**Specifiche Esame per superamento del Corso Programmazione Avanzata in “unione” con CV&DL**

**A.A. 2022/2023**

Gruppo: SI (gruppo)

---------------------------------------------------------------------------

Si chiede di realizzare un back-end utilizzando i seguenti framework / librerie:

* Node.JS
* Express
* Sequelize
* RDBMS a scelta del gruppo (es. Postgres, MySQL, sqlite,…)
* Gestione delle code delle richieste mediante Bull ( <https://github.com/OptimalBits/bull> ) JS o mediante Celery in python (<https://docs.celeryq.dev/en/stable/getting-started/introduction.html> )
* Integrazione node + py (Esempio): <https://github.com/petranb2/Node_Python>
* Modello (SAM) : <https://www.kaggle.com/code/mrinalmathur/segment-anything-model-tutorial>

**Descrizione del progetto:**

Si realizzi un sistema che consenta di gestire l’inferenza a partire da immagini o video forniti dall’utente su un modello pre-addestrato. Devono essere predisposte le seguenti rotte:

* [U] Creazione di un data-set (fornire metadati minimi come nome ed una serie di tag sotto forma di lista di parole); all’inizio il data-set risulta vuoto.
* [U] Cancellazione (logica) di un data-set
* [U] Ottenere la lista dei data-set
* [U] Aggiornamento di un data-set (con verifica della non sovrapposizione con progetti dello stesso utente con lo stesso nome)
* [U] Inserimento di un contenuto all’interno del data-set
  + Caricamento di una immagine
  + Caricamento di un insieme di frame sotto forma di zip
  + Il costo associato ad ogni immagine è di 0.5 token; deve essere verificato se il credito disponibile è sufficiente a gestire la richiesta.
* [U] Effettuare un’inferenza su uno specifico data-set utilizzando SAM. Restituire l’id del processamento che consentirà di richiedere lo stato del processamento ed alla fine consentirà di ottenere il JSON contenente i dettagli dell’inferenza.
  + Ogni richiesta di inferenza ha un costo di 4 token / immagine. L’inferenza ha luogo se i crediti associati all’utente sono sufficienti. Annullare apriori il processamento se il credito non è sufficiente.
* [U] Creare una rotta che consenta di valutare lo stato di avanzamento del processamento distinguendo le fasi. Ad esempio, WAITING (in coda), PROGRESS (in fase di inferenza), ERROR (in caso di errore riportando anche la tipologia di errore), ABORTED (credito non sufficiente), COMPLETED (processamento data-set completo).
  + In caso di COMPLETED ritornare anche il risultato dell’inferenza sotto forma di JSON espresso come per ogni immagine il numero di oggetti segmentati e per ciascuno l’area espressa sotto forma di percentuale rispetto all’area totale;
* [U] Creare una rotta che consenta dato l’id del processamento, se COMPLETED, di scaricare una immagine o uno zip che contiene il risultato dell’inferenza (maschera di segmentazione).

L’inferenza deve avvenire mediante una gestione delle code con Bull o Celery che consenta di interfacciare il codice già disponibile per effettuare l’inferenza (codice python).

* [U] Restituire il credito residuo di un utente (necessaria autenticazione mediante token JWT)

Si chiede di sviluppare il codice possibilmente utilizzando typescript.

[U] corrisponde ad una rotta autenticata mediante JWT.

I dati di cui sopra devono essere memorizzati in un database esterno interfacciato con Sequelize. La scelta del DB è a discrezione degli studenti.

Le richieste devono essere validate.

Ogni utente autenticato (ovvero con JWT) ha un numero di token (valore iniziale impostato nel seed del database).

Nel caso di token terminati ogni richiesta da parte dello stesso utente deve restituire 401 Unauthorized.

Prevedere una rotta per l’utente con ruolo admin che consenta di effettuare la ricarica per un utente fornendo la mail ed il nuovo “credito” (sempre mediante JWT).

Il numero residuo di token deve essere memorizzato nel db sopra citato. Si deve prevedere degli script di seed per inizializzare il sistema.

Si chiede di utilizzare le funzionalità di middleware.

Si chiede di gestire eventuali errori mediante gli strati middleware sollevando le opportune eccezioni.

Si chiede di commentare opportunamente il codice.

**Note:**

Nello sviluppo del progetto è richiesto l’utilizzo di Design Pattern che dovranno essere documentati opportunamente nel Readme.MD.

I token JWT da usare possono essere generati usando gli script mostrati a lezione.

La chiave privata da usare lato back-end deve essere memorizzata un file .env e caricata mediante la libreria

**Specifiche Repository**

* Il codice deve essere reso disponibile su piattaforma github con repo pubblico
* Nel repository è obbligatario inserire un Readme.md che descriva:
  + Obiettivo del progetto
  + Progettazione
    - diagrammi UML
    - descrizione dei pattern usati motivandone la scelta
  + Come avviare il progetto mediante docker-compose (preferibile) per comporre i servizi richiesti.
  + Test del progetto mediante chiamate effettuate con curl o wget o con Postman
* Il Readme.MD può essere redatto in lingua italiana o inglese (non vi saranno differenziazioni nel processo di valutazione)

**Specifiche Consegna**

* La consegna avviene esclusivamente mediante moodle all’indirizzo di seguito riportato dove dovranno essere indicati:
  + URL del repository pubblico
  + Commit id che verrà usata dal docente per effettuare la valutazione.
  + Data per lo svolgimento dell’esame
* Indirizzo per la consegna: <https://learn.univpm.it/mod/assign/view.php?id=431110>

Buon lavoro 😉

Il docente

Adriano Mancini