Specifiche dei progetti per l'esame di *Applicazioni dinamiche per il Web* per l'anno accademico 2022/2023

Aggiornato al 09 maggio 2023

1 Progetto per gli appelli di giugno e luglio 2023 (versione 1.3)

Considerate la base di dati disponibile all'indirizzo
`https://www.postgresqltutorial.com/postgresql-getting-started/postgresql-sample-database/` che contiene tabelle e dati di esempio per la gestione di un noleggio di DVD in diversi punti vendita (`store`).

Si realizzi un'applicazione web dinamica per il cliente in cui un cliente, una volta autenticato, può:

1. Vedere tutti i DVD disponibili e poter noleggiare uno o più tra quelli disponibili (non si chiede di gestire la prenotazione)
   - Per ogni film, deve poter vedere subito in elenco *titolo, anno, rating, genere, lingua, costo* (e altri campi che si ritengono utili)
   - L'elenco dei film deve prevedere il filtraggio per categoria e ricerca per nome.
   - Cliccando sul titolo di un film, si deve aprire una layer (sopra la pagina corrente o creando uno spazio nella pagina corrente) che mostra altri dati di interesse (come *attori*, durata massima noleggio, lunghezza film, etc) e **dove il DVD è disponibile** (vedere relazioni `inventory, rental`).
     Circa il costo del noleggio, `rental_rate` indica il costo in $ per un periodo pari o inferiore a `rental_duration`.
     Da una verifica dei dati presenti nella base di dati, ogni store ha da 0 a 4 copie di ciascun film e una copia **non** è disponibile se la `return_date` è NULL.
   - Sia nell'elenco, sia nella pagina di dettaglio, ci deve essere un bottone per chiedere il noleggio del DVD se disponibile (limitiamo i casi). Nell'elenco il bottone deve essere su ciascuna linea.
   - Il layer di noleggio deve permettere di specificare la data inizio noleggio (corrente o entro i prossimi 2 giorni). Il layer deve mostrare il negozio (con indirizzo) dove si trova il DVD richiesto. Deve, infine, dare conferma che il noleggio è accordato.
   - Non si deve gestire il pagamento.
2. Vedere i noleggi fatti in passato con il calcolo di quanto speso. Nella tabella `payment`, il campo `amount` indica quanto pagato per il noleggio. La durata dello stesso si ricava da `return_date` e `rental_date`.
   - Anche qui, un elenco con possibilità di avere i dettagli cliccando su un noleggio.

Non si deve gestire la restituzione di un DVD.

Circa gli utenti, l'applicazione deve, in una base di dati separata, gestire gli account dei possibili utenti che accedono alla applicazione mediante autenticazione per garantire il principio che la base di dati di dominio deve essere separata dalla base di dati di funzionamento dell'applicazione. Una login dell'applicazione di un cliente deve avere il campo `customer_id` che accoppia la login al cliente nella tabella `customer`.

Il cliente deve poter vedere la lista dei suoi noleggi ordinabile al volo secondo le diverse colonne (questo per vedere come viene implementato il fatto di avere viste diverse con gli stessi dati).

In questo appello si richiede di sviluppare l'applicazione in modalità SPA (2 tiers) usando solo Express.js lato server e Angular.js lato client.

Ulteriori richieste di progetto:

1. Il gruppo di sviluppo **deve** essere di 2-3 studenti in cui le responsabilità di progetto devono essere ben definite.
2. Il DBMS deve essere PostregSQL $\geq$ 14. A tal riguardo, una volta scaricato l'archivio e seguite le istruzioni circa il `$$PATH$$` nel file `restore.sql`, sempre nello stesso file cancellate le direttive `LC_COLLATE = 'English_United States.1252' LC_CTYPE = 'English_United States.1252';` prima di fare il restore. Il restore avviene senza warning se si usa PostregSQL $\geq$ 14.
3. Il lato server deve usare GraphQL per realizzare i microservizi di aggiornamento base di dati.

4. Il lato client deve essere Angular (ultima versione in `https://angular.io`)
5. L'applicazione deve essere accessibile anche via tablet o smartphone (usare CSS per rendere responsive).
6. Garantire il livello AA WCAG 2.1 solo per le pagine degli elenchi (sufficiente per valutare come viene fatto).

Ulteriori informazioni

Se sono necessarie ulteriori specifiche, **non** mandate una email, ma scrivete le forum del sito di elearning del corso. In questo modo le informazioni sono condivise tra tutti risparmiando tempo mio e vostro. Sono ammesso anche scambi di idee durante lo sviluppo.

Modalità di presentazione all'esame

- Preparare massimo 2-3 slide in cui si spiegano i criteri/principi decisi per lo sviluppo dell'applicazione.
- Preparare massimo 2 slide in cui si mostra i servizi resi disponibili secondo il modello GraphQL.
- Preparare 2 in cui si mostra il processo di validazione AA WCAG 2.1 (che strumenti si sono scelti, che prove si sono fatte). Queste prove si devono poi mostrare dal vivo.
- È necessario saper dimostrare di essere in grado di presentare/modificare qualsiasi riga del codice sorgente a richiesta del docente.
- È possibile che venga richiesto di aggiungere una funzionalità al volo (di piccola entità).

Ricordo, infine, le raccomandazione fatte alla presentazione del corso:

- Il progetto deve funzionare esattamente come da specifiche
- Il progetto deve essere sviluppato su un proprio PC (per chi non ha PC, ci sono i PC dei laboratori)
- Il progetto deve venire presentato da tutto il gruppo insieme.
- Non sono ammesse tecnologie che richiedano compilazioni o procedure complesse di aggiornamento del codice eseguito. Deve essere possibile aprire un file sul server e modificarlo al volo durante la prova.