

# Simulazione della prova pratica d'esame – S2

## Gestione iscrizioni e check-in per eventi di formazione aziendale

### Obiettivo della prova

La prova è finalizzata a valutare la capacità dell'esaminando di progettare e sviluppare un'applicazione web full stack con autenticazione, gestione di dati relazionali, interazione tra frontend e backend tramite API REST.

### Note per la valutazione

- Corretta implementazione dell'**autenticazione**
- Corretta gestione dei **casi d'uso**
- Uso corretto delle **API REST**
- Interfaccia utente **chiara e funzionale**
- **Organizzazione del codice** e aderenza alle **best practices**
- Il **deployment**, se effettuato, contribuisce all'attribuzione di punteggio aggiuntivo

### Requisiti generali

- L'applicazione deve essere sviluppata con **separazione tra backend e frontend**
- L'applicazione deve essere realizzata con **tecnologia client-side rendering (CSR)**
- L'esaminando deve preparare uno strumento per il **test dell'API** esposta dal backend (Swagger, Postman, Insomnia o altro)
- L'applicazione dev'essere **inizializzata con un set di dati realistici**, sufficienti per testare tutti i casi d'uso previsti
- Il codice deve essere **strutturato e documentato**

### Architettura dell'applicazione

#### Backend

- Un **database** (a scelta del candidato)
- Un **servizio** che espone un'**API REST** consumata dal frontend
- Le tecnologie per il backend (linguaggi, framework) sono a scelta del candidato

#### Frontend

- Il framework per il frontend è a scelta del candidato

### Descrizione del caso d'uso

I dipendenti di un'azienda possono iscriversi a eventi di formazione interna. Ogni evento ha una lista di iscritti. All'arrivo dei partecipanti, un organizzatore registra il loro check-in.

## Requisiti della UI

- L'interfaccia deve essere **chiara, responsive e coerente nello stile**
- Le pagine devono essere **accessibili da un menu**
- Tutte le pagine devono essere **facilmente navigabili**

## Pagine da sviluppare

- Login / Registrazione
- Dashboard dipendente con:
  - Lista eventi disponibili
  - Iscrizione ad un evento
  - Lista iscrizioni personali
  - Annullamento dell'iscrizione ad un evento
- Dashboard organizzatore con:
  - Lista degli eventi, con inserimento, modifica ed eliminazione
  - Pannello per gestire i check-in ad un evento

## Regole di sicurezza

Solo gli organizzatori possono creare nuovi eventi, modificare eventi esistenti ed eliminare eventi

Solo gli organizzatori possono effettuare i check-in.

Un dipendente può vedere solo le proprie iscrizioni personali

Un dipendente può annullare solo una propria iscrizione

Un dipendente può iscriversi ad un evento fino al giorno prima della data dell'evento

Un dipendente può annullare l'iscrizione ad un evento fino al giorno prima della data dell'evento

## Requisito avanzato

Implementare una pagina che consente di visualizzare la lista degli eventi già avvenuti, con l'indicazione, per ciascun evento, del numero di check-in effettuati e della percentuale di partecipazione rispetto agli iscritti. La visualizzazione di queste statistiche può essere implementata in forma tabellare e/o in forma grafica. Implementare un filtro per data (dal - al). Questa pagina deve essere visualizzabile solo dagli organizzatori.

## Chiamate API da sviluppare

Alcune API prevedono l'autenticazione, altre sono aperte.

Per ogni metodo, prevedere lato server la validazione dei dati ricevuti.

Le regole di sicurezza possono essere descritte nella sezione dedicata oppure all'interno della descrizione delle singole API.

## API di registrazione e autenticazione

- POST /api/utenti/register Registrazione nuovo utente (Nome, Cognome, Email, Password)
- POST /api/utenti/login Autenticazione utente (Email, Password; viene restituito un token)

## API con autenticazione

- CRUD eventi (richiamabile solo dagli organizzatori)
- Elenco eventi (visibile da tutti i dipendenti)
- Iscrizione ad un evento (richiamabile da tutti i dipendenti)
- Disiscrizione da un evento (richiamabile da tutti i dipendenti)
- Registrazione del check-in (richiamabile solo dagli organizzatori)

#### API senza autenticazione:

Nessuna

### Modello dei dati

Le informazioni che si vogliono mantenere sono le seguenti:

- **Utente**
  - UtenteID (PK)
  - Nome
  - Cognome
  - Email
  - Password
  - Ruolo (Dipendente / Organizzatore)
- **Evento**
  - EventoID (PK)
  - Titolo
  - Data
  - Descrizione
- **Iscrizione**
  - IscrizioneID (PK)
  - UtenteID (FK)
  - EventoID (FK)
  - CheckinEffettuato (boolean)
  - OraCheckin

Il data model proposto è indicativo. Il candidato può integrarlo secondo necessità.

### Deployment

L'applicazione va pubblicata su un server a propria scelta. L'URL di accesso deve contenere **nome e cognome dell'allievo**.

Esempio: `https://app-nome-cognome.nomehost.dominio`