****

**Sviluppo di un sito web per la gestione di corsi accademici**

**Introduzione**

**Il progetto si propone di sviluppare un sito web per la gestione dei corsi accademici presso la facoltà di Informatica di Unicam. Questa relazione fornirà una panoramica dettagliata del progetto, comprese le tecnologie utilizzate, i modelli dei dati, le funzionalità principali, la sicurezza implementata e le possibilità di sviluppo futuro.**

**Tecnologie Utilizzate**

**L'applicazione è stata sviluppata utilizzando le seguenti tecnologie:**

* **Django (Backend): Django è stato utilizzato per creare il lato server del sito. Offre una robusta gestione delle API, un sistema di autenticazione integrato e facilita la creazione di modelli dati complessi.**
* **Django REST framework: Questa libreria è stata utilizzata per creare API RESTful nel backend. Semplifica la creazione, la serializzazione e l'autenticazione delle risorse API.**
* **Angular (Frontend): Angular è stato utilizzato per creare il lato client del sito. Fornisce un'architettura modulare e gestisce efficacemente la vista per creare interfacce utente dinamiche.**
* **Angular Calendar: Questa libreria è stata utilizzata per visualizzare i calendari. Offre funzionalità avanzate per la visualizzazione e la gestione degli eventi.**

**BACKEND (DJANGO)**

**Modelli dei Dati**

**Il backend dell'applicazione utilizza i seguenti modelli di dati per rappresentare gli oggetti chiave del sistema:**

1. **User (Utente): Il modello utente di base di Django è stato esteso per includere informazioni aggiuntive come nome, cognome, anno accademico e altri dati personali.**
2. **Corso: Questo modello rappresenta un corso accademico ed è collegato a un docente tramite una chiave esterna. Contiene informazioni come il nome del corso, il semestre, l'anno accademico e i crediti.**
3. **Docente: Questo modello rappresenta un docente e contiene informazioni come il nome, il cognome e un link personale.**
4. **Lezione: Questo modello rappresenta una lezione e contiene informazioni come il giorno della settimana, l'orario di inizio e fine, il corso associato e l'aula.**

**API RESTful**

**Il backend fornisce API RESTful per consentire agli utenti di accedere a risorse come corsi, docenti e lezioni. Le API sono state create utilizzando Django REST framework, semplificando la gestione delle richieste HTTP e delle risposte JSON.**

**Sicurezza**

**Diverse misure di sicurezza sono state implementate nel progetto:**

* **Autenticazione: L'autenticazione degli utenti avviene utilizzando token JWT (JSON Web Token). Questo sistema di autenticazione è robusto e offre protezione contro attacchi come l'intercettazione di sessioni.**
* **Gestione delle Autorizzazioni: Sono state definite diverse classi di autorizzazione, inclusa l'autorizzazione di base degli utenti autenticati e l'autorizzazione degli amministratori. Queste autorizzazioni determinano chi può accedere alle diverse risorse API.**
* **Protezione CSRF: Django offre una protezione CSRF integrata per prevenire attacchi di falsificazione di richieste tra siti (CSRF).**
* **Validazione dei Dati: I dati inviati attraverso le richieste API vengono attentamente validati e sanificati per evitare inserimenti malevoli o dannosi nel database.**
* **Limitazione dei Tentativi di Accesso: Sono state implementate limitazioni sui tentativi di accesso errati per proteggere gli account utente da attacchi di forza bruta.**

**FRONTEND (ANGULAR)**

**Componenti Principali**

1. **CorsiComponent: Questo componente visualizza una lista di corsi e offre la possibilità di filtrarli in base all'anno accademico. Gli utenti possono fare clic su un corso per ottenere ulteriori dettagli.**
2. **SchedulerComponent: Questo componente consente agli utenti di visualizzare un calendario delle lezioni. Gli eventi nel calendario sono associati ai corsi, ai docenti e alle aule. Gli utenti possono filtrare le lezioni per anno accademico e visualizzare ulteriori dettagli sugli eventi.**

**Servizi Principali**

1. **AuthService: Questo servizio gestisce l'autenticazione degli utenti. Gestisce il processo di accesso, registrazione e logout degli utenti.**
2. **ApiService: Questo servizio effettua richieste HTTP per ottenere dati dal backend. Gestisce la comunicazione tra il frontend e il backend.**

**Funzionalità Principali**

**Il sito offre diverse funzionalità chiave:**

* **Autenticazione Utente: Gli utenti possono registrarsi e accedere all'applicazione utilizzando un sistema di autenticazione basato su token JWT (JSON Web Token).**
* **Visualizzazione Corsi: Gli utenti possono visualizzare una lista completa di corsi accademici.**
* **Visualizzazione Calendario delle Lezioni: Gli utenti possono visualizzare un calendario delle lezioni accademiche. Possono filtrare le lezioni in base all'anno accademico e ottenere ulteriori dettagli sugli eventi.**
* **Filtro per Anno Accademico: Gli utenti possono filtrare la lista dei corsi e dei docenti e il calendario delle lezioni in base all'anno accademico desiderato.**
* **Logout: Gli utenti possono effettuare il logout dall'applicazione, invalidando il token JWT.**

**Possibili Sviluppi Futuri**

**Il progetto offre opportunità di sviluppo futuro, tra cui:**

* **Migliorare l'interfaccia utente con funzionalità aggiuntive e un design più accattivante.**
* **Integrazione con servizi di terze parti come un sistema di pagamento per le tasse accademiche.**