOUS SUBMISSIONS: NO NAMES OR BRANDING

Note that judging is conducted in a manner intended to be unbiased and fair. To this end "blind judging" is the process by which hackathon solutions (entries) are reviewed and judged, meaning the judges do not know whose work they are judging, they don't know who the other judges are, and they don't share or publish their scores. To ensure anonymity, be very careful to remove your names, your organization (school, company, etc.) names and logos, and any other identifying information from your solution front-end (user interface).

## 2.5 FREE CYCLES

At the conclusion of the hackathon you will submit a link to your program running on the Internet Computer main network (mainnet). To do this you must deploy your solution to the Internet Computer mainnet and fuel your canister with cycles.

You can obtain free cycles via the **Cycles Faucet** detailed in the official Internet Computer developer documentation below:

- **Developer Documentation**: https://internetcomputer.org/docs/current/developer-docs (see "Acquiring and using cycles" and "Mainnet Deployment")
- **Video**: https://www.youtube.com/watch?v=eynEk3Bz7QY&t=902s
- **Developer Journey**: https://internetcomputer.org/docs/current/tutorials/developer-journey
- **Developer Journey Discussion forum**: https://forum.dfinity.org/t/developer-journey-feedback-and-discussion/23893

*Please note that we receive a high volume of cycle requests at the end of every hackathon.*

*Typically you will receive cycles from the **Cycles Faucet** within 1 to 2 days, but with the increased volume of requests at the end of each hackathon there is a chance that you **may not receive the cycles in time**.*

*Be careful to **submit your cycle requests as early as possible** so that you have them in time to compete.*

## 2.6 DEVELOPER RESOURCES

HACKATHON CHEATSHEET: ICP Hackathon CheatSheet

DEVELOPER DOCUMENTATION:

- **Internet Computer Developer Documentation**
  https://internetcomputer.org/docs/current/developer-docs

DEVELOPER DISCUSSION FORUM AND DISCORD:

- **Internet Computer Developer Discussion Forum**
  https://forum.dfinity.org

- **Internet Computer Developer Discord**
  https://discord.com/invite/jnjVVQaE2C

*The official DFINITY Developer Office Hour (on Discord) is especially helpful as you can receive live (realtime) advice from professional Internet Computer developers:*

**DFINITY Developer Office Hour : Every Wednesday (2 sessions)**
Session 1: 09:00 CET
Session 2: 10:30 PDT

DEVELOPER JOURNEY:

- **Developer Journey**
  https://internetcomputer.org/docs/current/tutorials/developer-journey

- **Developer Journey Video Tutorial Series (YouTube videos)**
  https://www.youtube.com/watch?v=oBUpJ4CqmN0&list=PLuhDt1vhGcrdR2h6nPNyIXKS4u8L-efvD

- **Developer Journey Discussion Forum**
  https://forum.dfinity.org/t/developer-journey-feedback-and-discussion/23893

# 3. JUDGING PREFERENCES

## Judging Preferences

The following recommendations are not requirements, but are important "judging preferences" that are applied during the judging phase of the hackathon.

Solutions (entries) that are otherwise equivalent will be awarded higher scores when the following preferences are followed:

- **MOTOKO Preference**: Preference is given to solutions that code back-end functionality in Motoko, the open-source programming language created specifically for the Internet Computer.
- **SVELTE Preference**: Preference is given to solutions that code front-end functionality using Svelte, the open-source front-end component framework.
- **SINGLE-PAGE or MINIMAL-PAGE Preference**: Preference is given to single-page solutions where all user interactions occur on the same web page (as opposed to interfaces that use multiple web pages), or using a minimal number of web pages when single-page solutions are not possible or not ideal.
- **MOBILE-FIRST Preference**: Preference is given to "mobile-first" solutions. Mobile devices (phones, tablets, etc.) are the primary target, although your solution's user interface should still be functional using desktop/laptop computers.
- **DESIGN DOCUMENTS Preference**: Preference is given to solutions that are accompanied with corresponding design documents (architecture designs, data designs, flow diagrams, user interface mock-ups, interaction diagrams, etc.)
- **TESTING Preference**: Preference is given to solutions that implement unit, integration and end-to-end testing (if applicable). For details see the ICP Developer Journey 2.5 video and documentation.

## 4. HACKATHON TRACKS: AI & DECOM

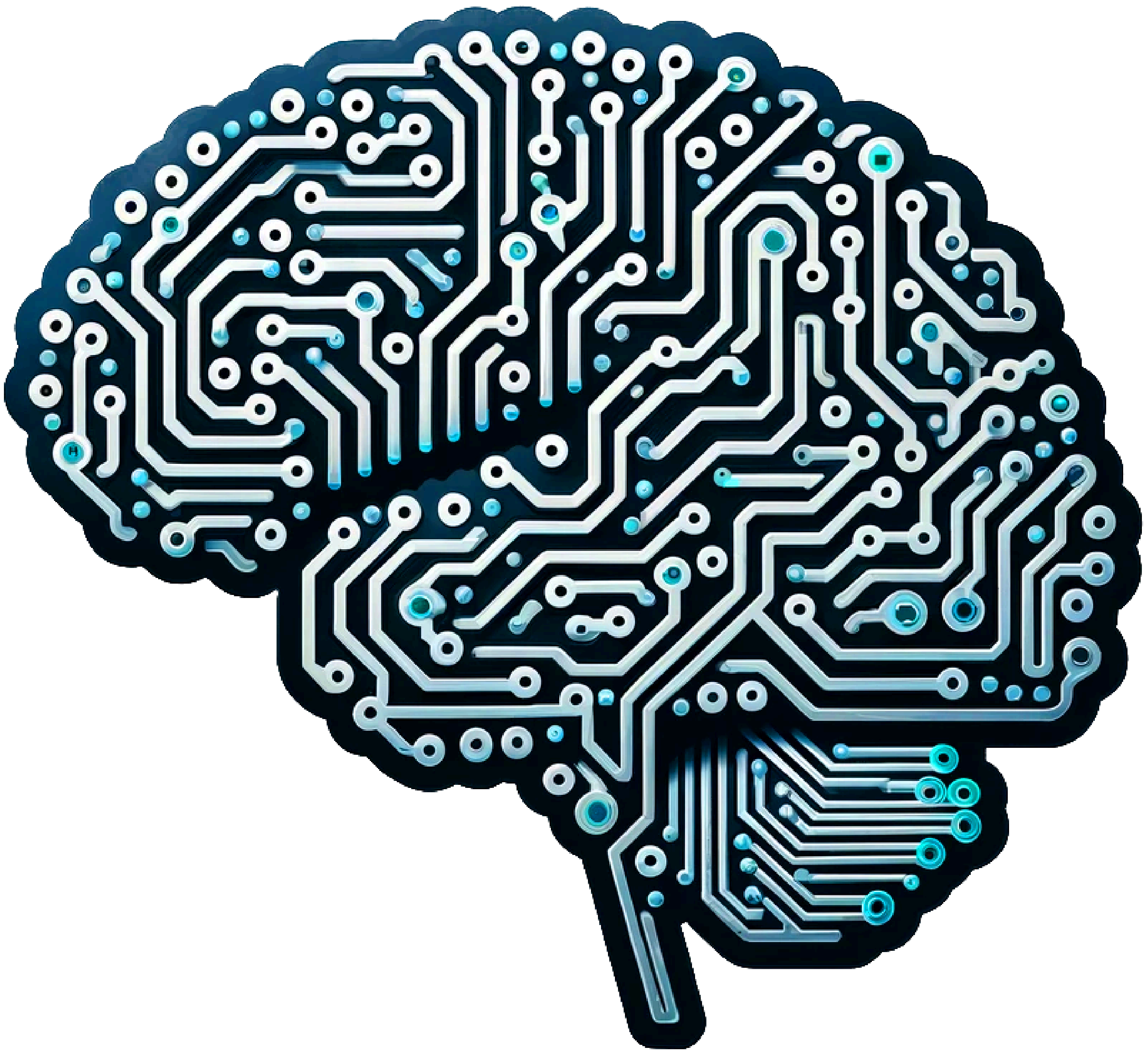
This hackathon is organized into two distinct tracks. You can compete in either one, or both, of the following tracks:

**Track 1**: AI (Artificial Intelligence)

**Track 2**: DeCom (Decentralized Commerce)

*Note that your entire solution, including how you represent, store and retrieve data, must run on the Internet Computer. It is not permissible to use services that run outside of the Internet Computer.*

# 5. TRACK 1: Artificial Intelligence



*Please note that the following instructions are the full extent of detail provided to hackathon competitors. The instructions intentionally do not provide technical specifications, nor technical guidelines. A primary goal of the hackathon is to enable competitors to design their own solutions based on a minimal set of general guidelines.*

## AI Personal Tutor (Learning Guide)

The target of this track is an AI personal tutor, or learning guide, that helps learners comprehend complex and/or dense subject matter and materials.

### Learning Materials

Your AI solution should teach learners the following subject matter using the information sources provided below.

Learners should be given the ability to decide for themselves what subject they would like to learn about at any given time (each heading below should be considered a unique subject).

The learning materials for each subject should be stored on the Internet Computer and should not be used directly from the web.

*Don't use these materials directly from the links below; bring this information into your Internet Computer solution so that your entire solution, including all learning materials, is on the Internet Computer.*

***UNITED NATIONS CONVENTION AGAINST CORRUPTION***

https://unodc.org/documents/treaties/UNCAC/Publications/Convention/08-50026_E.pdf

***POLICY GUIDE FOR NATIONAL ANTI-CORRUPTION AUTHORITIES ON MEANINGFUL YOUTH ENGAGEMENT IN ANTI-CORRUPTION WORK***

https://grace.unodc.org/grace/en/youth-empowerment/policy-guide-for-national-anti-corruption-authorities-on-meaningful-youth-engagement-in-anti-corruption-work.html

***YouthLED TALKS***

https://grace.unodc.org/grace/en/youth-empowerment/youthled-talks.html

***A TEACHER'S GUIDE TO USING FORUM THEATRE TO PROMOTE THE RULE OF LAW***
***https://grace.unodc.org/grace/en/secondary/acting-for-the-rule-of-law.html***

***UNIVERSITY MODULE SERIES ON ANTI-CORRUPTION***

https://grace.unodc.org/grace/en/academia/module-series-on-anti-corruption.html

## Tracking Learner Progress via Full Name

Your solution should track the learner's progress, and also allow them to leave or stop (even quit their browser entirely) and then return at any time to resume. When a learner returns they should resume where they left off.

Anyone with a web browser should be able to use your solution: an Internet Computer account is not required.

To facilitate public access your solution is **not** required to support real user accounts, authentication or sign-in. You should instead **use the learner's full name** as a unique identifier.

Allow anyone to use your solution by providing their **full name**, such as:

Alice Smith
Raja Gupta
Naveed Khan

For the sake of simplicity you can assume that only one person is associated with a unique full name, meaning you can track learner progress based merely on their full name.

Note that in reality this approach is not adequate, as many people have the same full name, but for the purposes of this hackathon you can assume that "Alice Smith" is always the same person, for example.

*Use the learner's full name to track their individual progress, allowing them to quit (stop/leave) and come back (resume where they left off), merely by providing their full name each time they use your solution.*

## Assessment

Your solution should provide learners with the ability to assess their knowledge of any subject at any time. In this way learners can see, at any given point in time, how much (or little) they have learned about any subject that they have studied using your solution.

You are encouraged, but not required, to store the results of each assessment conducted so that learners can review the progress they have made at any time. For example, a learner might leave and return several days later and, upon return, decide to check on their overall progress for a given subject (or all subjects) as of that time.

# 6. TRACK 2: Decentralized Commerce

*Please note that the following instructions are the full extent of detail provided to hackathon competitors. The instructions intentionally do not provide technical specifications, nor technical guidelines. A primary goal of the hackathon is to enable competitors to design their own solutions based on a minimal set of general guidelines.*

## Global Multi-Currency Online Store

The target of this track is a decentralized online store where users located anywhere in the world can conduct commerce (buy and sell goods) using any number of traditional (fiat) currencies and digital (crypto) currencies.

### Access via Full Name

Your solution should allow anyone with a web browser to enter the store, where they can buy and sell as they wish, without requiring them to have an Internet Computer account.

To facilitate public access your solution is **not** required to support real user accounts, authentication or sign-in. You should instead **utilize the user's full name** as a unique identifier.

Allow anyone to use your store to buy and sell based only on their **full name**, such as:

Alice Smith
Raja Gupta
Naveed Khan

For the sake of simplicity you can assume that only one person is associated with a unique full name, meaning you can manage their entire store account using just their full name.

Note that in reality this approach is not adequate, as many people have the same full name, but for the purposes of this hackathon you can assume that "Raja Gupta" is always the same person, for example.

*Use full names to provide access to the store, allowing users to quit (stop/leave) and come back (resume where they left off), by providing only their full name each time they use your solution.*

## Seamless Buyer/Seller User Interface

Every user of the store is both a buyer and a seller: they can buy goods (products) that have been placed into the store by other users, and sell goods of their own.

As such, the solution that you develop should enable users to easily and seamlessly switch between *buyer* and *seller* modes at any time.

## Buyer Mode

When in Buyer mode the user is able to shop just as you would expect with a traditional online store such as Amazon, Rakuten, AliExpress, Shein, Temu, etc.

Basic features that your solution should support when in Buyer mode include:

- **Search** - the ability to search for products by name, keyword (tag) or category and see Previews of all matching items.
- **Sort** - the ability to sort products by name, category, price, date listed on the store, etc.
- **Preview** - thumbnail preview showing an image of the product, name of the product, short description of the product, price, and an Add to Cart button.
- **Details** - the ability to select a Preview to see more details about that product (full description, date listed on the store, seller's name, product rating, price, etc.)
- **Add to Cart** - button that the user can select to place a product into their personal shopping cart. This button should be available in both the Preview and Details views.
- **Check Out** - button that allows the user to see all of the products in their cart, the individual price for each, and the total price. Do not concern yourself with taxes or VAT (ignore taxes and VAT).
- **Place Order** - button that allows the user to execute the order (buy all of the products in their cart).

## Seller Mode

When in Seller mode the user is able to place products into the store for others to buy.

Basic Seller mode features that your solution should allow the user to add for a newly listed product, and ideally change or edit at any later time, include:

- **Product image** - small image of the product being offered for sale. All images should be stored on the Internet Computer (if you are not able to do so you can store images on your own server, although such an approach will not be judged as highly).

- **Product name** - short text name of the product.

- **Short description** - short text description of the product that is displayed in the Preview.

- **Long description** - full text description of the product that is displayed in the Details view (the Details view is shown when the user clicks/touches a Preview, as described in Buyer Mode above).

- **Price** - price of the product specified in the seller's currency of choice

## Currency of Choice (BASE Currency) on a Per-Product Basis

The store is multi-currency from both the buy and sell sides: Sellers choose the currency they want to receive for each product they sell, buyers choose the currency they want to pay with when buying a product.

Following, at a minimum, are the six currencies your solution should support:

- US Dollars
- Euro
- British Pound

- Bitcoin
- Ethereum
- ICP

Sellers specify the price for the products they sell in traditional (fiat) currency as well as digital (crypto) currency. The currency they choose to sell a product for is the BASE currency for that particular product; during the sale of that item all other currencies (fiat and crypto) are calculated relative to that BASE currency.

Every time a user decides to sell a product they choose the price amount and the currency for that particular product. They might, for example, offer a product for sale at the price of 23 Euro and another product for sale at the price of 0.000171 Bitcoin (BTC), and yet another product for sale at the price of 12 British Pounds.

For example, if a user offers to sell a product for $5.00 (five dollars) the BASE currency for that product is US Dollars. A buyer of that product would therefore pay either five US Dollars for the product, or the equivalent amount of any other currency based on the exchange rate at the time of purchase.

In this example the seller will still receive five US Dollars even when the buyer pays in a different currency. The buyer might purchase the product using the correct corresponding amount of British Pound, for example, which the store will convert to exactly five US Dollars at the time of sale.

Likewise, if a user offers to sell a product for 0.000043 Bitcoin (BTC) the BASE currency for that particular product is Bitcoin. A buyer would therefore pay either 0.000043 Bitcoin for the product, or the equivalent amount of any other currency based on the exchange rate at the time of purchase. In any case the seller will receive 0.000043 Bitcoin even if the buyer pays in a different currency (the buyer might purchase the product using the correct corresponding amount of Euro, for example, which the store will convert to 0.000043 Bitcoin at the time of sale).

*Note that your solution will not conduct any currency or cryptocurrency transactions. Do not create wallets or transact any currency or cryptocurrency. Simply calculate the prices and keep an accurate account of what the prices should be, for purposes of illustration, as explained below.*

## Calculating Currencies: Every 10 Minutes & At Time of Sale

Internet Computer's HTTPS Outcalls feature allows you to access the traditional web (aka "web 2.0") directly from your Internet Computer program, making it one of the ways in which you might consider calculating currency exchange rates to facilitate the purchase of products using currencies other than a given product's BASE currency.

Keep in mind, however, that HTTPS Outcalls are not "free" as they consume cycles and, ultimately, cycles cost money. As such, HTTPS Outcalls should be used sparingly.

For the purposes of this hackathon you do not need to constantly calculate currency exchange rates. Instead, calculate exchange rates once every 10 minutes and store the results in global variables or global data structures that can be accessed by all parts of your program that use currencies.

The **exception** comes at the time of sale (the actual sales transaction), since it's very important to show the buyer the actual amount they will be paying in their currency of choice if they're not paying with a product's BASE currency. You should therefore: **1)** calculate exchange rates immediately before the user purchases a product(s), **2)** alert them to any changes in price(s), and **3)** receive their approval/confirmation for changed price(s) before executing the transaction.

### Pause 10 Minute Currency Exchange Rate Calculations When Not In Use

Note that it would be extremely wasteful to retrieve exchange rates and calculate currencies every 10 minutes when no users are using your solution.

Pause exchange rate retrieval and currency calculations when nobody is using your store, or when all signed-in users have been idle for a period of time.

## Time of Sale and Record Keeping

Your solution should keep a record of every product transaction (buy/sell) and provide key aspects of those details, as appropriate, to both buyer and seller.

At the time of sale (when a product is purchased), record all relevant details so that the buyer can review their purchase records in detail at any time. Likewise, allow the seller to see relevant details for the products they have sold.

This is similar to how traditional online stores (Amazon, etc.) provide records for both buyers and sellers.

Note, however, that your solution is **not** actually transferring currency or working directly with any fiat currency or cryptocurrency other than to merely calculate exchange rates for the purpose of displaying accurate prices as described above.

In other words, your solution is ***illustrating the concept*** but not actually transferring, converting or exchanging currencies nor is your solution executing real buy/sell transactions.

*Your solution will not conduct any fiat currency or cryptocurrency transactions. Do not create wallets or transact any currency or cryptocurrency. Only calculate prices, for purposes of illustration, as described above.*

# 7. SUBMITTING YOUR SOLUTION

**SUBMIT YOUR SOLUTION ON TIME!**

For details on **how and when** to submit your hackathon solution please refer to the **SUBMIT YOUR SOLUTION** section of the official hackathon page at:

https://summit.ImmersiveEducation.org/Oxford/2024/hackathon.html