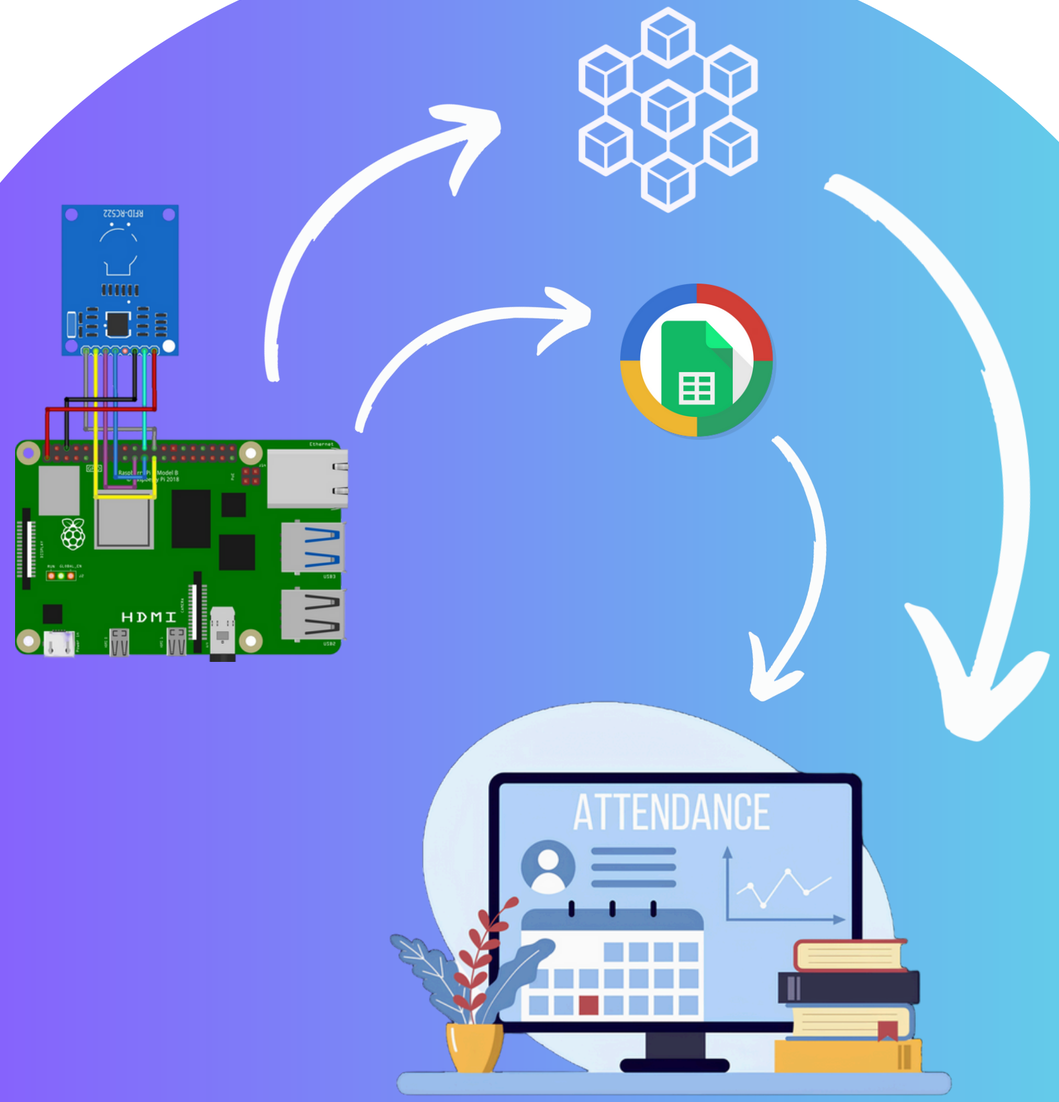
CARLO SORRENTINO
MAT. 0522501821

Smart-Tag Attendance System

Sistema automatizzato per la registrazione delle presenze degli studenti a lezioni e durante gli esami, basato su tecnologia RFID e Blockchain.



PROGETTO COMBINATO PER I CORSI DI
SICUREZZA DEI DATI & IOT SECURITY

ANNO ACCADEMICO 2023/2024.

Iniziamo con uno scenario familiare...

Iniziamo con uno scenario familiare...

Iniziamo con uno scenario familiare...

Iniziamo con uno scenario familiare...

Iniziamo con uno scenario familiare...

Iniziamo con uno scenario familiare...

SVANTAGGI DEL SISTEMA PRESENTATO

- Problema di Archiviazione
- Difficoltà di Condivisione
- Fatica di Raccolta e Recupero Dati
- Dispendio in Termini di Tempo
- Inefficienza
- Limitata Tracciabilità
- Mancanza di Backup
- Prone ad Errore Umano
- Rischio di Frode
- Sicurezza Limitata
- ...

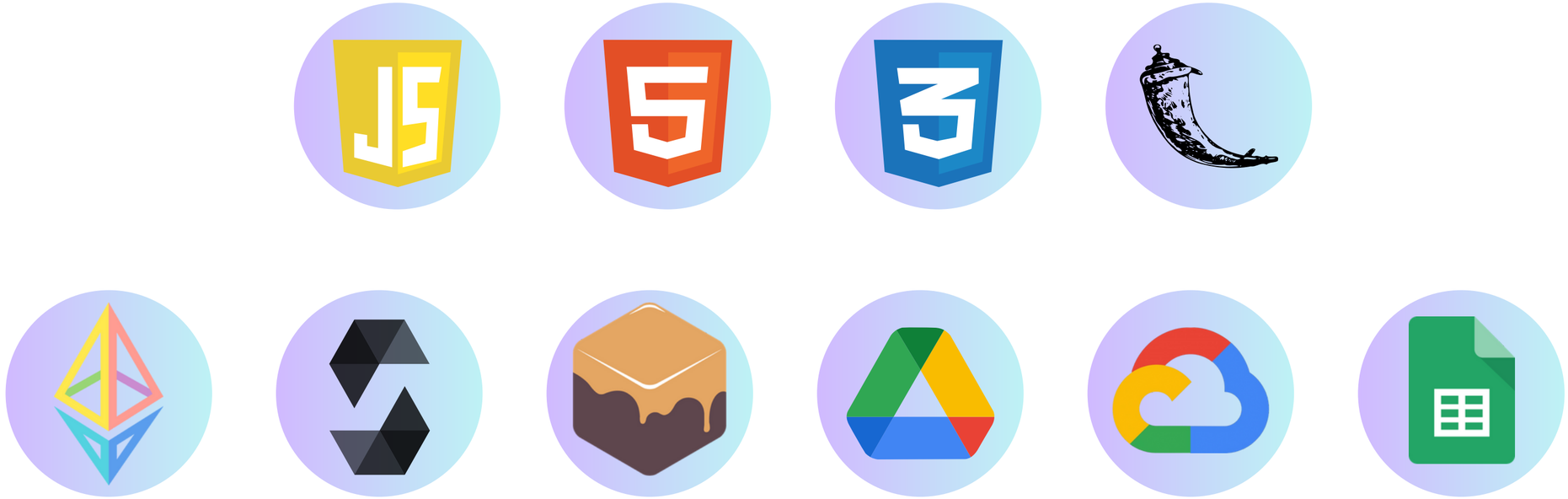
Obiettivo...

Superare l’Obsolescenza.

MIGLIORARE L'ACCURATEZZA E L’EFFICIENZA DELLA REGISTRAZIONE DELLE PRESENZE,
SUPERANDO IL METODO CARTACEO ANTICO, GARANTENDO ALLO STESSO TEMPO
LA SICUREZZA DEI DATI E FACILITANDO L'ACCESSO E LA GESTIONE DELLE INFORMAZIONI.

Componenti e Tecnologie Utilizzate

Componenti e Tecnologie Utilizzate



RASPBERRY-PI RFID PYTHON SQLite

Base

Componenti e Tecnologie Utilizzate

ETHEREUM

SOLIDITY

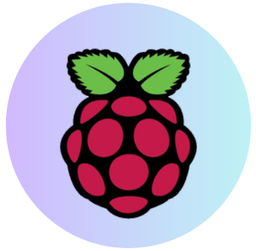
GANACHE

G. DRIVE

G. CLOUD

G. SHEET

Gestione Dati



RASPBERRY-PI

RFID

PYTHON

SQLite

Componenti e Tecnologie Utilizzate


JAVASCRIPT


HTML


CSS


FLASK

Interfaccia di Monitoraggio


ETHEREUM

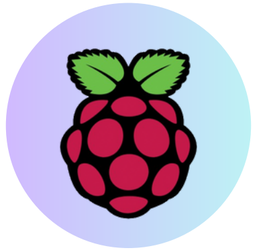

SOLIDITY


GANACHE


G. DRIVE


G. CLOUD


G. SHEET

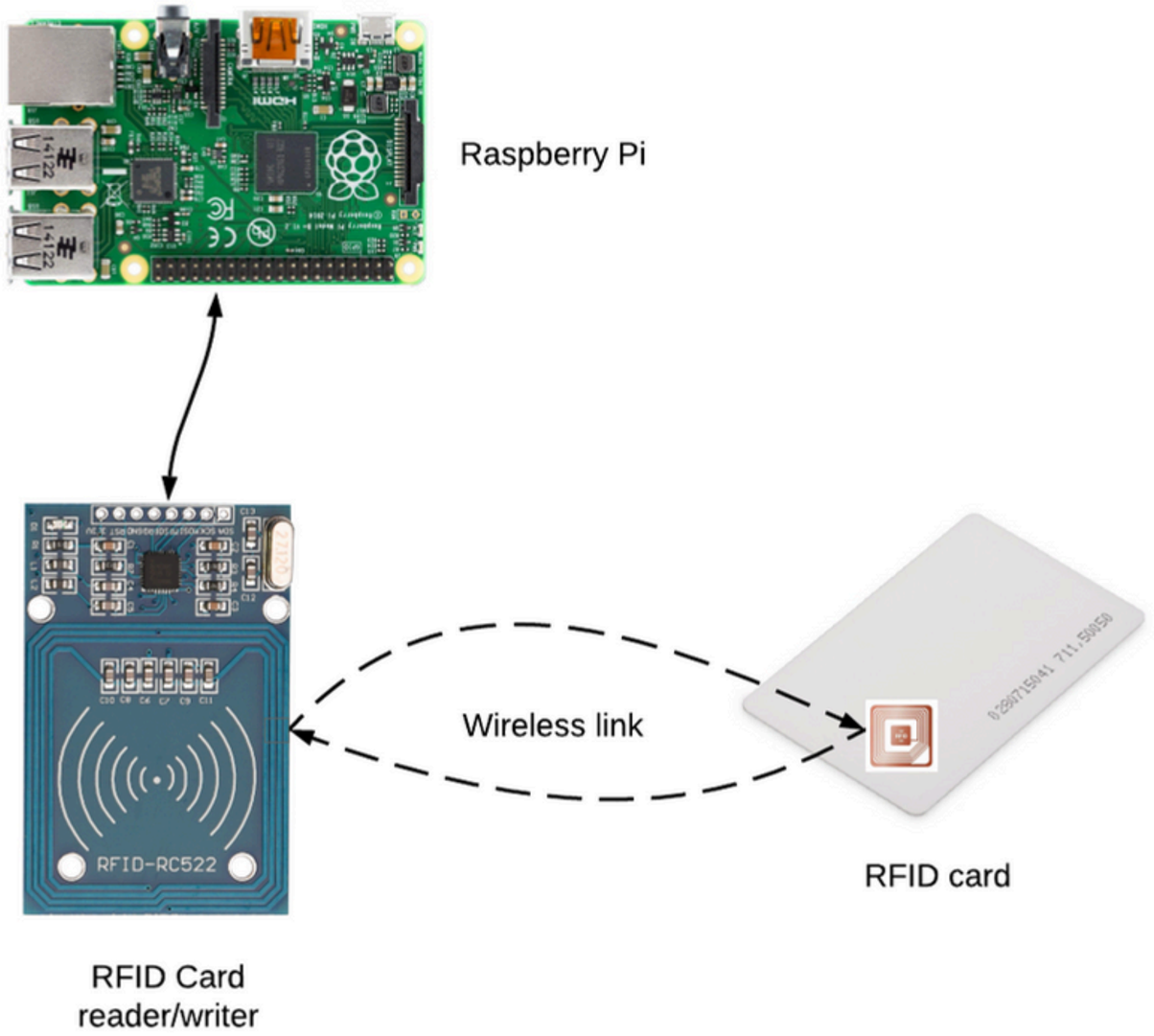

RASPBERRY-PI


RFID


PYTHON


SQLite

Il Cuore del Sistema: Raspberry Pi



Implementazione di base.

Il Cuore del Sistema: Raspberry Pi

Il Cuore del Sistema: Raspberry Pi

Il Cuore del Sistema: Raspberry Pi

1. ISCRIZIONE dello studente al CORSO.

Il Cuore del Sistema: Raspberry Pi

1. ISCRIZIONE dello studente al CORSO.
2. REGISTRAZIONE della presenza a LEZIONE.

Il Cuore del Sistema: Raspberry Pi

1. ISCRIZIONE dello studente al CORSO.
2. REGISTRAZIONE della presenza a LEZIONE.
3. REGISTRAZIONE della presenza all’ESAME.

Il Cuore del Sistema: Raspberry Pi

1. ISCRIZIONE dello studente al CORSO.
2. REGISTRAZIONE della presenza a LEZIONE.
3. REGISTRAZIONE della presenza all’ESAME.

Il Cuore del Sistema: Raspberry Pi

1. ISCRIZIONE dello studente al CORSO.
2. REGISTRAZIONE della presenza a LEZIONE.
3. REGISTRAZIONE della presenza all’ESAME.

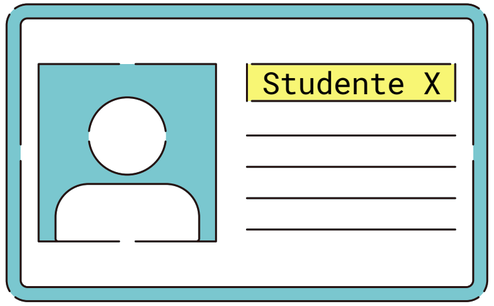
Cifratura dei Dati

Cifratura dei Dati

Cifratura dei Dati

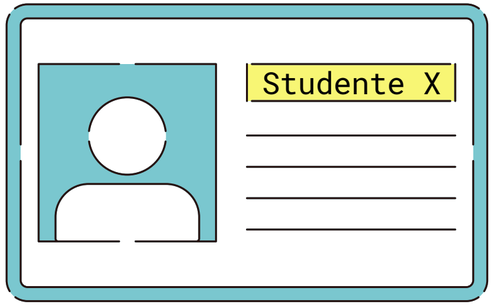
Codice **UNIVOCO** che permette di risalire al Profilo Digitale dello Studente salvato nel Database.

Cifratura dei Dati

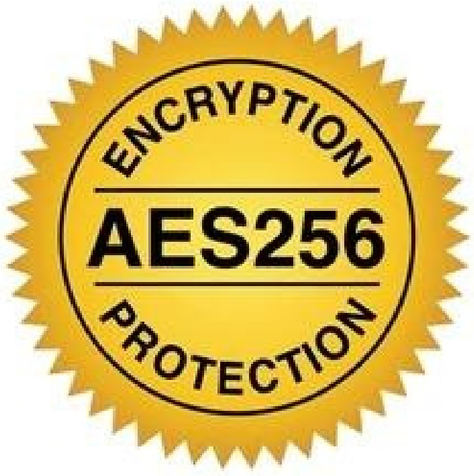


Codice **UNIVOCO** che permette di risalire al Profilo Digitale dello Studente salvato nel Database.

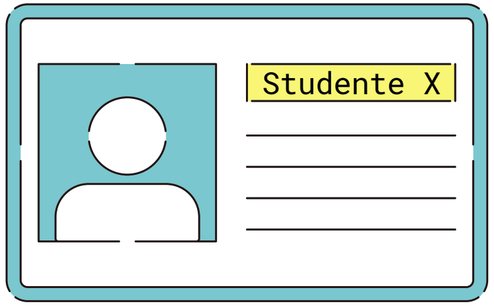
Cifratura dei Dati



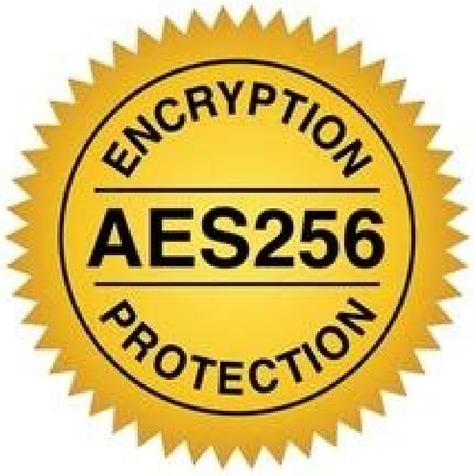
Codice **UNIVOCO** che permette di risalire al Profilo Digitale dello Studente salvato nel Database.



Cifratura dei Dati

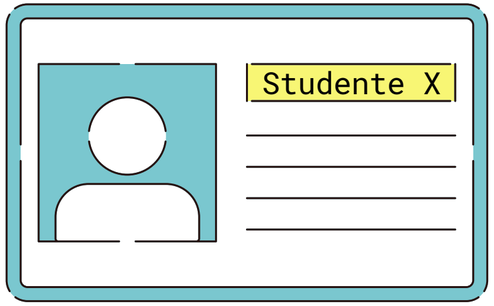


Codice **UNIVOCO** che permette di risalire al Profilo Digitale dello Studente salvato nel Database.



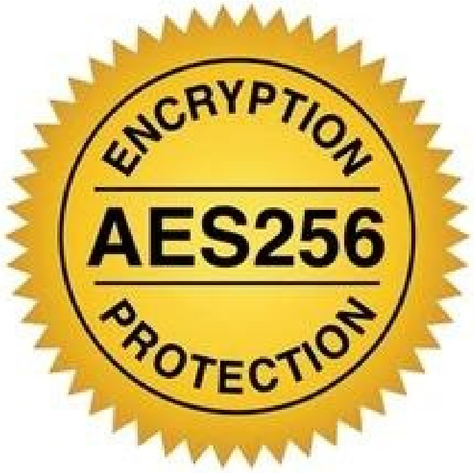
e IV

Cifratura dei Dati



Codice **UNIVOCO** che permette di risalire al Profilo Digitale dello Studente salvato nel Database.

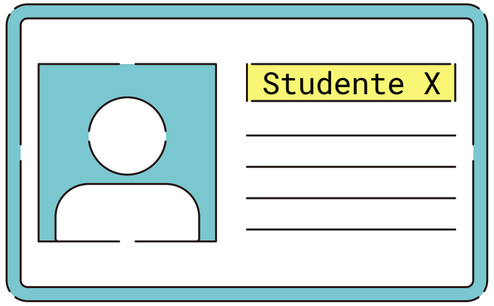
\+



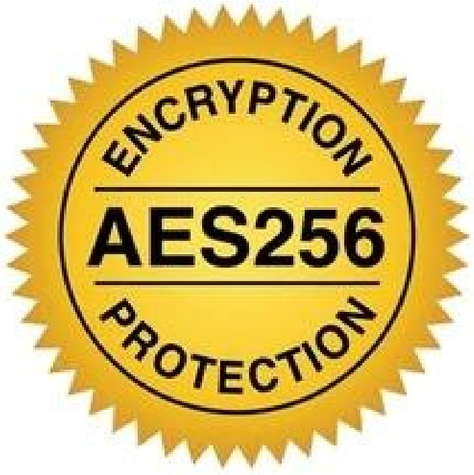
\=

e IV

Cifratura dei Dati

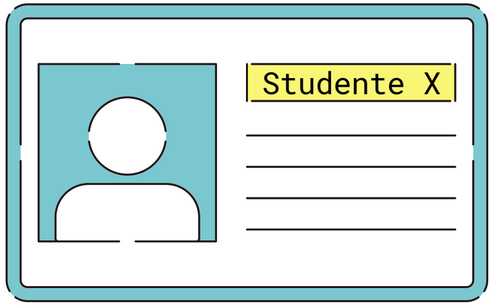


Codice **UNIVOCO** che permette di risalire al Profilo Digitale dello Studente salvato nel Database.

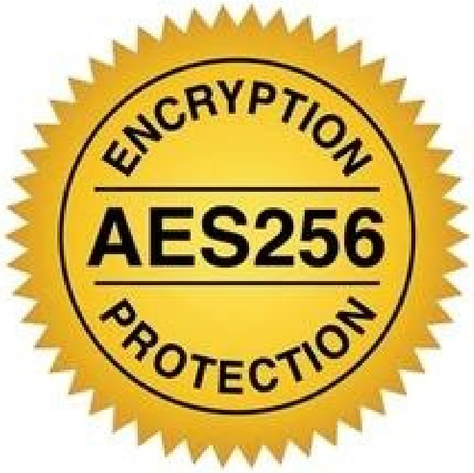


e IV

Cifratura dei Dati



Codice **UNIVOCO** che permette di risalire al Profilo Digitale dello Studente salvato nel Database.



e IV

Gestione  e IV

Gestione  e IV

$$\forall \text{ Operazione} \in \{\text{Iscrizione, Lezione, Esame}\}$$
$$\exists! \ (k,iv) \in \text{Wallet} \mid \text{Operazione} \leftrightarrow (k,iv)$$

Gestione 🔑 e IV

∀ Operazione ∈ {Iscrizione, Lezione, Esame}
∃! (k,iv) ∈ Wallet | Operazione ↔ (k,iv)

Gestione  e IV

$$\forall \text{ Operazione} \in \{\text{Iscrizione, Lezione, Esame}\}$$
$$\exists! \ (k,iv) \in \text{Wallet} \mid \text{Operazione} \leftrightarrow (k,iv)$$

ChaviCorso

File Modifica Visualizza Inserisci Formato Dati Strumenti Estensioni Guida

100% € % .0 .00 123 Predefi.. 10

J17

	A	B	C	D	E	F
1	Operazione	Chiave	IV			
2	Registrazione	b'\xbd\x0b\x9bB\	b"\xb6\xb9\xb2'\xe6\xad\xa47\x1c\x8f\xb4\xb32=a\x0c"			
3	Lezione_1_Intro	b'\x9e\xd6\x9e\x	b'\xa1\xe8l\xc5\xfd\x9e/^\xd3\xc8\xcfc\x13\\\xb6\x84'			
4	06/09/2025	b'\xc0g]\x16\xd8	b'CRF}\xbbH\x06\xb2\xb4E\xfe\xd2\xba4\xb8\x9e'			
5						

Gestione  e IV

$$\forall \text{ Operazione} \in \{\text{Iscrizione, Lezione, Esame}\}$$
$$\exists! \ (k,iv) \in \text{Wallet} \mid \text{Operazione} \leftrightarrow (k,iv)$$

Gestione 🔑 e IV

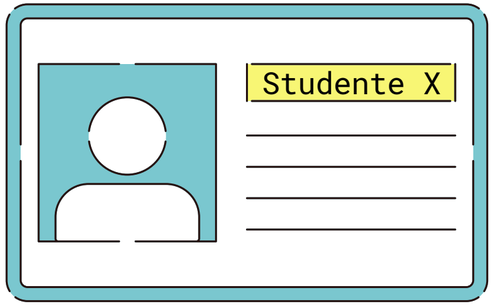
∀ Operazione ∈ {Iscrizione, Lezione, Esame}
∃! (k,iv) ∈ Wallet | Operazione ↔ (k,iv)

Codice UNIVOCO di
Studente X **CIFRATO con k**
e iv di Lezione_1

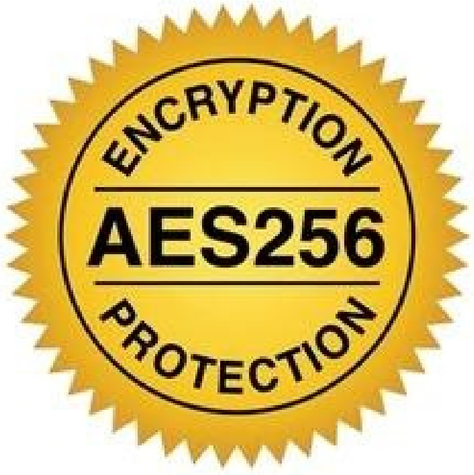
≠

Codice UNIVOCO di
Studente X **CIFRATO con k**
e iv di Lezione_2

Cifratura dei Dati

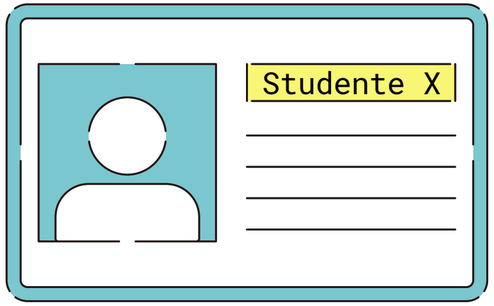


Codice **UNIVOCO** che permette di risalire al Profilo Digitale dello Studente salvato nel Database.



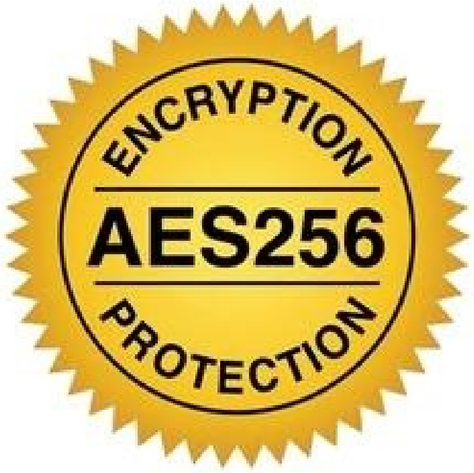
e IV

Cifratura dei Dati



Codice **UNIVOCO** che permette di risalire al Profilo Digitale dello Studente salvato nel Database.

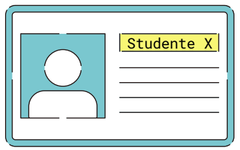
\+



\=

e IV

Cifratura dei Dati



Codice **UNIVOCO** che permette di risalire al Profilo Digitale dello Studente salvato nel Database.

e IV

Codice UNIVOCO CIFRATO

- **Iscrizione**
- **Lezione**
- **Esame**

Tipo Operazione
con
Nome Lezione / Data Esame

Nome Corso

Dati Ottenuti

Codice UNIVOCO CIFRATO

+

- **Iscrizione**
- **Lezione**
- **Esame**

Tipo Operazione
con
Nome Lezione / Data Esame

+

Nome Corso

Dati Ottenuti

Codice UNIVOCO CIFRATO

+

- **Iscrizione**
- **Lezione**
- **Esame**

Tipo Operazione
con
Nome Lezione / Data Esame

+

Nome Corso

Dati per la Transazione

BlockChain

BlockChain

SMARTCONTRACT

```solidity
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract AttendanceTracker {
    struct Record {
        string operationType; // "Registrazione", "Lezione", "Esame"
        string courseName; // Nome del corso
        string additionalInfo; // Può essere "nomeLezione" o "dataEsame", se l'operazione è "Registrazione" il campo è vuoto
        string encryptedId; // ID dello studente cifrato
    }

    // Array di tutti i record
    Record[] private records;

    // Evento emesso ogni volta che un record viene creato
    event RecordCreated(string operationType, string courseName, string additionalInfo, string encryptedId);

    // Funzione per aggiungere un record
    function addRecord(string memory operationType, string memory courseName, string memory additionalInfo, string memory encryptedId) public {
        records.push(Record(operationType, courseName, additionalInfo, encryptedId));
        emit RecordCreated(operationType, courseName, additionalInfo, encryptedId);
    }

    // Funzione per contare le registrazioni di un dato corso
    function countRegistrations(string memory courseName) public view returns (uint) {
        uint count = 0;
        for(uint i = 0; i < records.length; i++) {
            if(keccak256(bytes(records[i].operationType)) == keccak256(bytes("Registrazione")) && keccak256(bytes(records[i].courseName)) == keccak256(bytes(courseName))) {
                count++;
            }
        }
        return count;
    }

    // Funzione per contare le presenze per una data lezione di un corso specifico
    function countLessonAttendances(string memory courseName, string memory lessonName) public view returns (uint) {
        uint count = 0;
        for(uint i = 0; i < records.length; i++) {
            if(keccak256(bytes(records[i].operationType)) == keccak256(bytes("Lezione")) &&
            keccak256(bytes(records[i].additionalInfo)) == keccak256(bytes(lessonName)) &&
            keccak256(bytes(records[i].courseName)) == keccak256(bytes(courseName))) {
                count++;
            }
        }
        return count;
    }
```

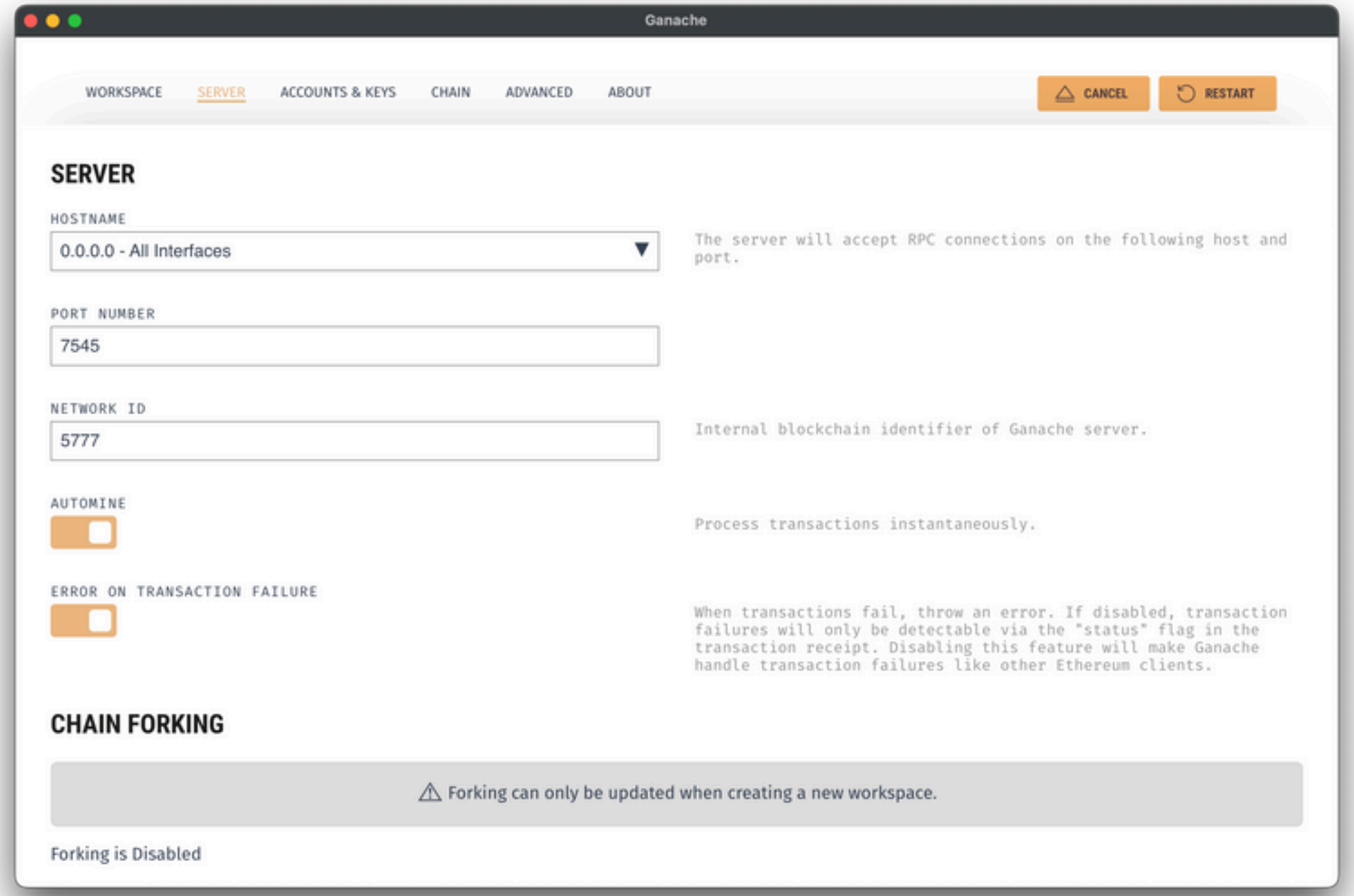
BlockChain

SMARTCONTRACT

```
47
48      // Funzione per contare quanti studenti hanno partecipato ad un dato esame di un determinato corso
49      function countExamParticipations(string memory courseName, string memory examDate) public view returns (uint) {
50          uint count = 0;
51          for(uint i = 0; i < records.length; i++) {
52              if(keccak256(bytes(records[i].operationType)) == keccak256(bytes("Esame")) &&
53              keccak256(bytes(records[i].additionalInfo)) == keccak256(bytes(examDate)) &&
54              keccak256(bytes(records[i].courseName)) == keccak256(bytes(courseName))) {
55                  count++;
56              }
57          }
58          return count;
59      }
60
61      // Funzione per ottenere i record in base al tipo di operazione e dettagli specificati
62      function getRecordsByOperation(string memory operationType, string memory courseName, string memory additionalInfo) public view returns (Record[] memory) {
63          Record[] memory tempRecords = new Record[](records.length);
64          uint count = 0;
65
66          for(uint i = 0; i < records.length; i++) {
67              bool matchOperationType = keccak256(bytes(records[i].operationType)) == keccak256(bytes(operationType));
68              bool matchCourseName = keccak256(bytes(records[i].courseName)) == keccak256(bytes(courseName));
69              bool matchAdditionalInfo = keccak256(bytes(records[i].additionalInfo)) == keccak256(bytes(additionalInfo)) || keccak256(bytes(additionalInfo)) == keccak256("");
70
71              if(matchOperationType && matchCourseName && (matchAdditionalInfo || keccak256(bytes(operationType)) == keccak256(bytes("Registrazione")))) {
72                  tempRecords[count] = records[i];
73                  count++;
74              }
75          }
76
77          Record[] memory filteredRecords = new Record[](count);
78          for(uint j = 0; j < count; j++) {
79              filteredRecords[j] = tempRecords[j];
80          }
81
82          return filteredRecords;
83      }
84  }
```

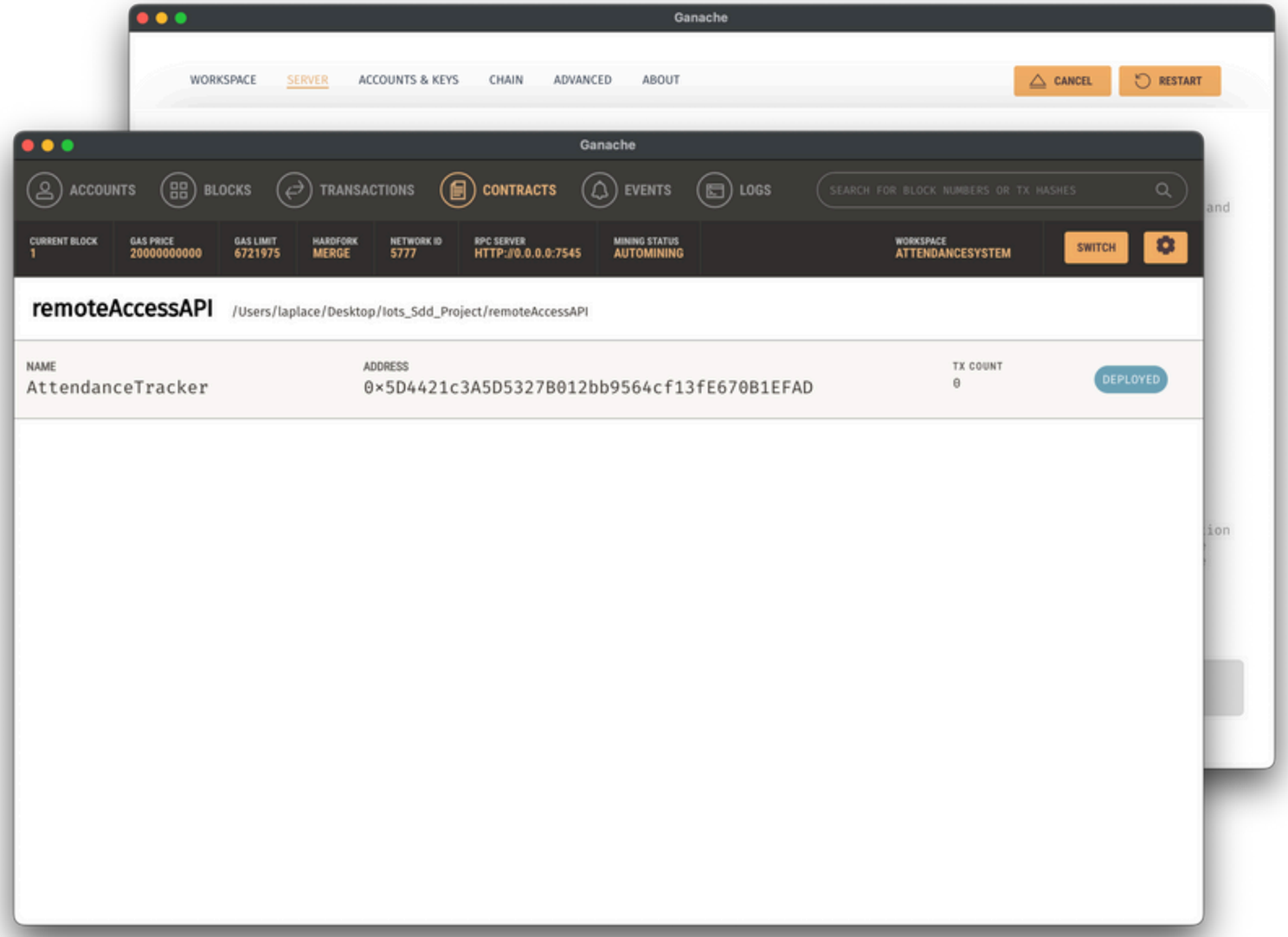
BlockChain

GANACHE



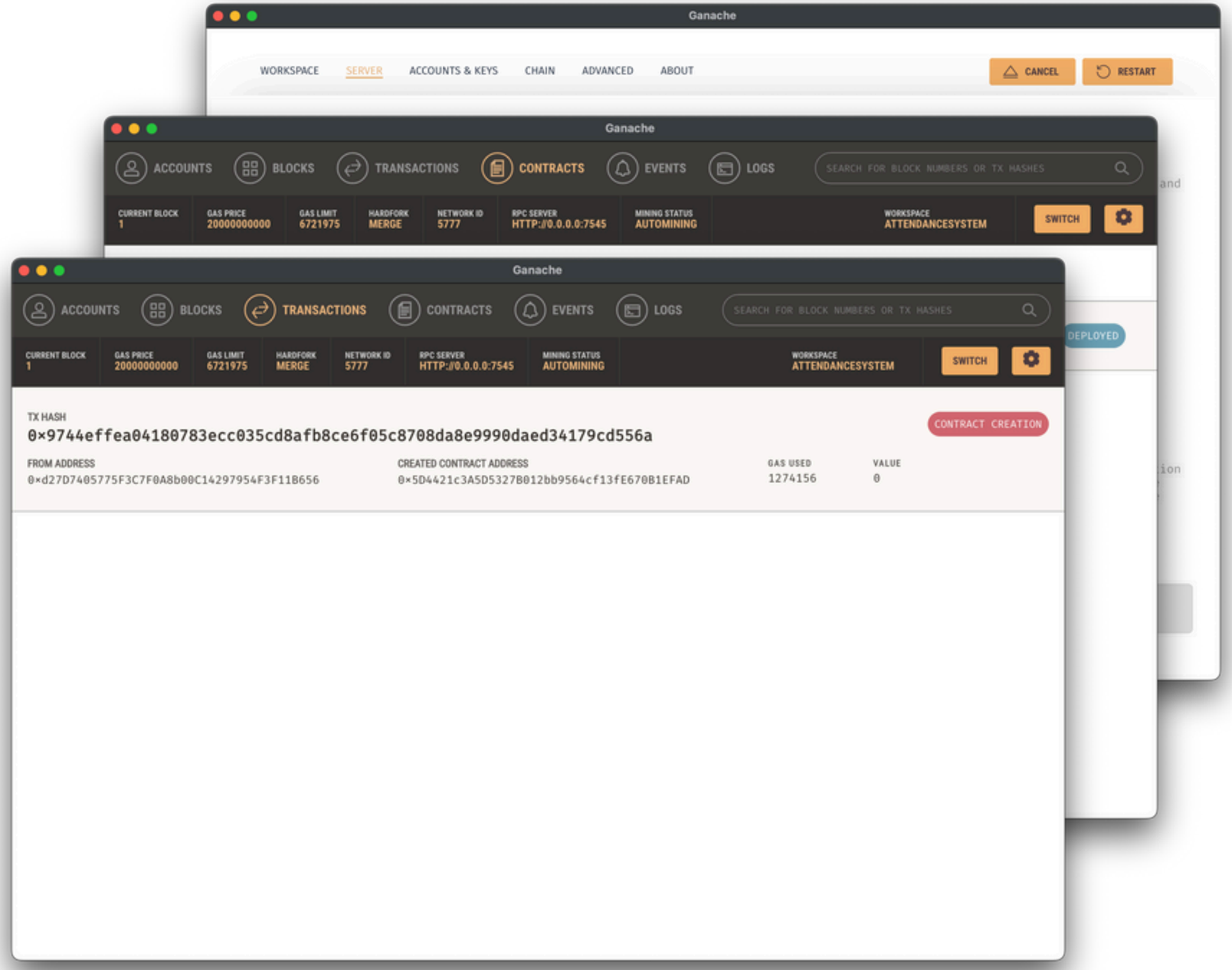
BlockChain

GANACHE



BlockChain

GANACHE

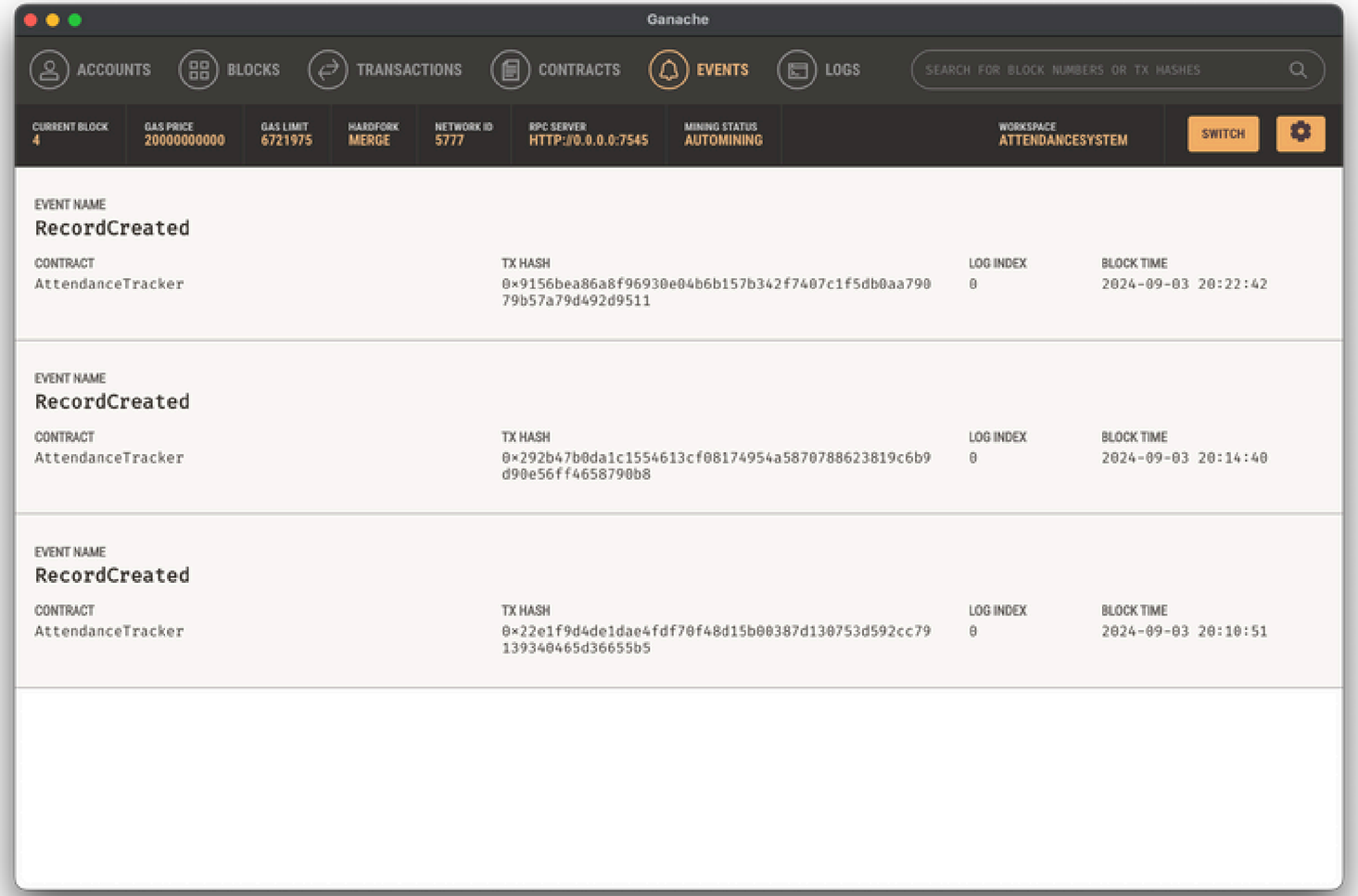


BlockChain

GANACHE

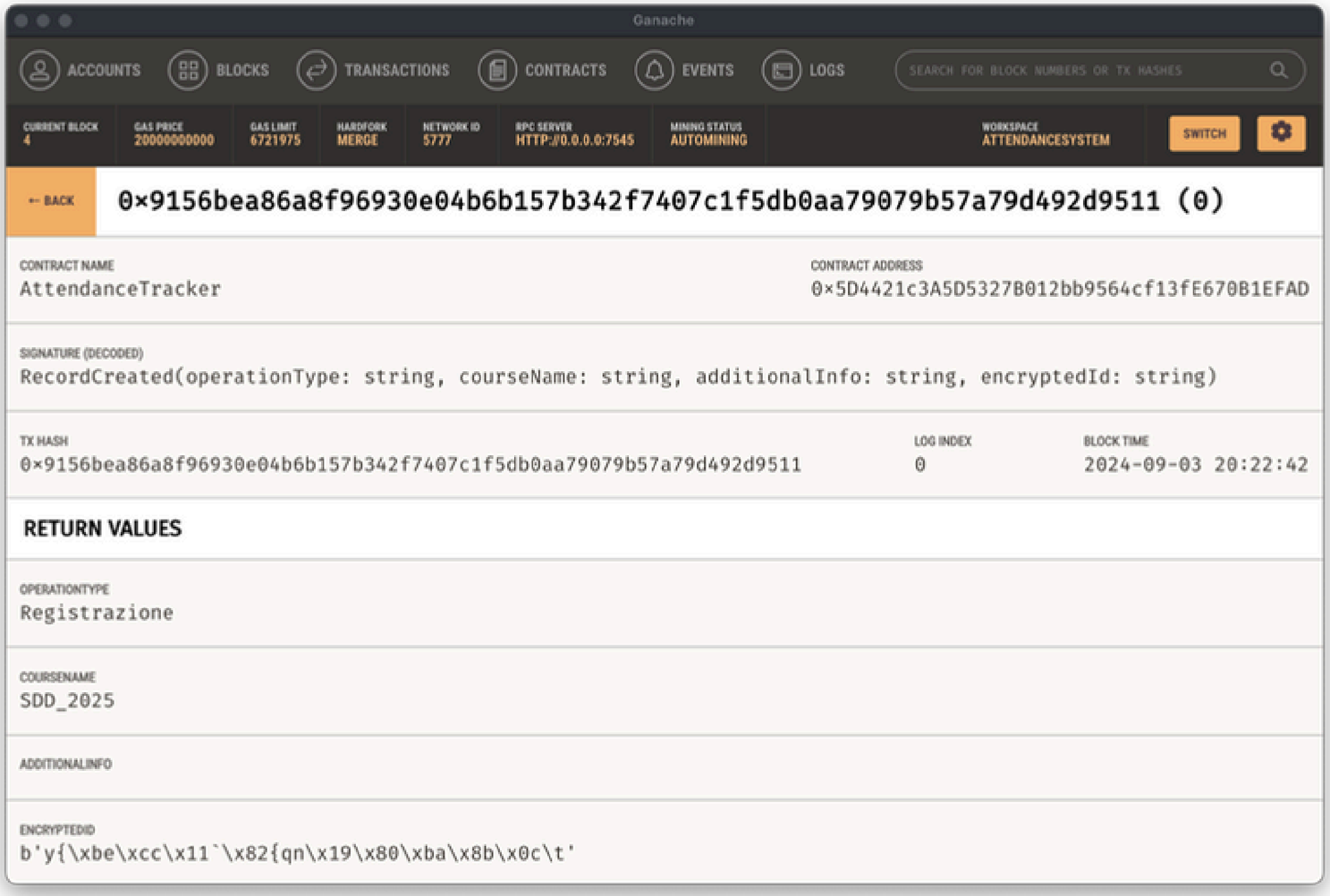
BlockChain

GANACHE



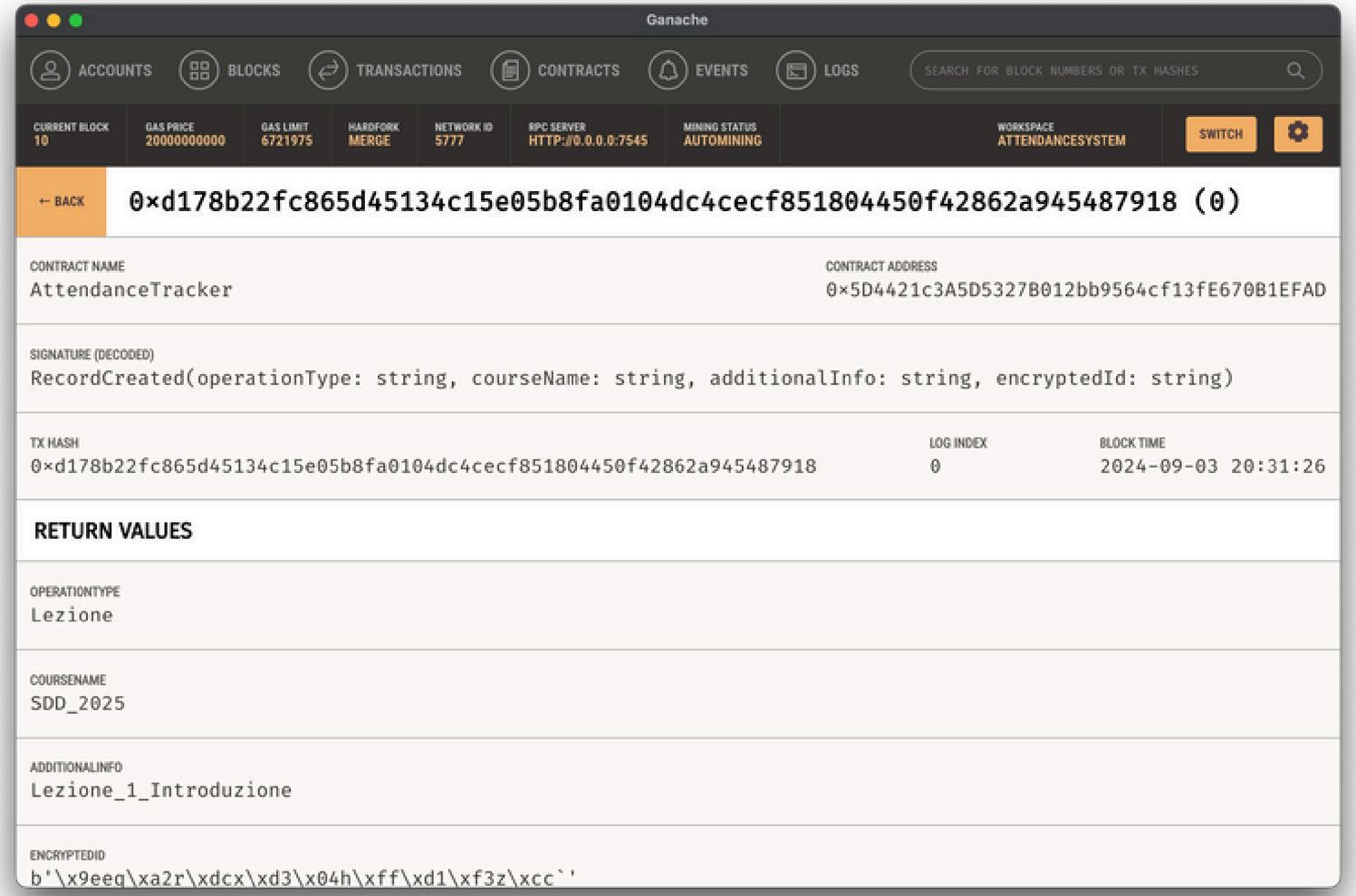
BlockChain

GANACHE



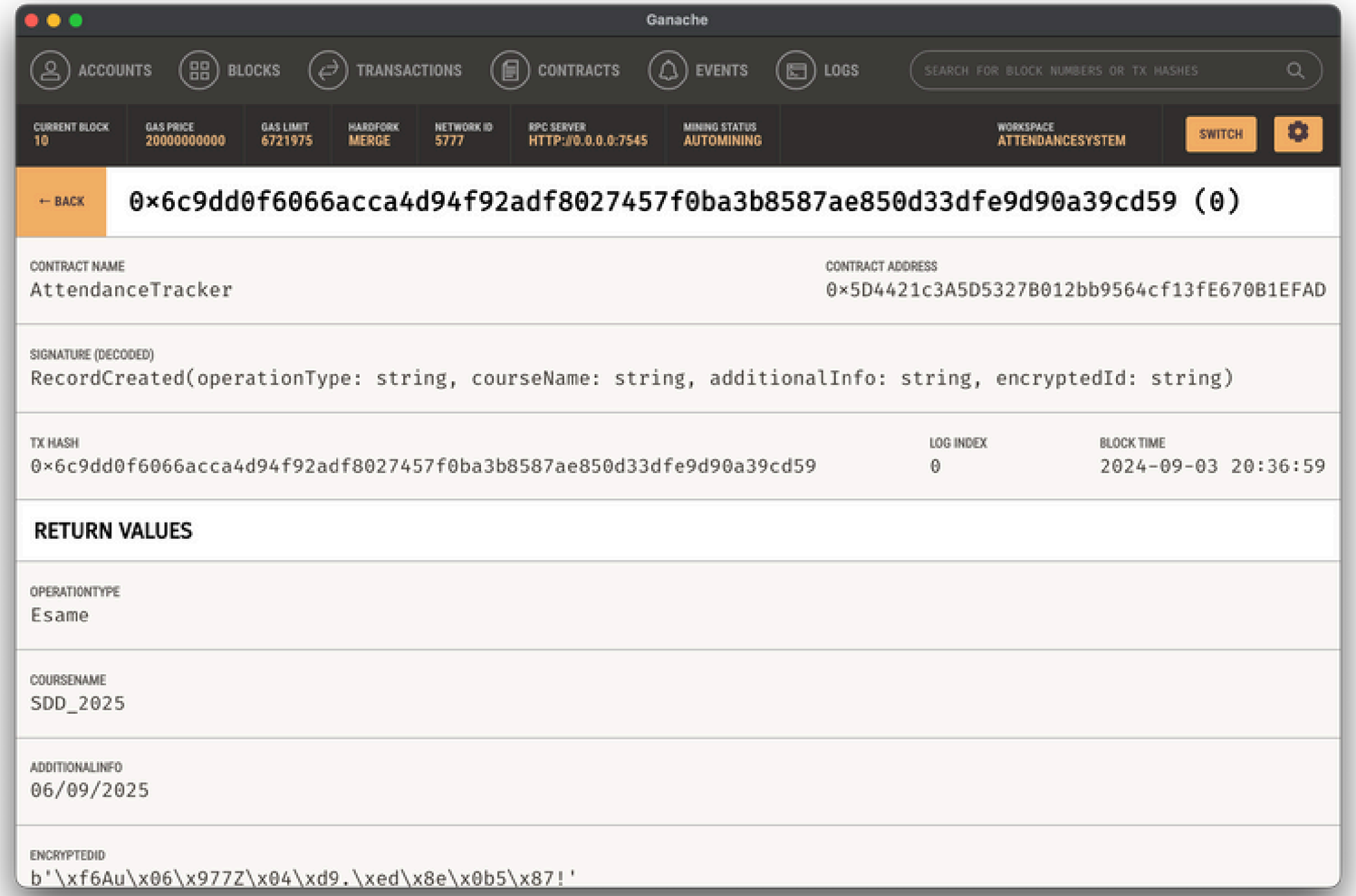
BlockChain

GANACHE



BlockChain

GANACHE



Servizi Google

Servizi Google

- Google Drive: Archiviazione e gestione dei file;

Servizi Google

- Google Drive: Archiviazione e gestione dei file;
- Google Sheets: Creazione e aggiornamento di fogli di calcolo per la registrazione dei dati;

Servizi Google

- Google Drive: Archiviazione e gestione dei file;
- Google Sheets: Creazione e aggiornamento di fogli di calcolo per la registrazione dei dati;
- Google Cloud: Autenticazione e gestione delle credenziali per l'accesso ai servizi.

Servizi Google

createGsheet.py

1. *Avvia il flusso di autenticazione dell'utente ai servizi Google.*

2. *Crea una cartella per il Corso.*

Servizi Google

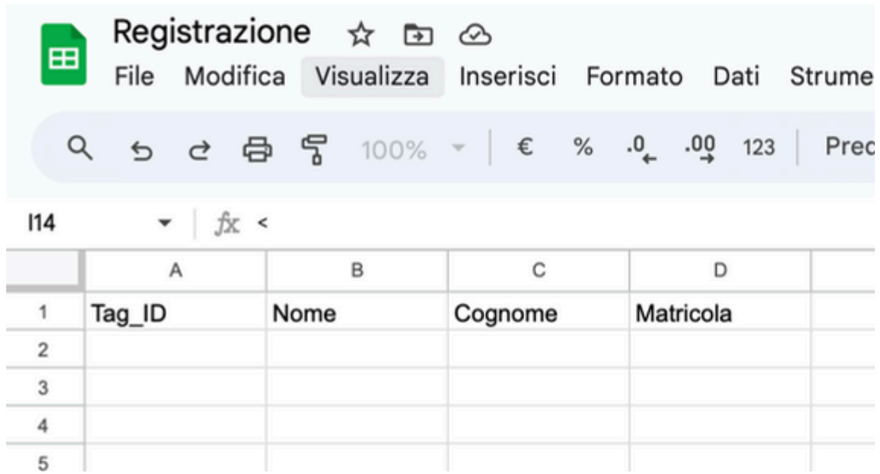
createGsheet.py

3. Crea il foglio di calcolo, con intestazioni diverse in base al tipo di operazione, pronto per contenere i dati degli studenti:

- *iscritti al corso;*
- *partercipanti alla lezione;*
- *iscritti all'esame.*

Servizi Google

createGsheet.py



Servizi Google

updateGsheet.py

1. *Avvia il flusso di autenticazione dell'utente ai servizi Google.*

2. *Cerca la cartella del Corso.*

Servizi Google

updateGsheet.py

3. Inserisce nel foglio di calcolo i dati degli studenti:

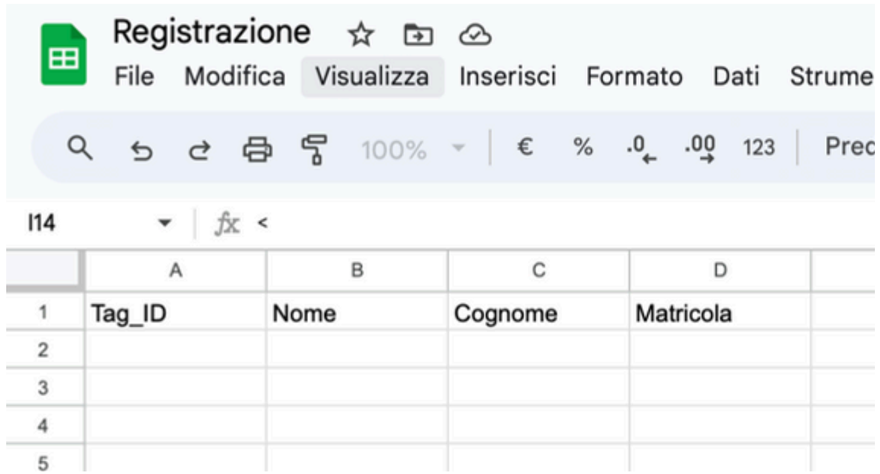
Per i nuovi iscritti al corso:
- *ID, Nome, Cognome, Matricola*

Per i partercipanti alla lezione:
- *ID, Nome, Cognome, Matricola, Orario di Arrivo*

Per gli iscritti all'esame:
- *ID, Nome, Cognome, Matricola*

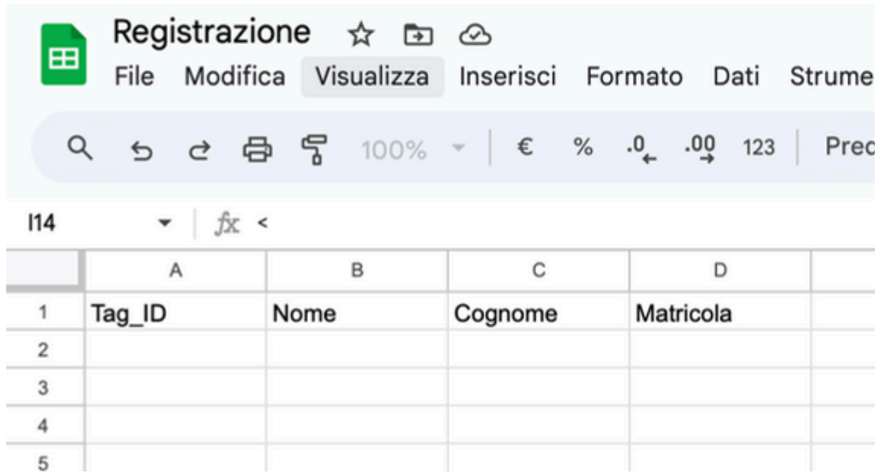
Servizi Google

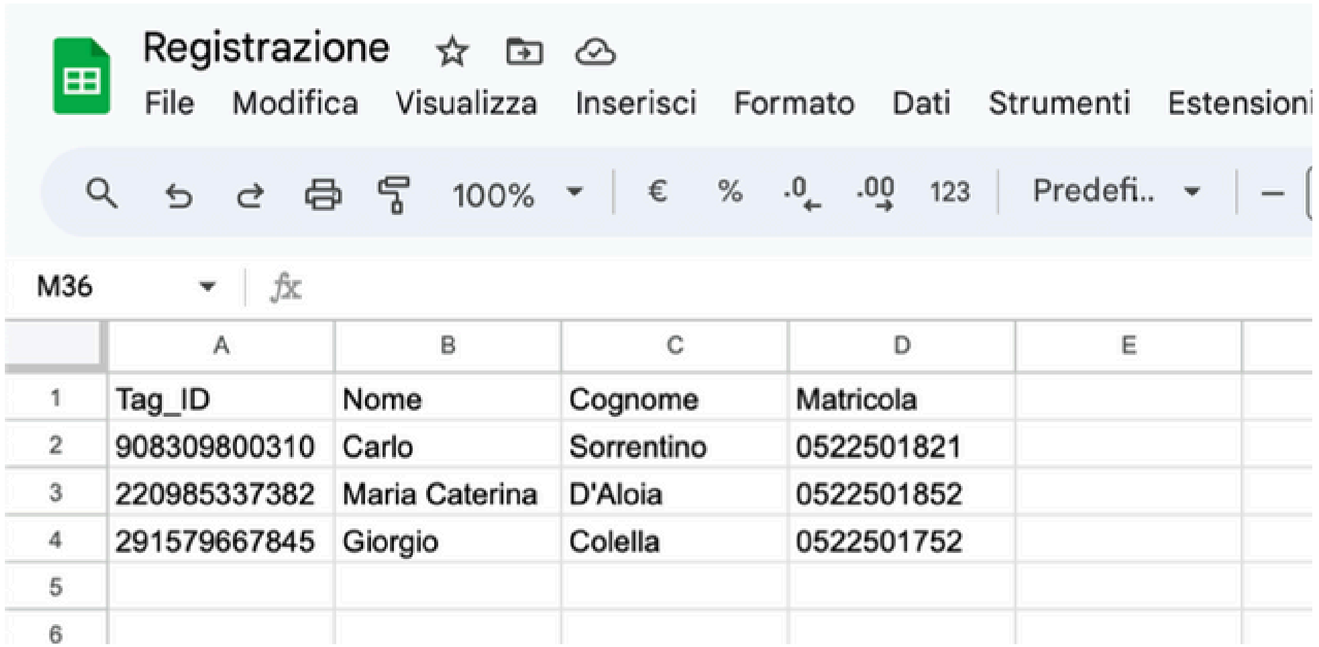
updateGsheet.py



Servizi Google

updateGsheet.py





Servizi Google

keyChainGsheet.py

1. *Avvia il flusso di autenticazione dell'utente ai servizi Google.*

2. *Cerca la cartella del Corso.*

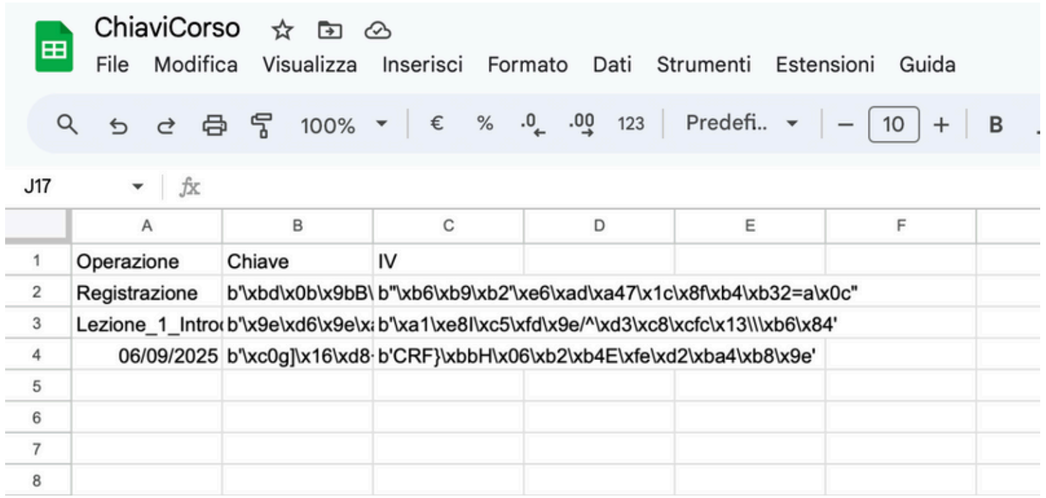
Servizi Google

keyChainGsheet.py

3. Cosa restituisce?
   - *Se il Wallet non esiste crea il foglio di calcolo con intestazioni: Operazione, Chiave ed IV ed inserisce nei rispettivi campi i dati appena creati e li restituisce;*
   - *Se il Wallet esiste ed esiste anche l'operazione passata restituisce direttamente la Chiave e l'IV dell'Operazione del Corso richiesto;*
   - *Se il Wallet esiste ma non esiste l'operazione passata inserisce nel Wallet una nuova Chiave ed IV che poi restituisce.*

Servizi Google

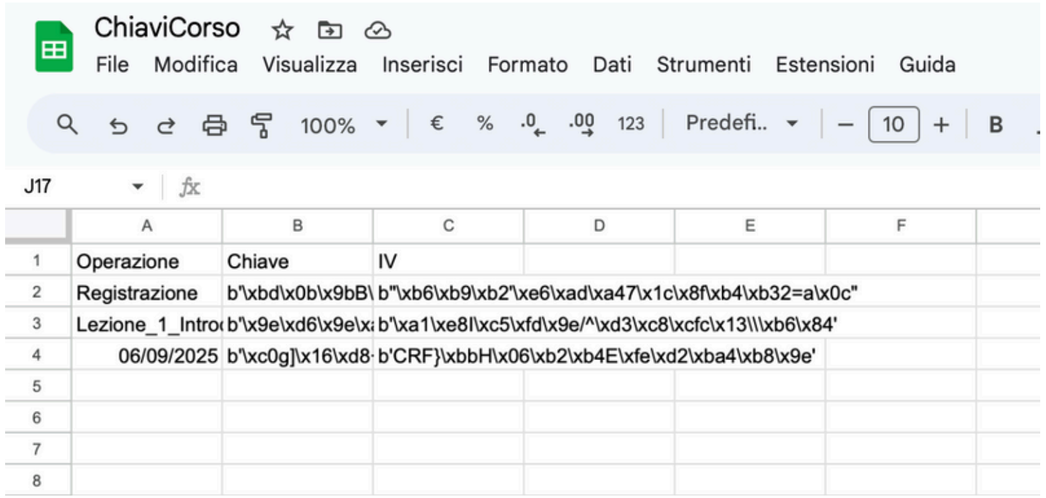
keyChainGsheet.py



Servizi Google

keyChainGsheet.py

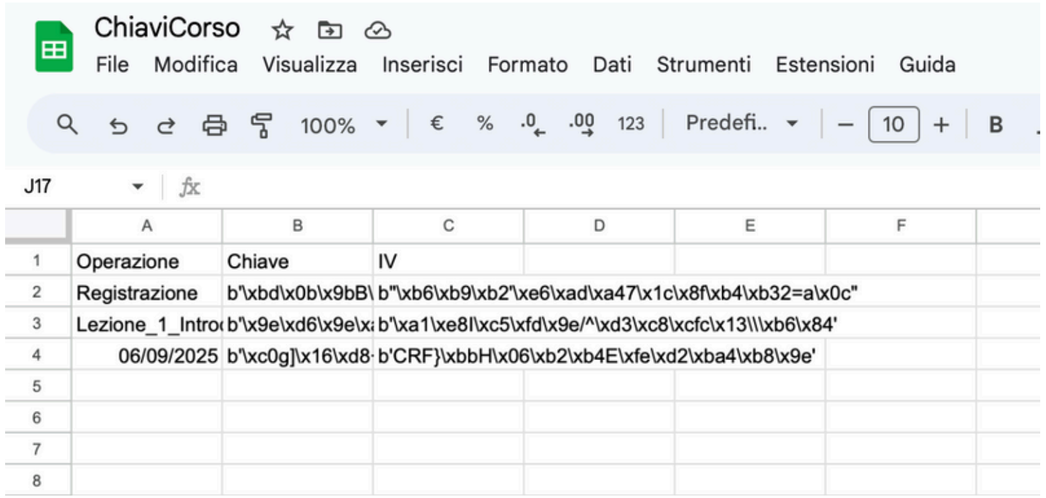
Insert Key&IV



Servizi Google

keyChainGsheet.py

Insert Key&IV



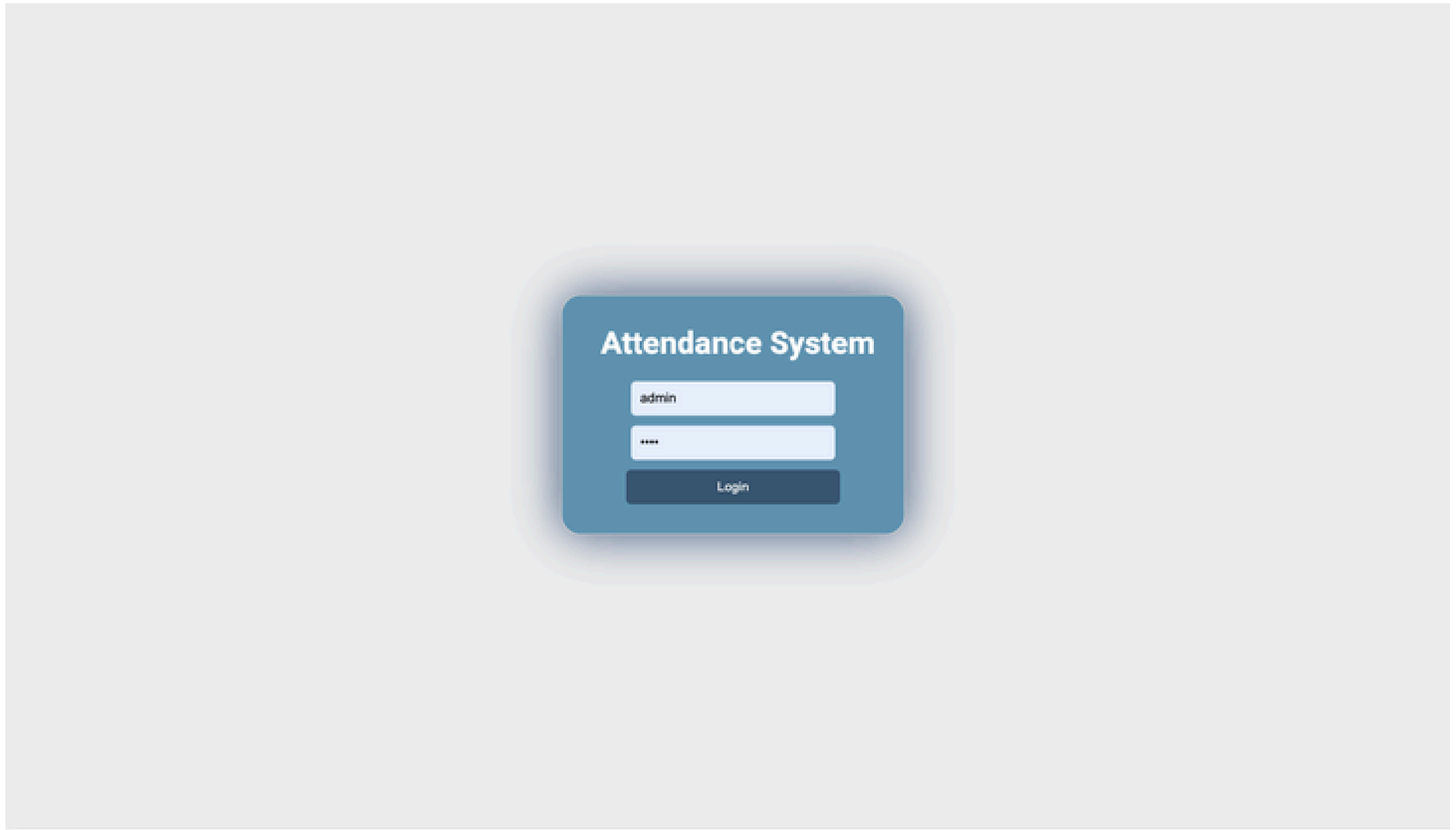
Return Key&IV

Servizi Google

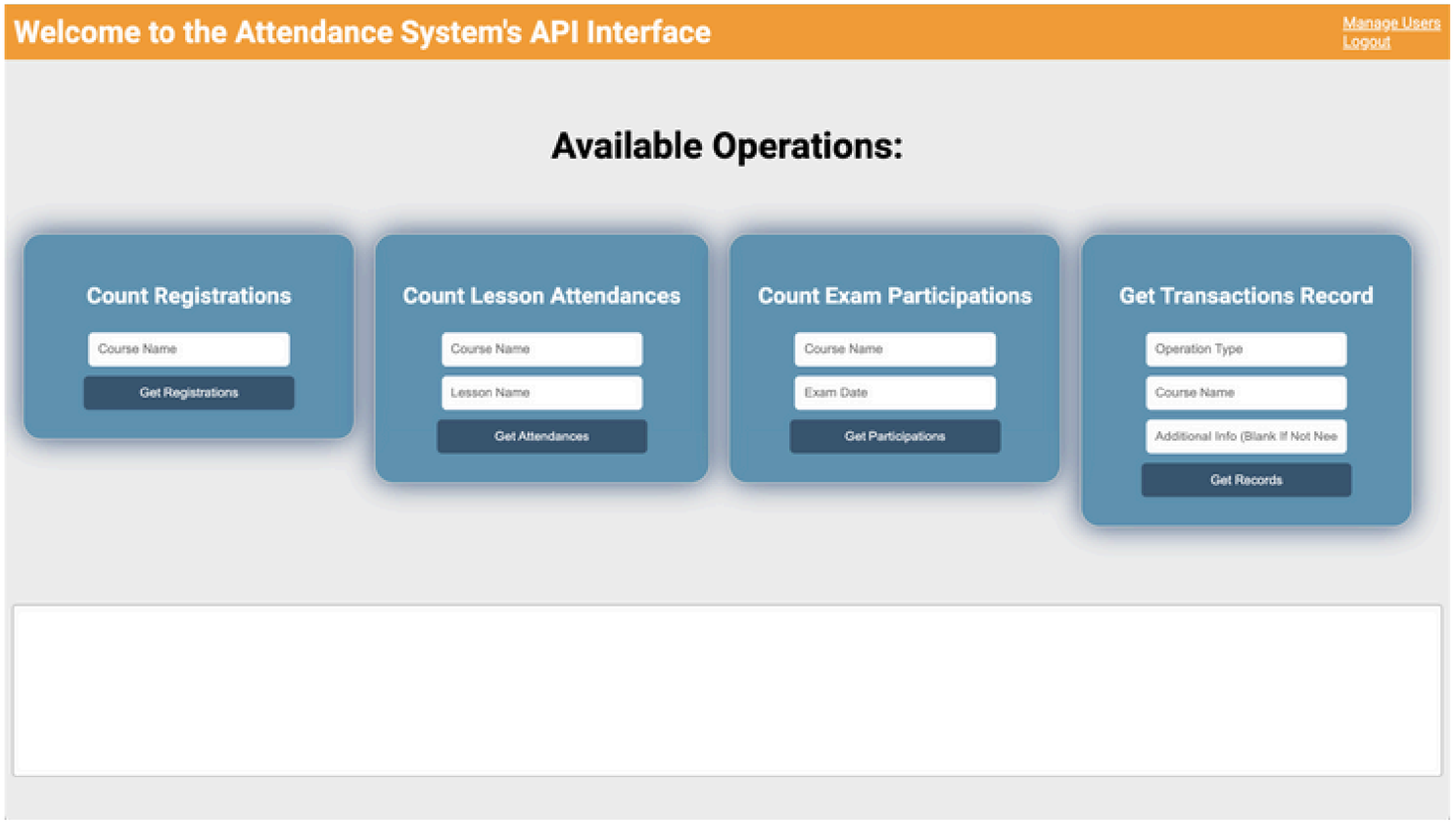
Risultato Finale

Interfaccia di Monitoraggio

Interfaccia di Monitoraggio



Interfaccia di Monitoraggio



Interfaccia di Monitoraggio

01 *Count Registration*

02 *Count Lesson Attendances*

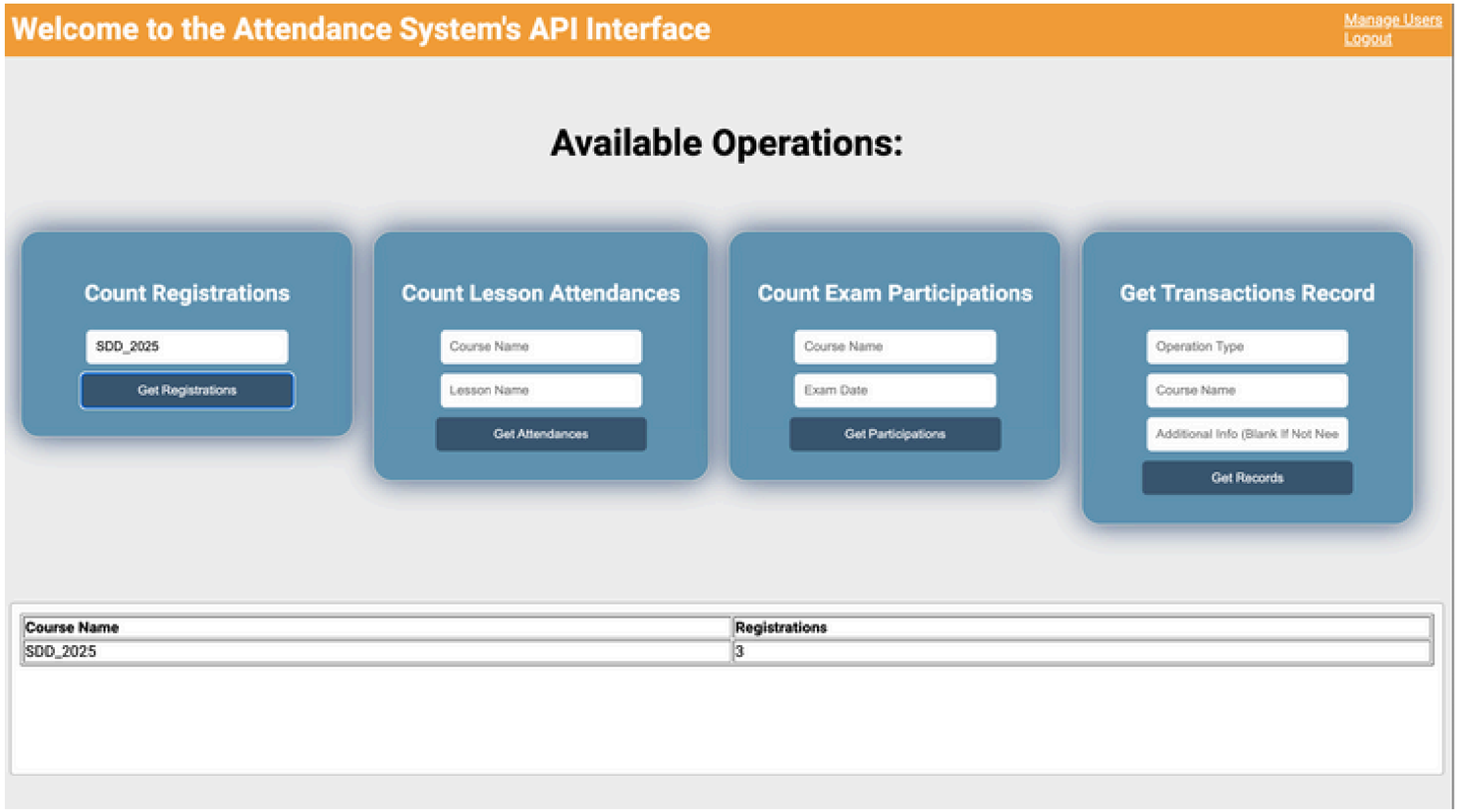
03 *Count Exam Participations*

04 *Count Lesson Attendances*

05 *Manage Users*

Interfaccia di Monitoraggio

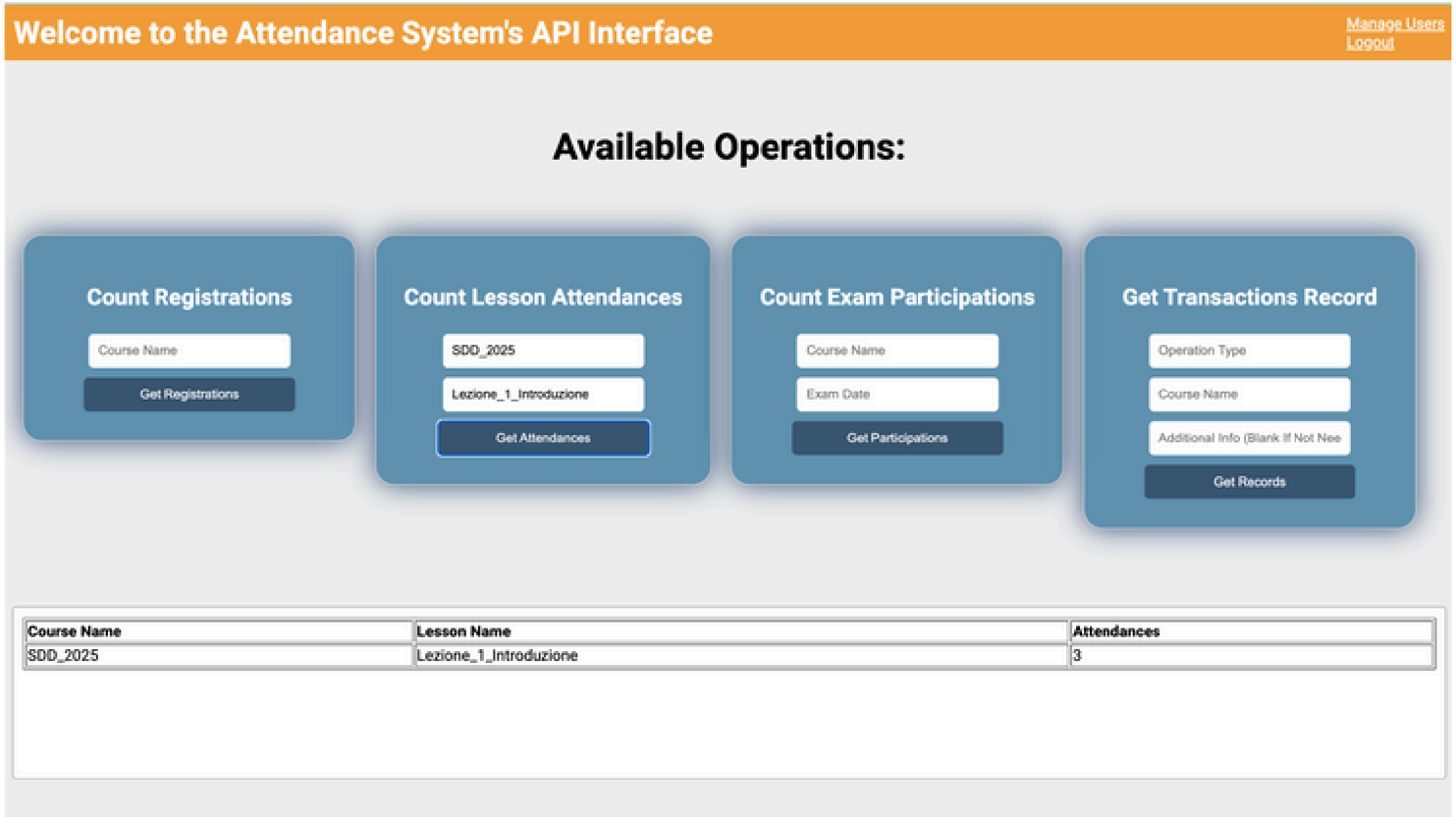
01 **Count Registration**



Interfaccia di Monitoraggio

02

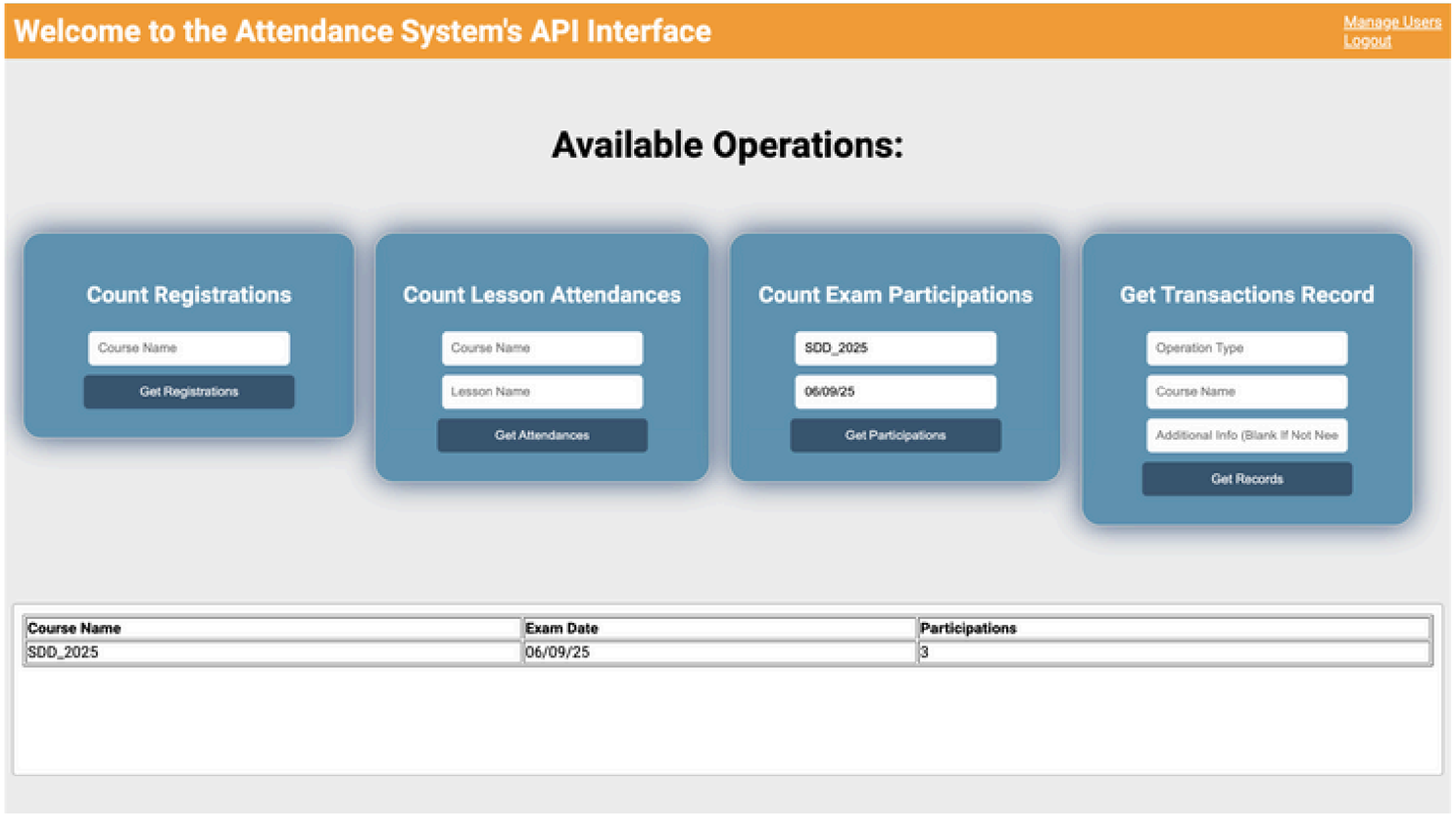
Count Lesson Attendances



Interfaccia di Monitoraggio

03

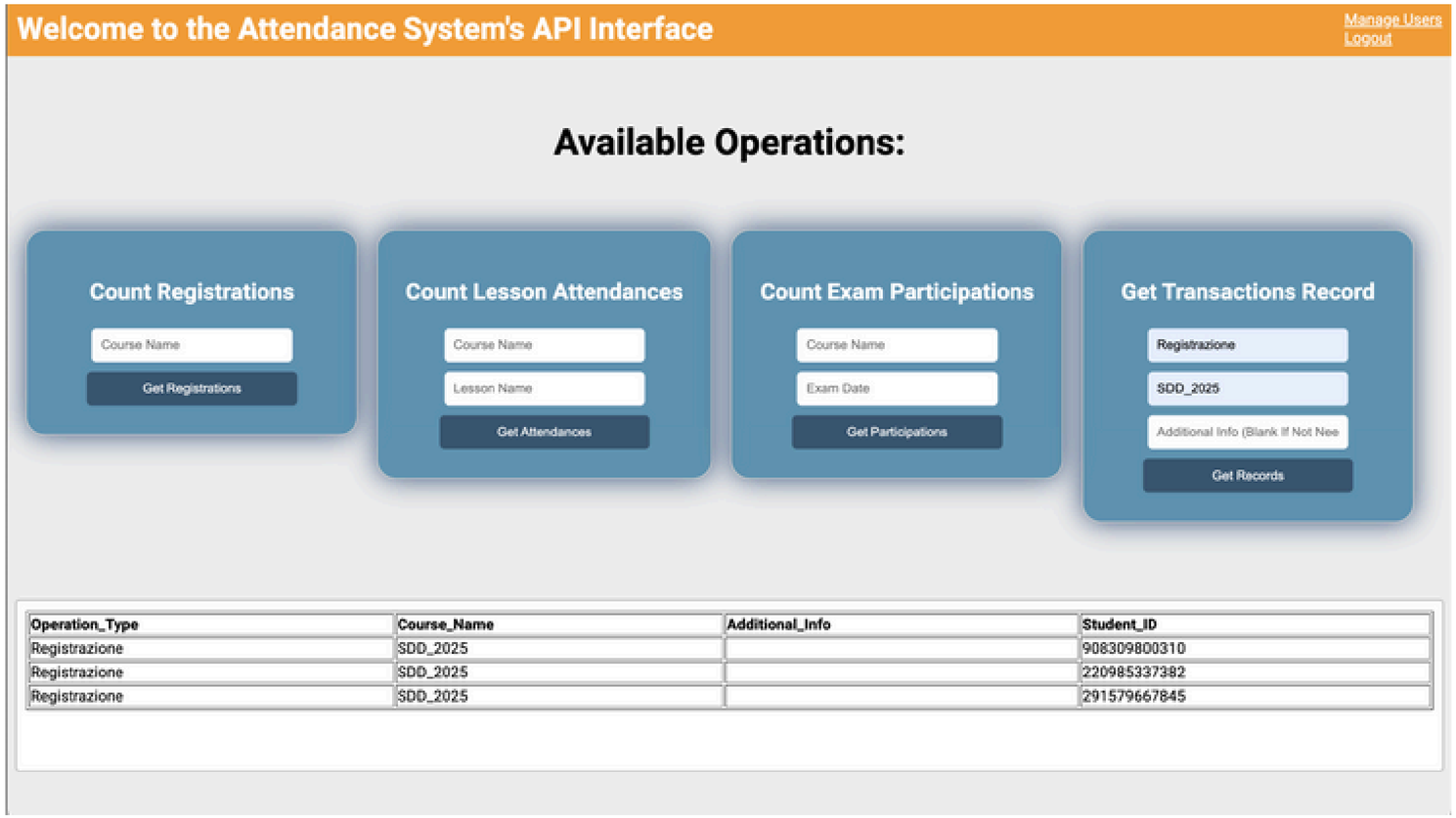
Count Exam Participations



Interfaccia di Monitoraggio

04

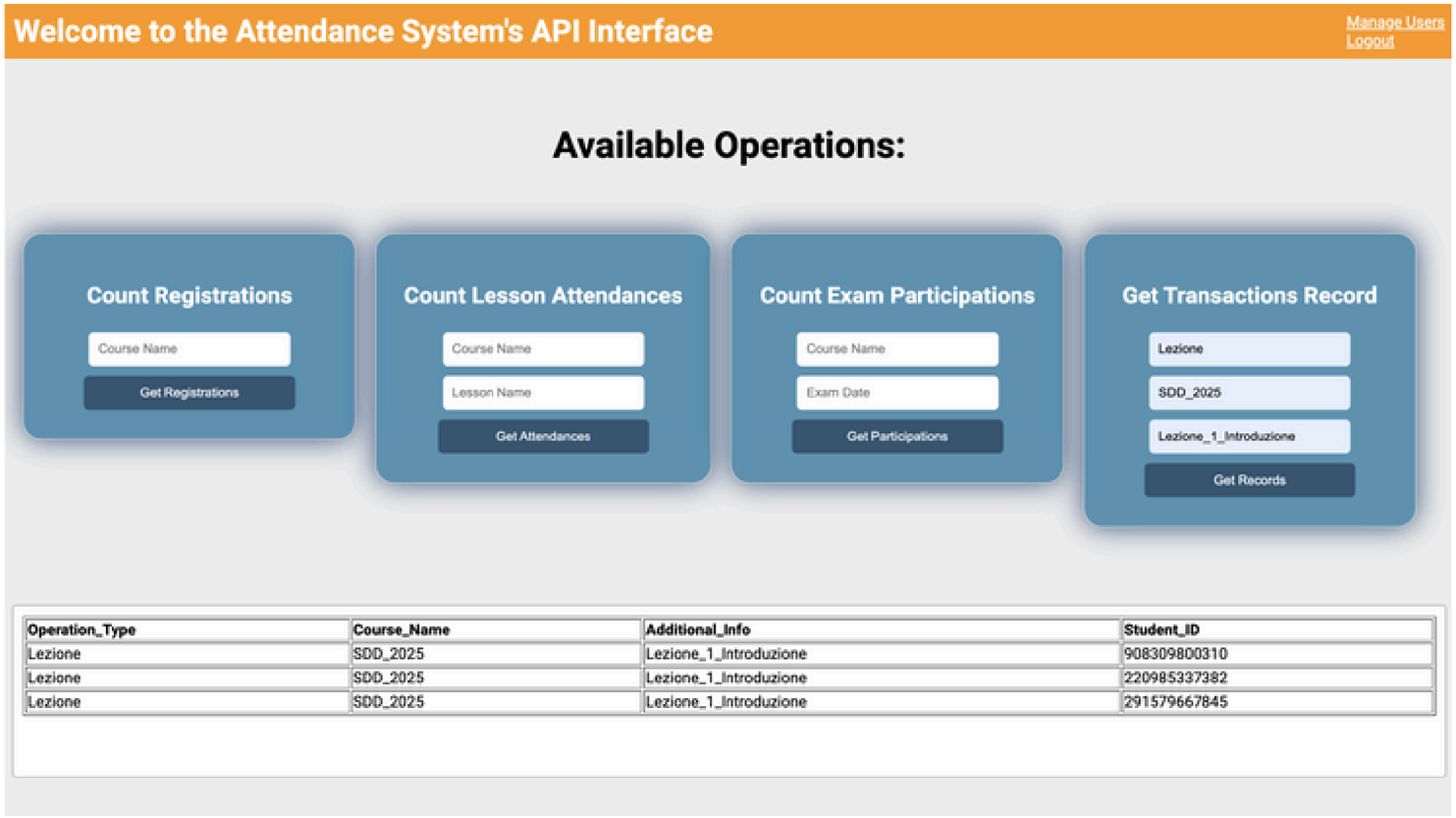
Get Transactions Record Registrazione Corso



Interfaccia di Monitoraggio

04

Get Transactions Record Lezione

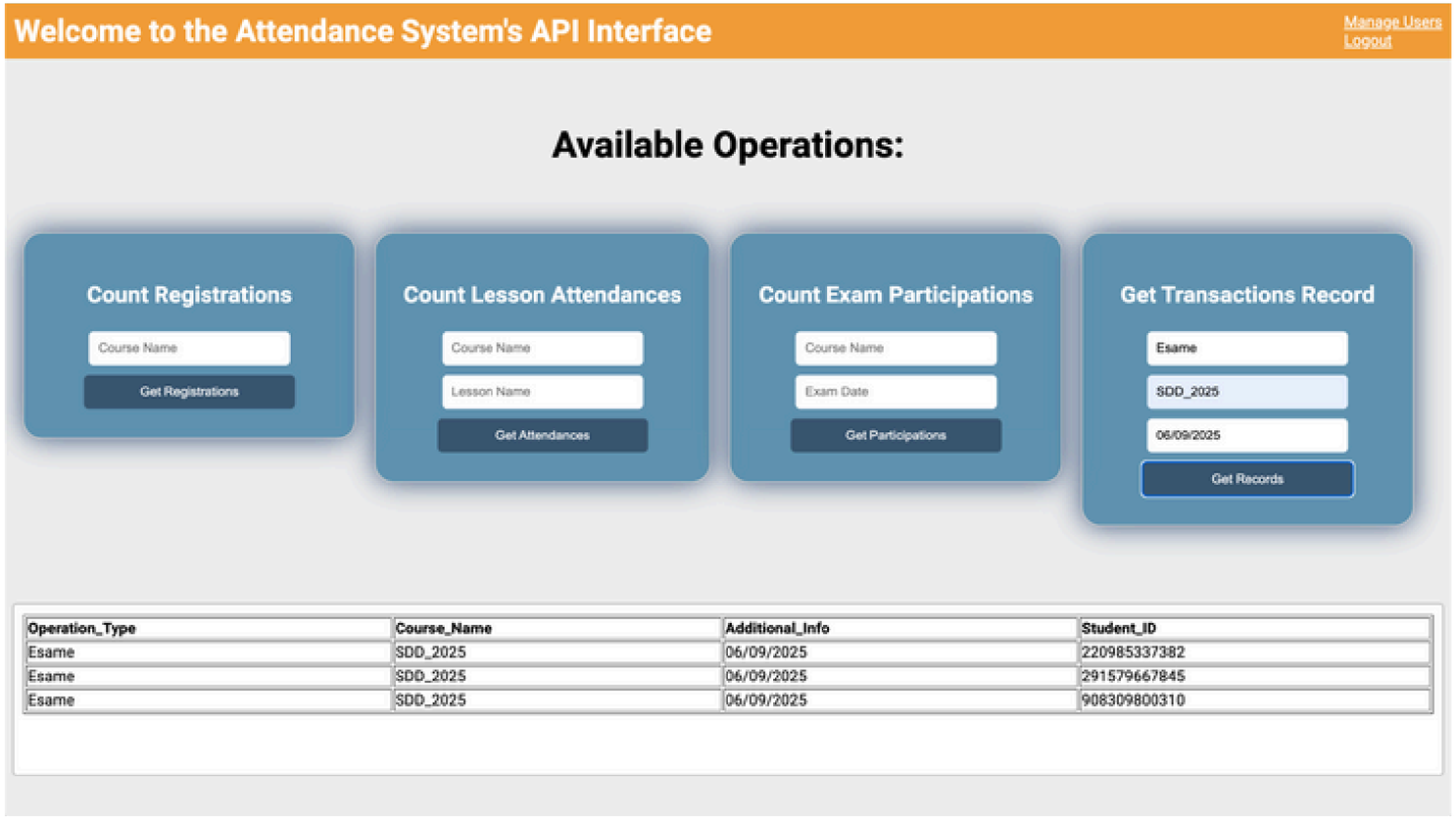


Operation_Type	Course_Name	Additional_Info	Student_ID
Lezione	SDD_2025	Lezione_1_Introduzione	908309800310
Lezione	SDD_2025	Lezione_1_Introduzione	220985337382
Lezione	SDD_2025	Lezione_1_Introduzione	291579667845

Interfaccia di Monitoraggio

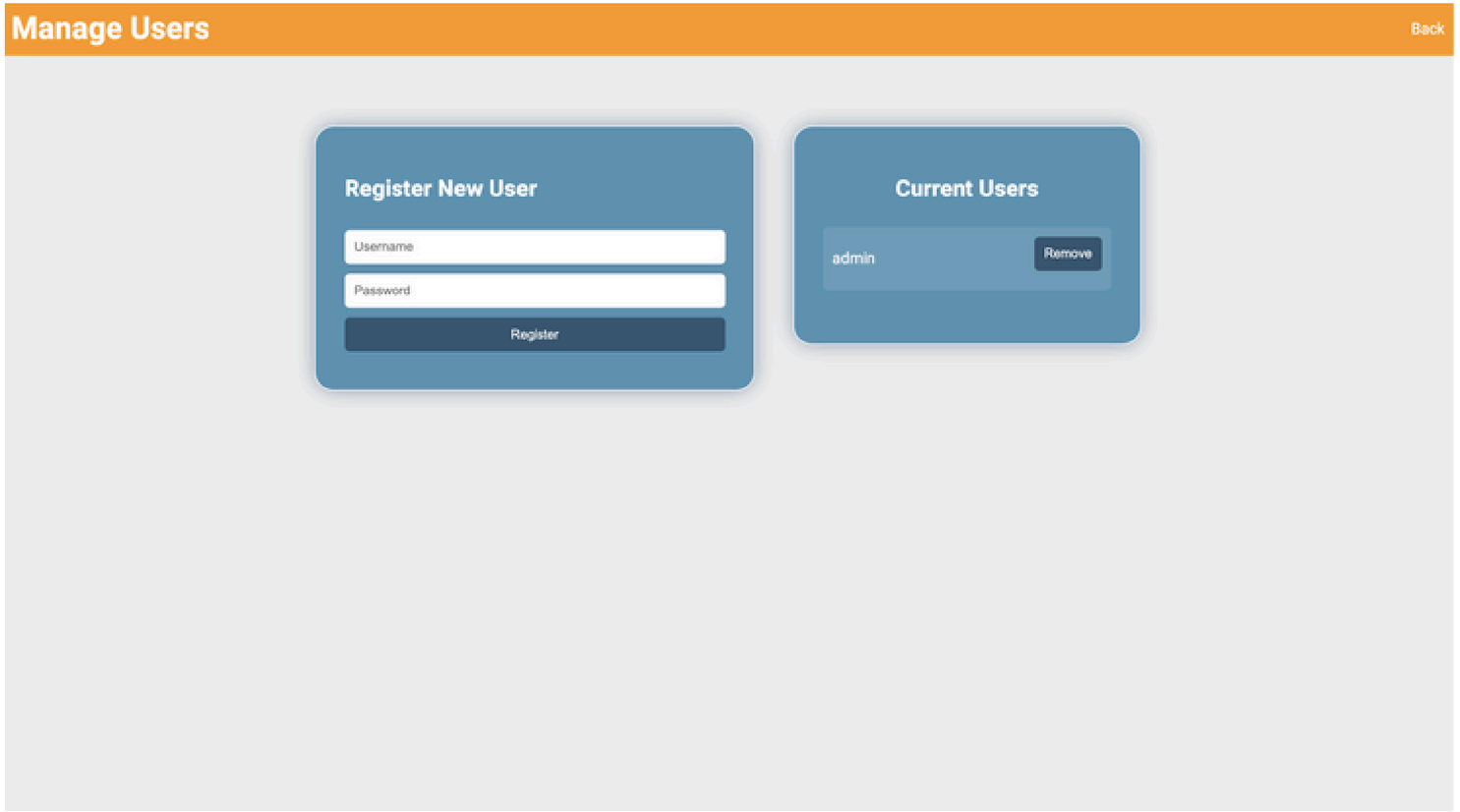
04

Get Transactions Record Esame



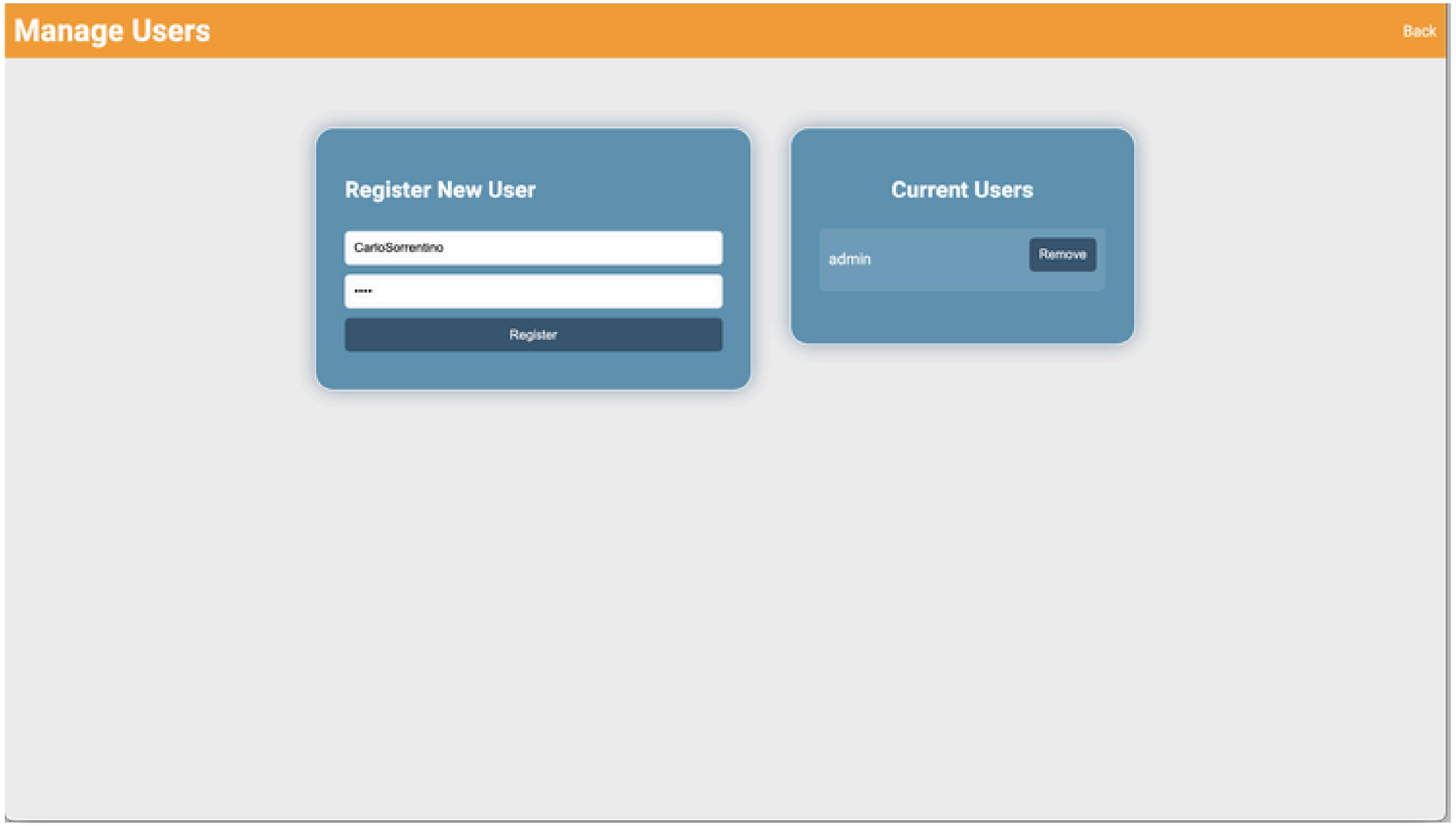
Interfaccia di Monitoraggio

05 Manage Users



Interfaccia di Monitoraggio

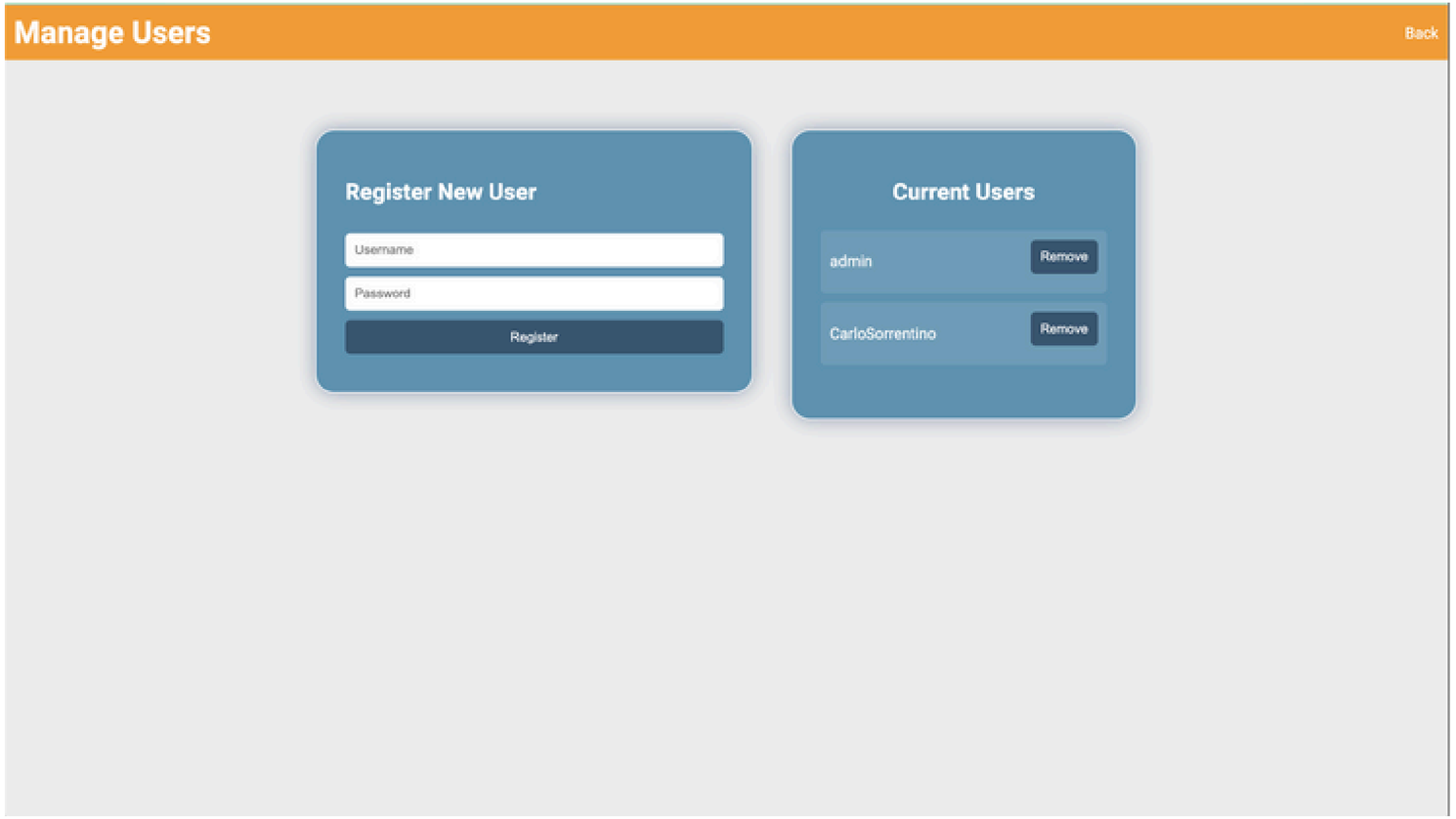
05 Manage Users



Interfaccia di Monitoraggio

05

Manage Users



Interfaccia di Monitoraggio

Misure di Sicurezza Adottate

- *HTTPS per garantire la trasmissione sicura dei dati; di fatti i cookie di sessione siano cifrati e contrassegnati con attributi HttpOnly e Secure.*
- *JWT (JSON Web Token) sono utilizzati per autenticare gli utenti senza mantenere sessioni lato server. La protezione CSRF con JWT sui cookie garantisce che solo richieste legittime provenienti dal client autentico possano accedere alle risorse del server.*
- *CSRF (Cross-Site Request Forgery) impedisce quindi che utenti malintenzionati eseguano azioni indesiderate a nome di un utente autenticato poiché ogni richiesta che modifica lo stato del server (come un'operazione di registrazione o modifica dati) deve includere un token univoco e non prevedibile generato dal server.*
- *Hashing delle password degli Users.*
- *Le richieste AJAX nei file HTML gestiscono efficacemente gli errori, catturando le richieste fallite e mostrando messaggi di errore all'utente nel file app.log degli eventi critici tramite Loggers.*

Interfaccia di Monitoraggio

Altre Misure di Sicurezza Fondamentali

- *Sanitizzazione degli input con la libreria bleach che viene utilizzata per eliminare codice potenzialmente dannoso dagli input degli utenti, prevenendo attacchi di tipo XSS (Cross-Site Scripting). Sanitizzando l'input, si evita che l'utente possa iniettare codice malevolo, come script che potrebbero essere eseguiti nel contesto della pagina web di altri utenti.*

- *Validazione degli input con WTForms con check dei dati lato server il quale assicura che i campi del form contengano dati validi e che rispettino regole specifiche (come lunghezza minima e massima). Questo riduce il rischio che input malevoli o formati errati vengano inviati al server.*

Flusso Operativo Completo

01

Lo studente scansiona il tag attraverso il lettore collegato al RaspBerry Pi.

Flusso Operativo Completo

01 Lo studente scansiona il tag attraverso il lettore collegato al RaspBerry Pi.

02 Vengono recuperati tutti i suoi dati personali.

Flusso Operativo Completo

Flusso Operativo Completo

01 Lo studente scansiona il tag attraverso il lettore collegato al RaspBerry Pi.

02 Vengono recuperati tutti i suoi dati personali.

03 Il RaspBerry Pi cifra l'ID univoco letto dal tag.

Flusso Operativo Completo

01 Lo studente scansiona il tag attraverso il lettore collegato al RaspBerry Pi.

02 Vengono recuperati tutti i suoi dati personali.

03 Il RaspBerry Pi cifra l'ID univoco letto dal tag.

04 Vengono salvate le Chiavi e gli IV usati per poterli riutilizzare in futuro.

Flusso Operativo Completo

01 Lo studente scansiona il tag attraverso il lettore collegato al RaspBerry Pi.

02 Vengono recuperati tutti i suoi dati personali.

03 Il RaspBerry Pi cifra l'ID univoco letto dal tag.

04 Vengono salvate le Chiavi e gli IV usati per poterli riutilizzare in futuro.

05 Il dato cifrato insieme al Nome del Corso, al tipo di Operazione ed info aggiuntive vengono memorizzati sia nella BlockChain che dall'API in Google Sheet.

Flusso Operativo Completo

01 Lo studente scansiona il tag attraverso il lettore collegato al RaspBerry Pi.

02 Vengono recuperati tutti i suoi dati personali.

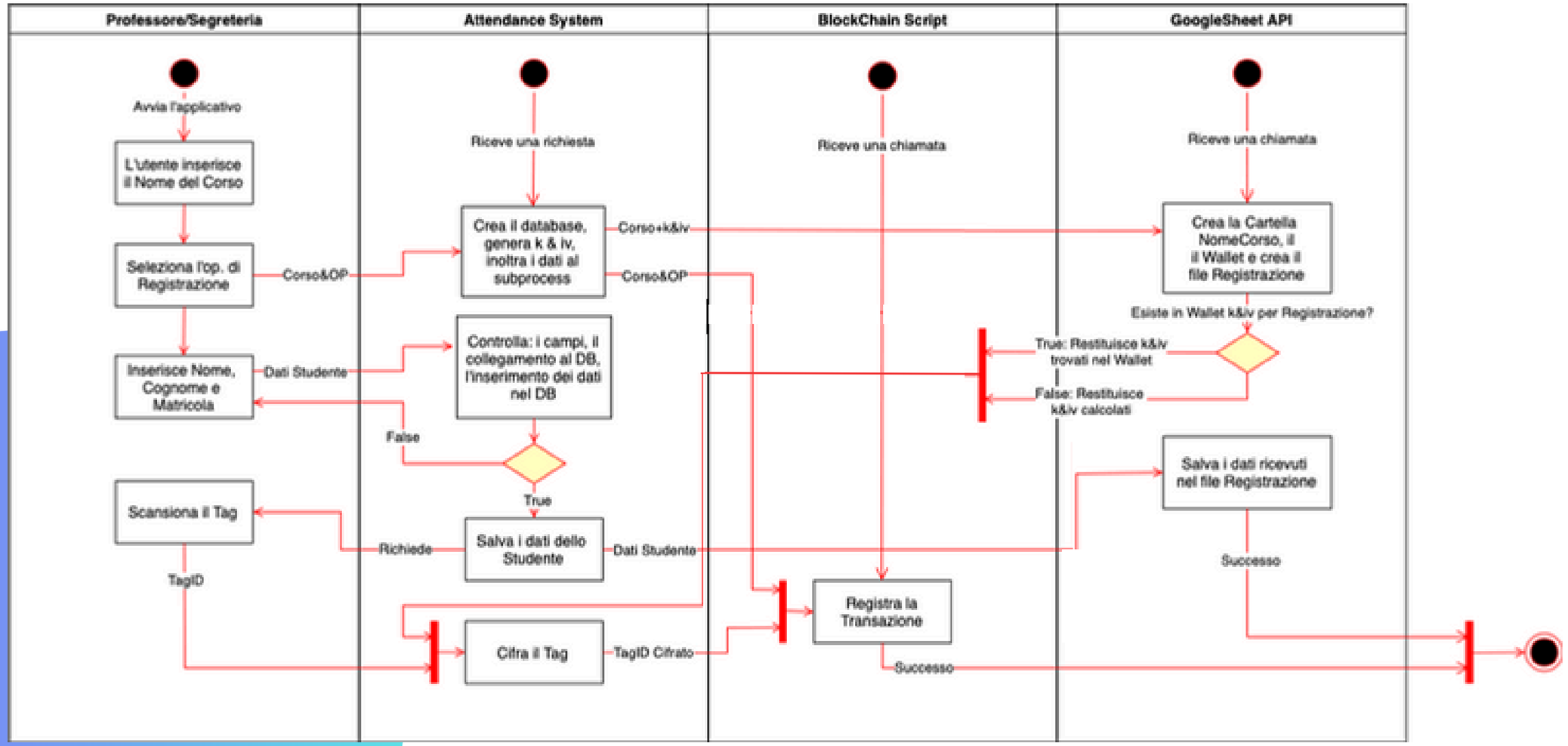
03 Il RaspBerry Pi cifra l'ID univoco letto dal tag.

04 Vengono salvate le Chiavi e gli IV usati per poterli riutilizzare in futuro.

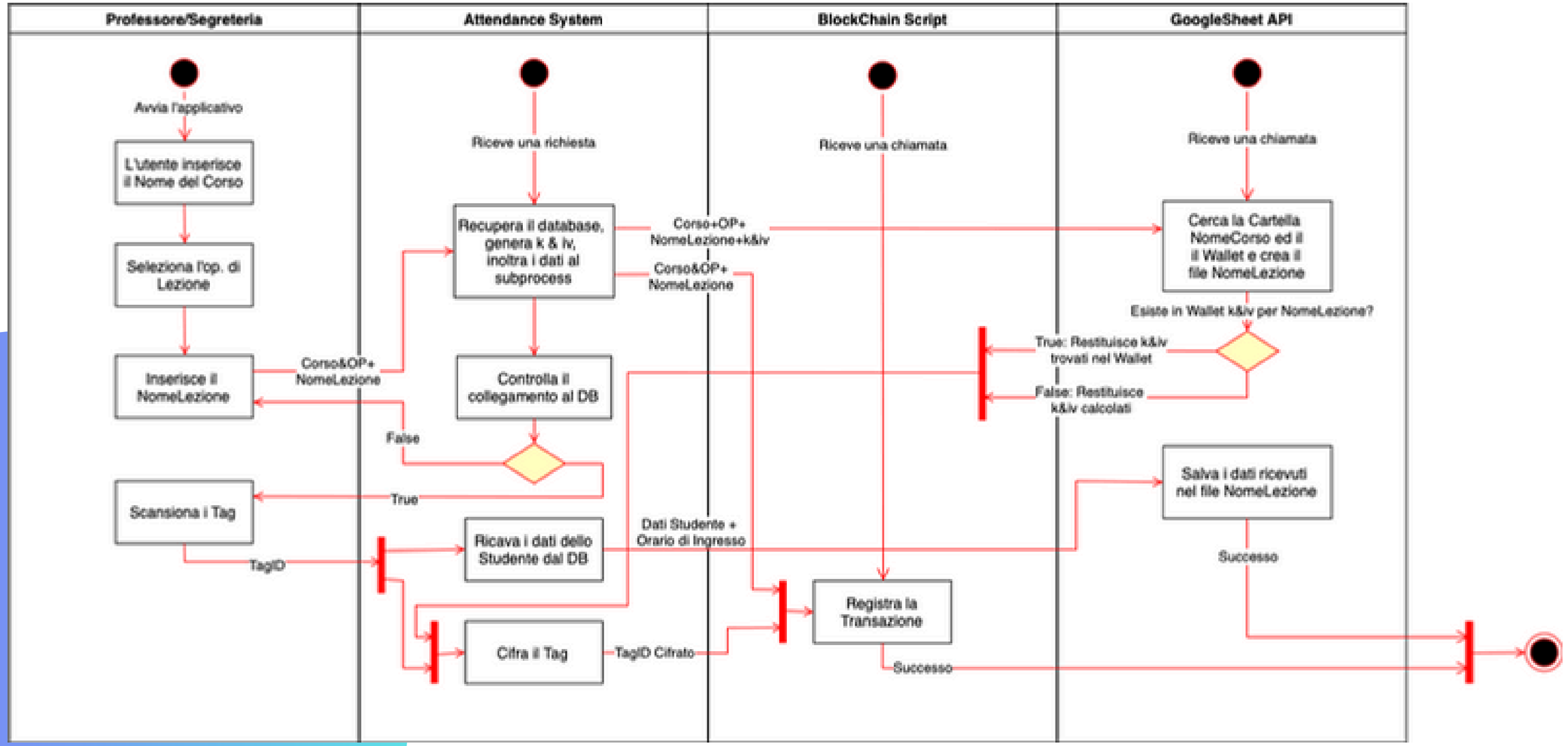
05 Il dato cifrato insieme al Nome del Corso, al tipo di Operazione ed info aggiuntive vengono memorizzati sia nella BlockChain che dall'API in Google Sheet.

06 Questi dati in qualsiasi momento possono essere consultati sia sulla BlockChain che sul proprio Drive Google oppure anche dall'applicazione di monitoraggio in Flask.

Activity Diagram (Registrazione)

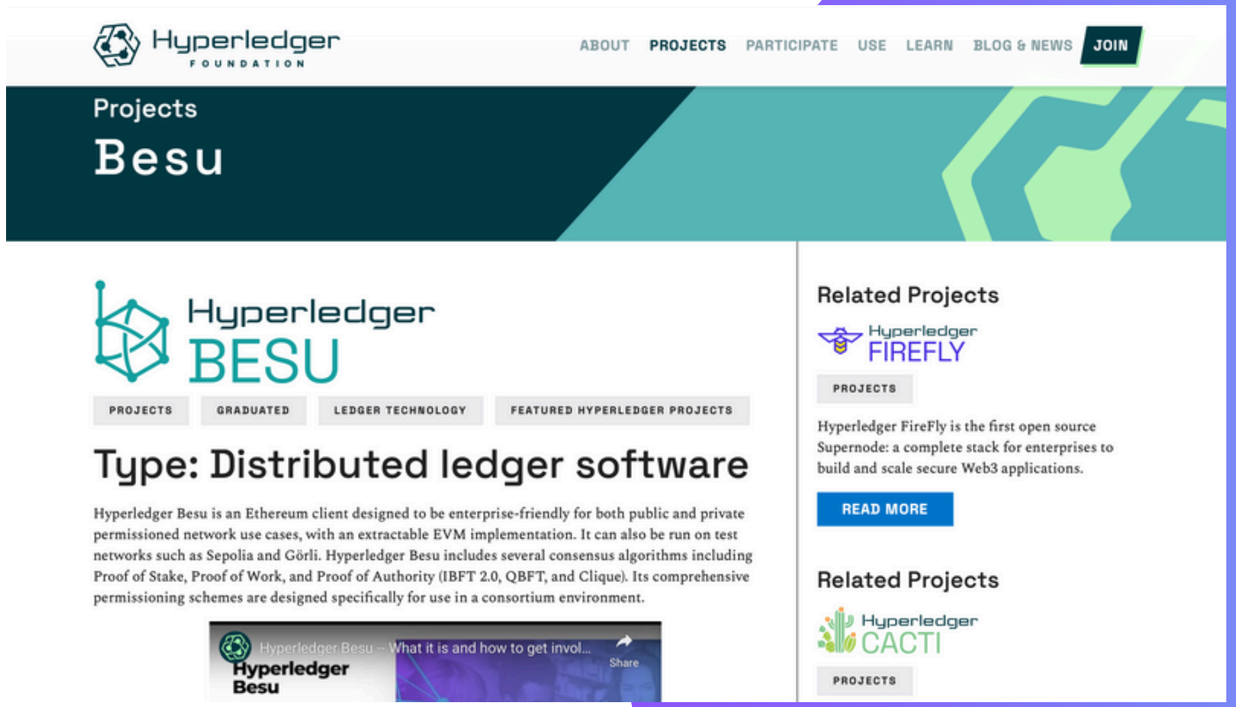


Activity Diagram (Lezione)

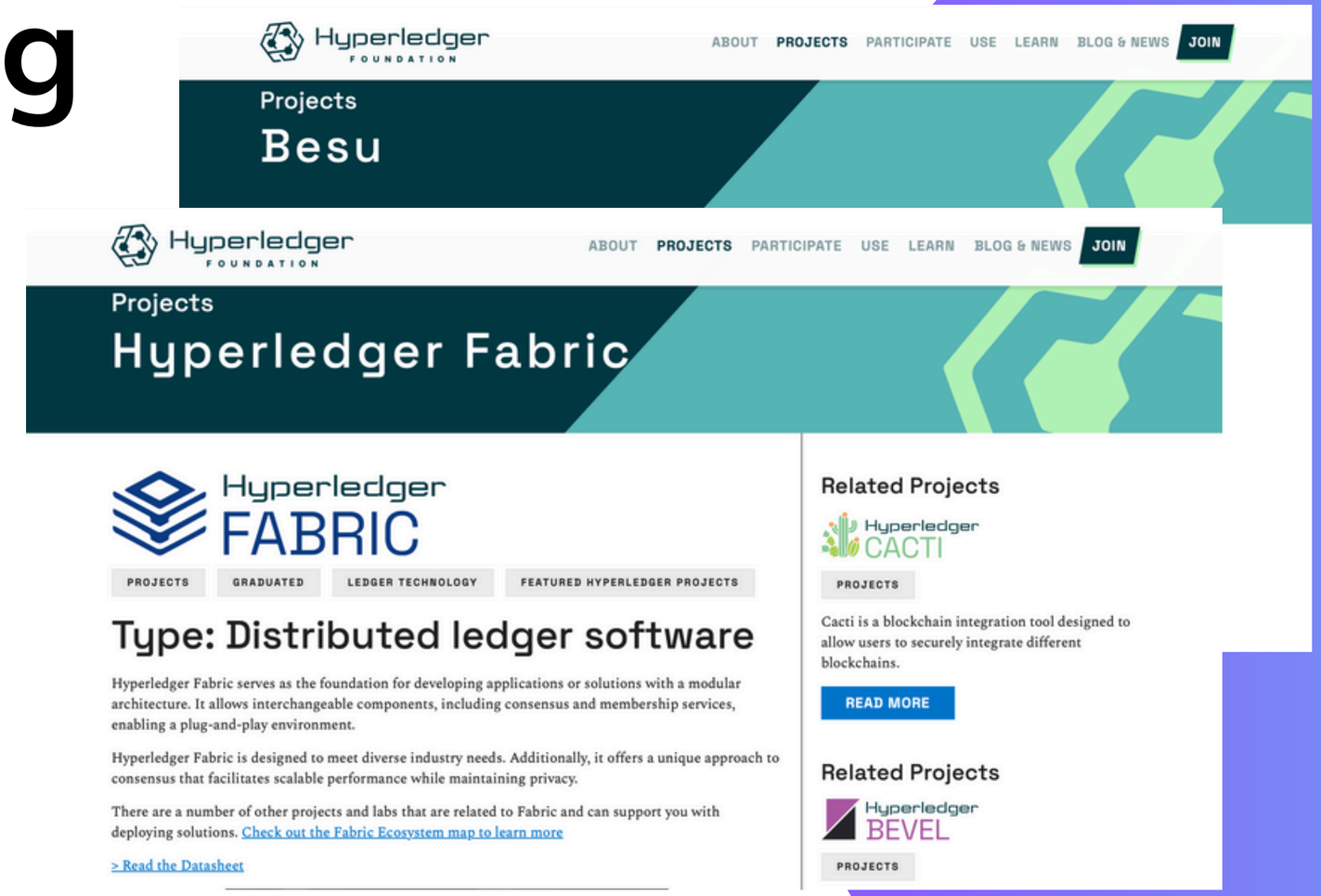


BrainStorming Iniziale & Conclusioni

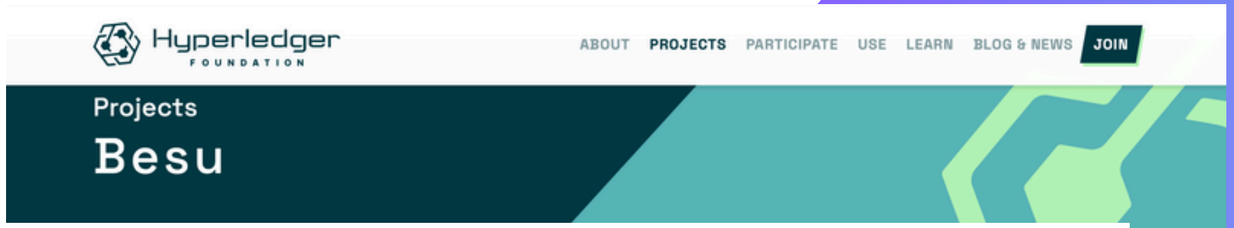
BrainStorming Iniziale & Conclusioni

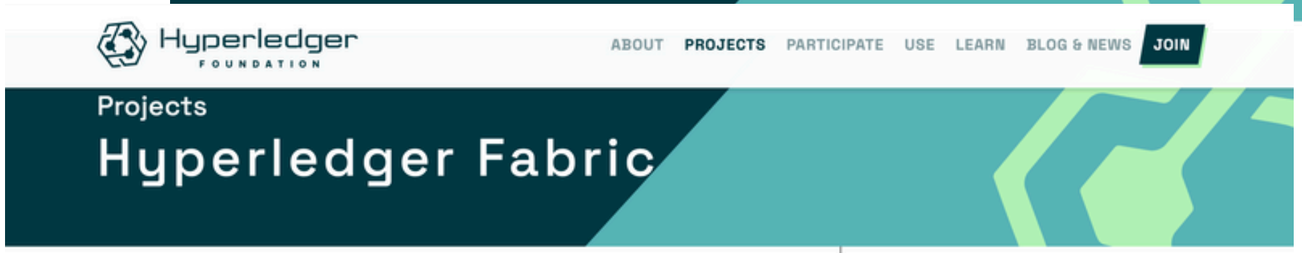


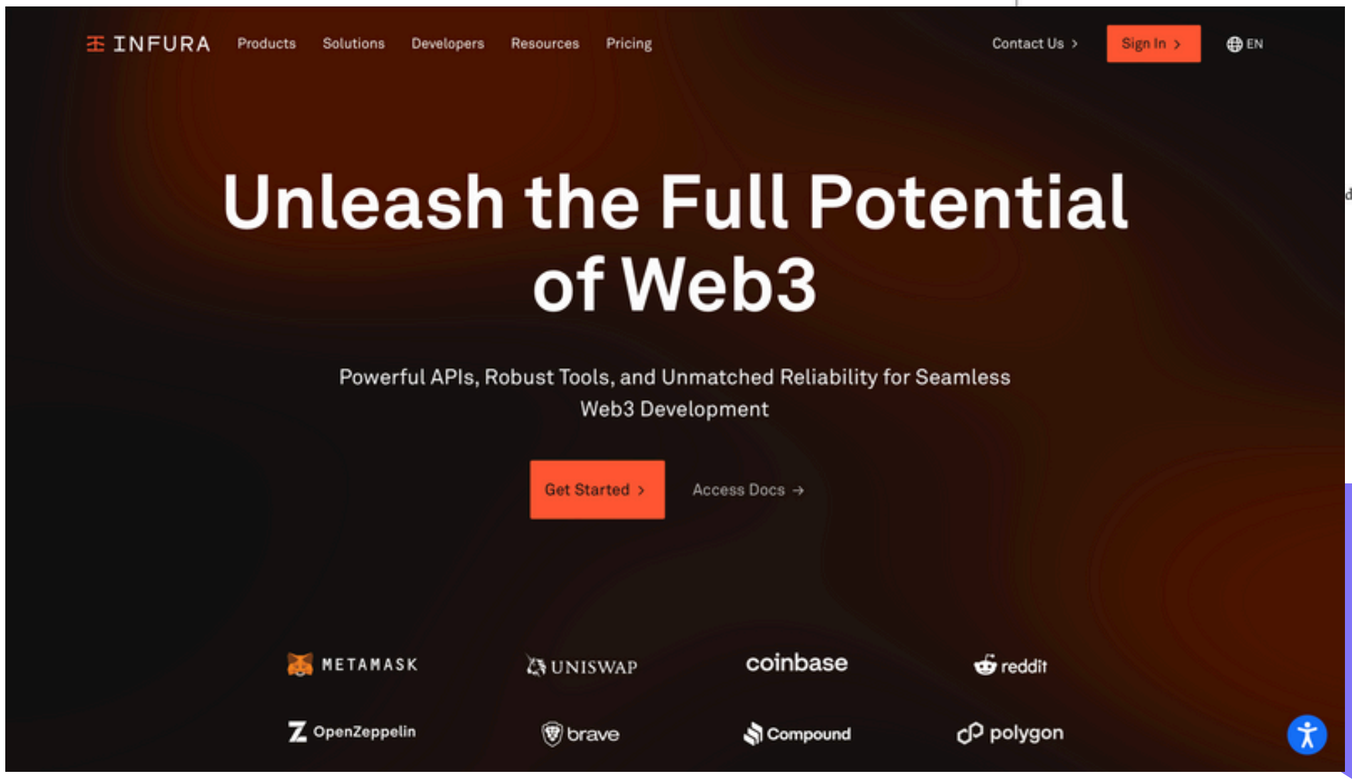
BrainStorming Iniziale & Conclusioni



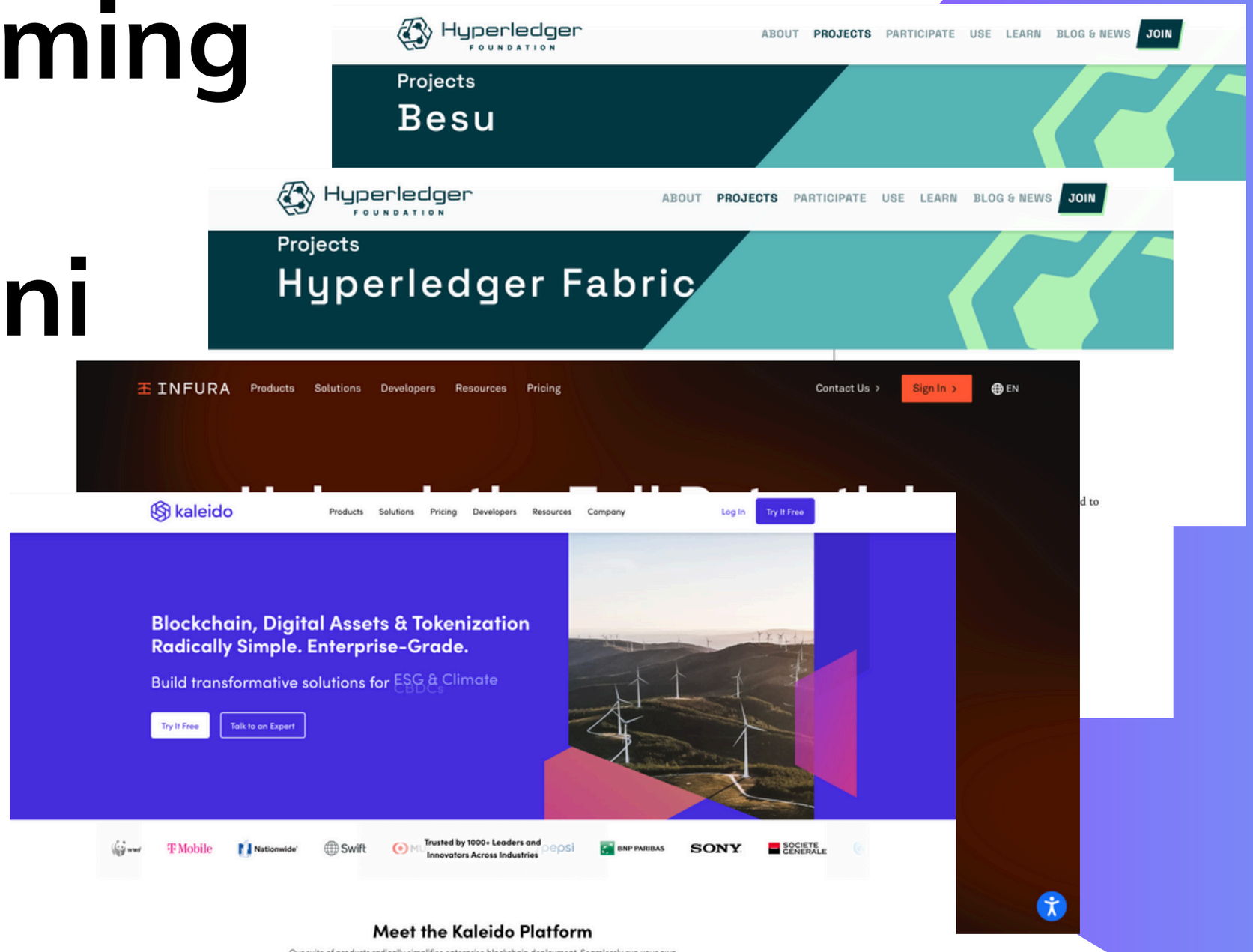
BrainStorming Iniziale & Conclusioni



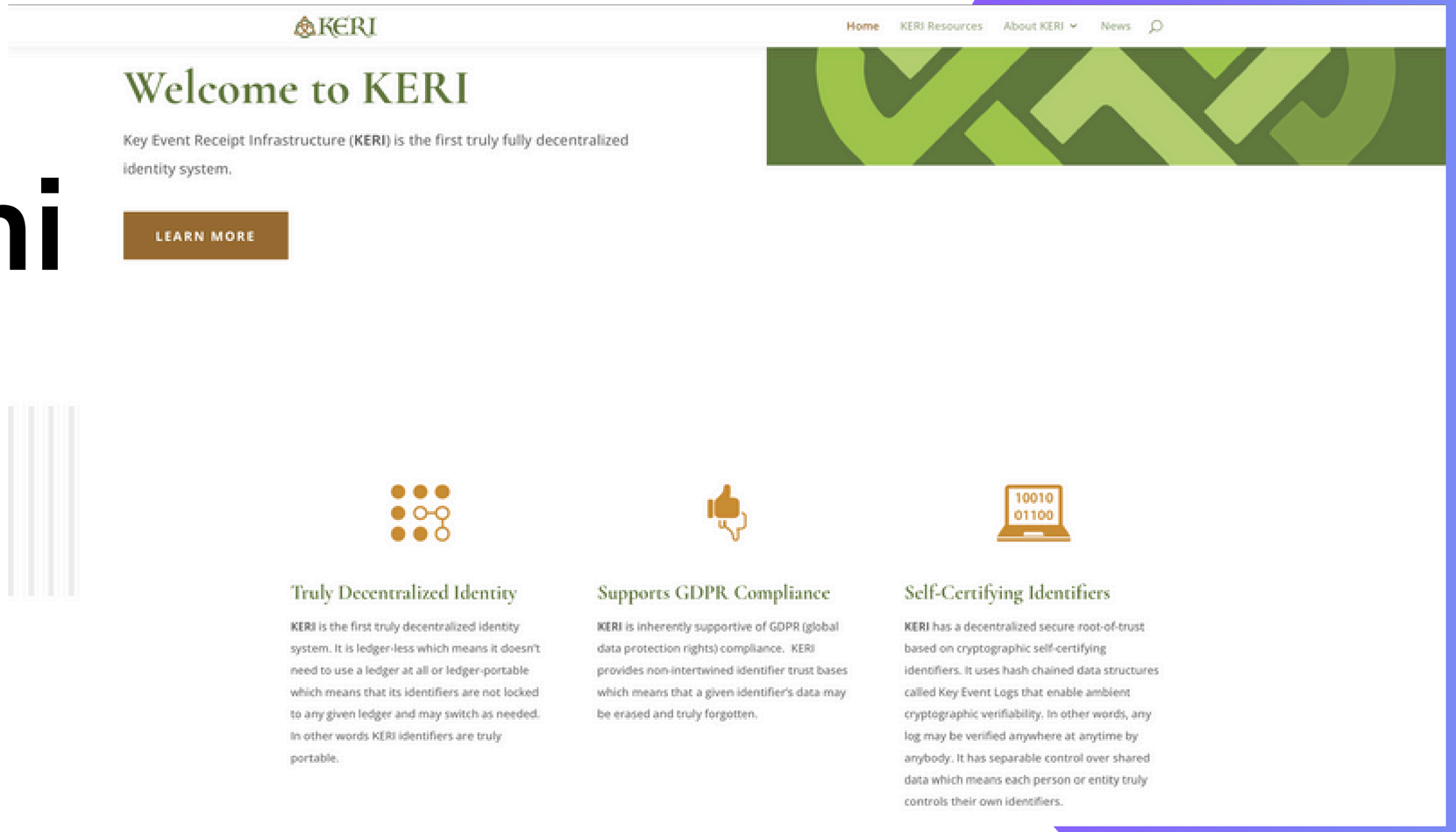




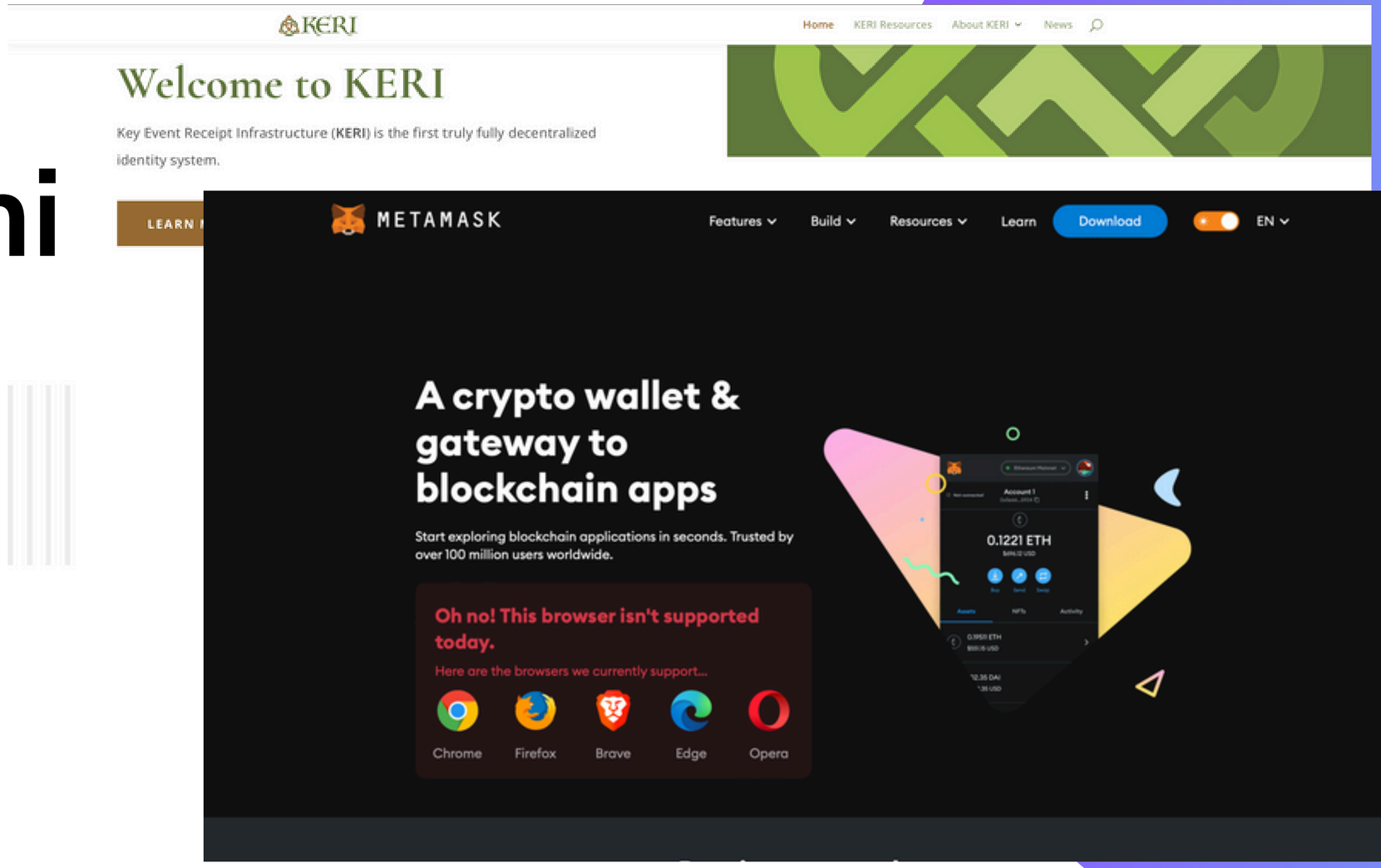
BrainStorming Iniziale & Conclusioni



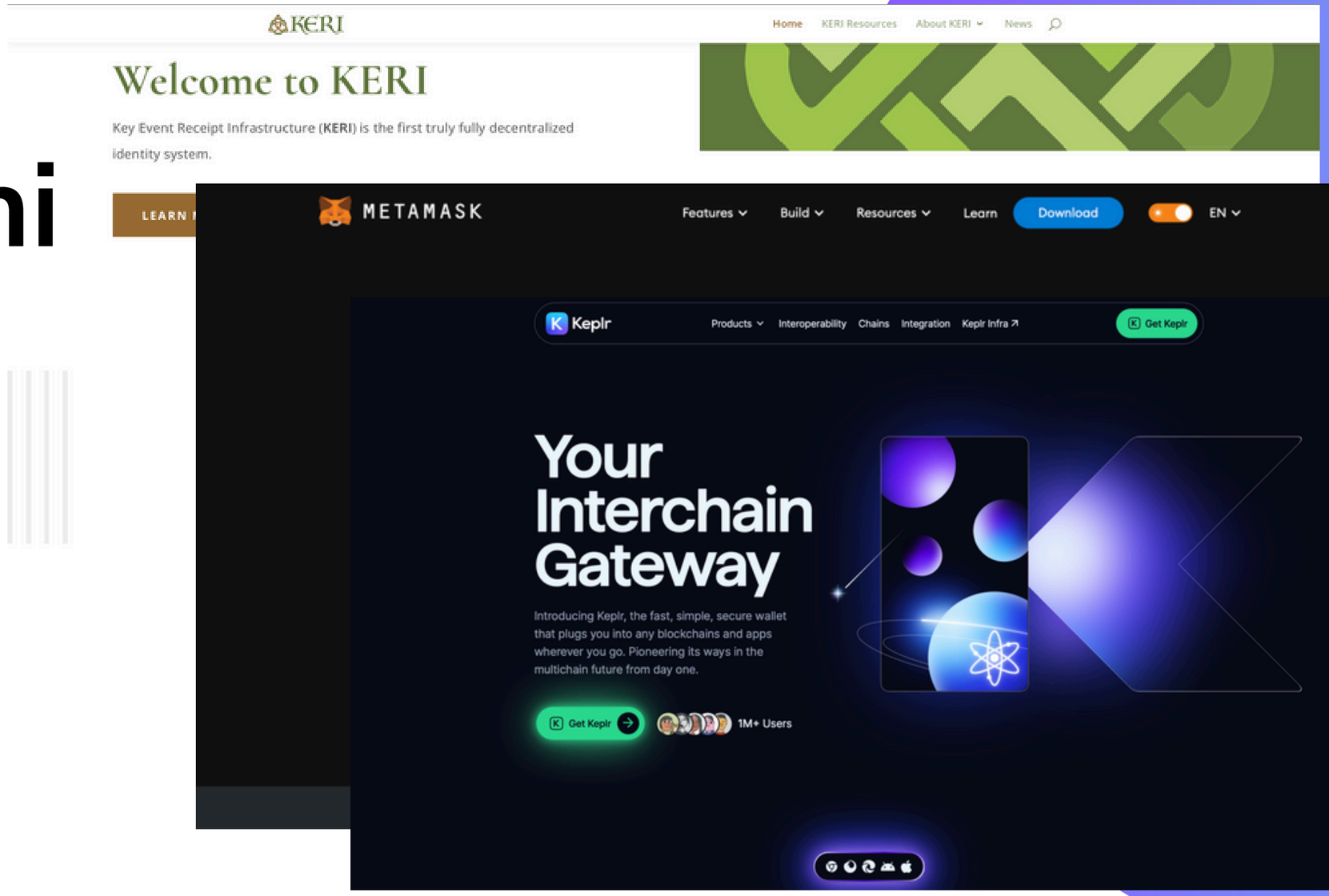
BrainStorming Iniziale & Conclusioni



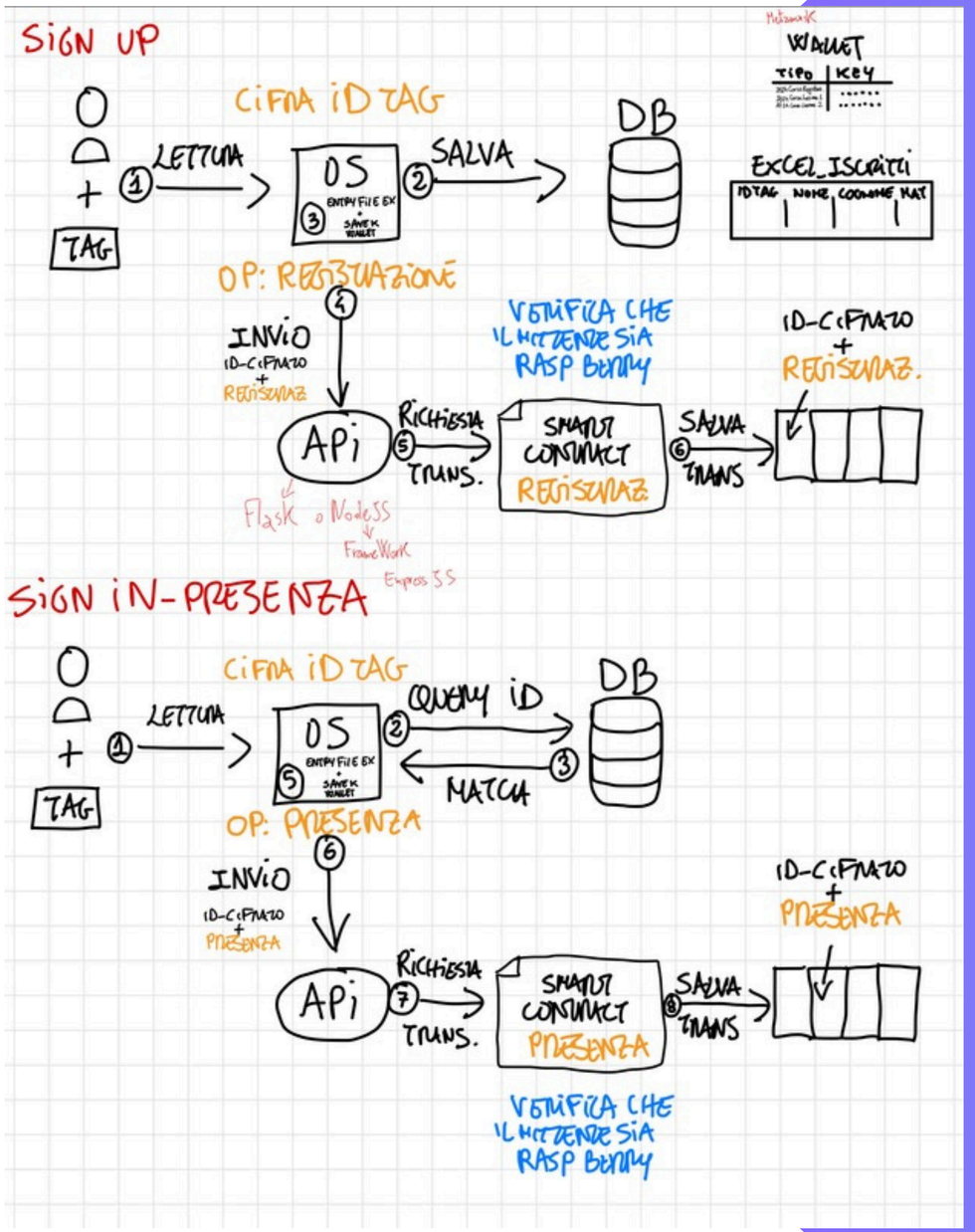
BrainStorming Iniziale & Conclusioni



BrainStorming Iniziale & Conclusioni



BrainStorming Iniziale & Conclusioni



Sviluppi Futuri

Sviluppi Futuri

Sviluppi Futuri

Eventuali Ottimizzazioni & Integrazioni

Chi è il mittente?

Eventuali Ottimizzazioni & Integrazioni

Chi è il mittente? E come lo identifico?

Eventuali Ottimizzazioni & Integrazioni

Aggiungere la firma del dispositivo alla transazione

Chi è il mittente? E come lo identifico?

Eventuali Ottimizzazioni & Integrazioni

BlackList / WhiteList / Access Control List

Aggiungere la firma del dispositivo alla transazione

Chi è il mittente? E come lo identifico?

Eventuali Ottimizzazioni & Integrazioni

BlackList / WhiteList / Access Control List

Verifica tramite IP o MAC Address

Aggiungere la firma del dispositivo alla transazione

Chi è il mittente? E come lo identifico?

Eventuali Ottimizzazioni & Integrazioni

BlackList / WhiteList / Access Control List

Aggiungere la firma del dispositivo alla transazione

Verifica tramite IP o MAC Address

Verifica tramite TLS/SSL con Certificato Client

Chi è il mittente? E come lo identifico?

Eventuali Ottimizzazioni & Integrazioni

BlackList / WhiteList / Access Control List

Aggiungere la firma del dispositivo alla transazione

Verifica tramite IP o MAC Address

Verifica tramite TLS/SSL con Certificato Client

Chi è il mittente? E come lo identifico?

Two-Factor Authentication (2FA) o Token OTP

Eventuali Ottimizzazioni & Integrazioni

Obiettivo...

Superare l’Obsolescenza.

MIGLIORARE L'ACCURATEZZA E L’EFFICIENZA DELLA REGISTRAZIONE DELLE PRESENZE,
SUPERANDO IL METODO CARTACEO ANTICO, GARANTENDO ALLO STESSO TEMPO
LA SICUREZZA DEI DATI E FACILITANDO L'ACCESSO E LA GESTIONE DELLE INFORMAZIONI.

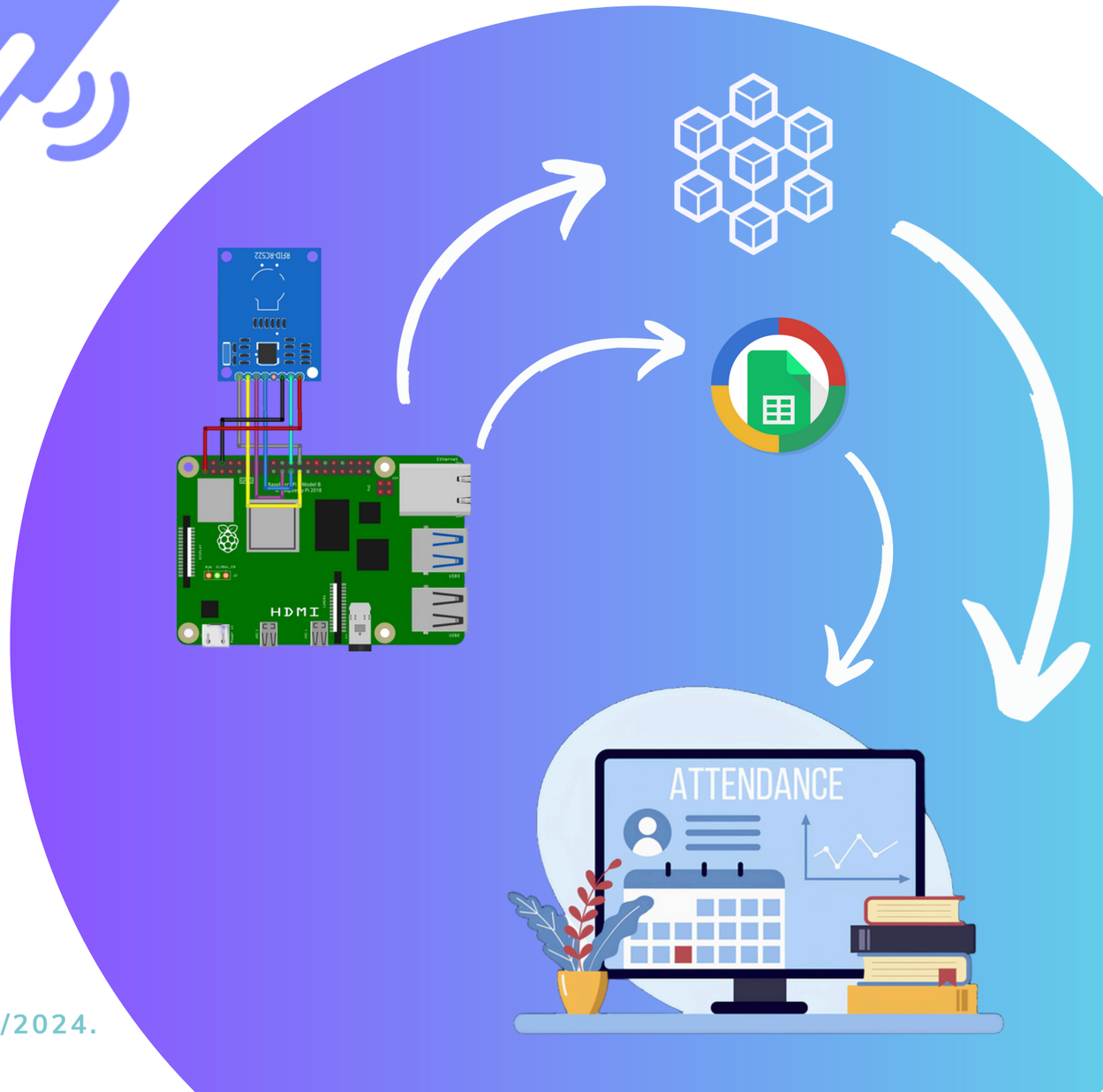
Obiettivo:

Superare l'obsolescenza.

MIGLIORARE L'ACCURATEZ ICIENZA DELLA REGISTRAZIONE DELLE PRESENZE,
SUPERANDO IL METOD EO ANTICO, GARANTENDO ALLO STESSO TEMPO
LA SICUREZZA DEI DATI E FA NDO L'ACCESSO E LA GESTIONE DELLE INFORMAZIONI.

CARLO SORRENTINO
MAT. 0522501821

Smart-Tag Attendance System



Grazie per la vostra attenzione!

PROGETTO COMBINATO PER I CORSI DI
SICUREZZA DEI DATI & IOT SECURITY

ANNO ACCADEMICO 2023/2024.