



UNIVERSITÀ DI TRENTO

Dipartimento di Ingegneria e Scienza dell'Informazione

Corso di Laurea in
Informatica

ELABORATO FINALE

TITOLO

Sottotitolo (alcune volte lungo - opzionale)

Supervisore
Prof. Paolo Giorgini
Prof. Alessandro Tomasi

Laureando
Stefano Dal Mas

Anno accademico 2022/2023

Ringraziamenti

OpenAI

Indice

Sommario	2
1 Obbiettivi del progetto	2
1.1 DISI Industry	2
2 Tecnologie utilizzate	3
2.1 Django	3
2.1.1 Architettura MVT	3
2.1.2 Sicurezza	4
3 Conclusioni	4
Bibliografia	4
A Titolo primo allegato	6
A.1 Titolo	6
A.1.1 Sottotitolo	6
B Titolo secondo allegato	7
B.1 Titolo	7
B.1.1 Sottotitolo	7

Sommario

Sommario è un breve riassunto del lavoro svolto dove si descrive l'obiettivo, l'oggetto della tesi, le metodologie e le tecniche usate, i dati elaborati e la spiegazione delle conclusioni alle quali siete arrivati.

Il sommario dell'elaborato consiste al massimo di 3 pagine e deve contenere le seguenti informazioni:

- contesto e motivazioni
- breve riassunto del problema affrontato
- tecniche utilizzate e/o sviluppate
- risultati raggiunti, sottolineando il contributo personale del laureando/a

1 Obiettivi del progetto

Il progetto ha come obiettivo la realizzazione di una WebApp sottoforma di modulo da poter aggiungere alla piattaforma DISI Industry. L'applicativo deve permettere alle aziende, enti esterni all'università, di poter formalizzare e proporre delle "Challenge". Esse possono essere definite come gare a lungo termine ove viene richiesto ai gruppi partecipanti di effettuare sia la fase di progettazione che eventualmente la fase implementativa di una soluzione rispetto ad un problema definito, oppure può essere proposta sottoforma di hackaton, ossia una competizione volte alla risoluzione di un problema in un lasso di tempo limitato mettendo in palio un premio per i team vincitori, come nel caso delle Reply Challenge, hackaton nelle quali ai gruppi viene richiesto di risolvere dei problemi di algoritmica e di programmazione.

Questo progetto nasce dalla mia partecipazione al Samsung Innovation Camp, un evento proposto da Samsung in collaborazione con Randstad, nel quale veniva richiesto agli studenti di risolvere una problematica proposta dall'azienda, differenziandola sulla base dell'ateneo scelto, proponendo soluzioni che sfruttassero le nuove tecnologie quali Internet of Things, Intelligenza Artificiale e tutte le nuove tecnologie correlate. La problematica proposta all'ateneo di Trento era inerente alla valorizzazione del patrimonio naturalistico, artistico e culturale mediante l'innovazione digitale, con l'obiettivo di sostenere il settore della Cultura e del Turismo.

La partecipazione a tale progetto mi ha permesso di conoscere nuove tecnologie, meccaniche aziendali e di lavorare in un team di sviluppo, permettendomi di migliorare le mie capacità di problem solving, di lavoro in team e di gestione del tempo, e mi ha fatto realizzare quanto sia importante avere una piattaforma nella quale gli studenti possano in modo semplice e veloce trovare tutte le informazioni necessarie per partecipare alle Challenge proposte dalle aziende.

1.1 DISI Industry

La scelta dell'integrazione di questo modulo all'interno di una WebApp già esistente è sorta dal fatto che DISI industry nasce già con l'idea di connettere gli studenti con l'industria, in quanto i primi possono cercare delle offerte di lavoro comparabili con le loro competenze apprese durante gli studi, mentre le compagnie possono offrire ruoli di lavoro per trovare il personale più adatto alle loro richieste. Essa può essere dunque definita come una piattaforma di recruiting online per gli studenti del DISI.

[1]

Mi è dunque venuta naturale l'idea di implementare il modulo all'interno della WebApp per l'affinità intrinseca tra la mia ideologia e quella proposta dalla piattaforma già esistente, così da permettere agli studenti non solo di poter cercare potenziali offerte di tirocinio o di lavoro, ma anche di poter partecipare a delle Challenge proposte dalle aziende, così da poter mettere in pratica le proprie conoscenze e competenze e di poterle migliorare, oltre che di poter vincere dei premi, il tutto mediante la stessa piattaforma per ridurre al minimo la fatica dovuta alla navigazione di molteplici pagine web e portali.

2 Tecnologie utilizzate

In questo capitolo vengono presentate le tecnologie utilizzate e le motivazioni dietro tali scelte. In particolare, vengono presentate le tecnologie utilizzate per la realizzazione del back-end, del front-end e del database.

2.1 Django

Il framework utilizzato è Django, esso è stato scelto per la sua semplicità di utilizzo e per la sua flessibilità. Django è un framework open-source di alto-livello per lo sviluppo di applicazioni web scritto in Python. Esso è basato sul pattern architetturale Model-View-Template (MVT) e permette di sviluppare applicazioni web in modo rapido e pulito, fornendo un'interfaccia di amministrazione per la gestione dei dati, così da non dover implementare un admin zone mediante un altro linguaggio, ma mantenendo unificata la codebase.

Un altro vantaggio risiede nella scalabilità del framework, il quale permette la creazione di moduli che possono essere inseriti o rimossi con l'aggiunta di poche righe di codice all'interno di un file apposito.

2.1.1 Architettura MVT

Il vantaggio dell'utilizzo di un framework basato su architettura MVT permette ad ogni livello, ossia Model, View e Template, di lavorare indipendentemente, il che permette una maggiore indipendenza tra le parti. Esso è conveniente nel caso si decida di scalare l'applicativo web, in quanto permette di aggiungere nuove funzionalità senza dover modificare il codice già esistente. Inoltre, un'architettura MVT permette di rendere più veloce e più semplice la trasmissione di dati via internet.

Vengono presentati in dettaglio le componenti dell'architettura MVT:

- **Model** : questo componente aiuta la gestione del database ed esso è un livello di accesso dati che contiene i campi ed il comportamento dei dati che stanno venendo gestiti. Esso utilizza un pattern ORM (Object-Relational Mapping) che mappa i campi del database con gli attributi della classe Python. Questo permette di utilizzare un linguaggio di programmazione orientato agli oggetti come Python per la gestione dei dati, permette un livello di astrazione tale da non rendere obbligatoria la stesura di query in linguaggio SQL, inoltre permette una maggiore semplicità durante la manutenzione del codice dovuta al fatto che solamente un linguaggio viene utilizzato. Mediante ORM è anche possibile cambiare database in modo efficace ed efficiente, in quanto non è necessario modificare il codice già esistente. I modelli aiutano ad effettuare le operazioni CRUD (Create, Read, Update, Delete) sul database in modo semplice e veloce. Inoltre, Django permette di definire delle relazioni tra i modelli, in modo da poter accedere ai dati in maniera semplice e veloce, oltre alla definizione di ciò che viene definita Business Logic del programma (es. validazione dei dati, ecc.). [2]
- **View** : Esso può essere visto come l'organoesecutivo della Business Logic, in quanto permette di interagire con i modelli definiti dal modulo Model e gestisce la renderizzazione in template. La view ottiene i dati dal Model, dopodiché può modificare i suddetti ed eventualmente renderizzarli in un template. Esso accetta richieste HTTP(HyperText Transfer Protocol), applica la business logic e fornisce risposte HTTP al client. [5]

- **Template** : Questo modulo è il livello più vicino all'utente, in quanto è un livello di presentazione che gestisce completamente l'interfaccia dell'utente. Essi sono file con codice HTML(HyperText Markup Language) che vengono utilizzati per renderizzare dati ed essi possono contenere dati statici o dinamici. Nel caso di Django viene utilizzato un linguaggio speciale chiamato Django Template, il quale permette la definizione di variabili, cicli, condizioni, ecc. oltre ai comandi classici di HTML. Inoltre, permette di definire un template base, il quale può essere esteso da altri template. [4]

2.1.2 Sicurezza

Un altro vantaggio del framework Django è la sicurezza, in quanto nativamente supporta dei controlli che possono essere facilmente inclusi sfruttando il livello d'astrazione fornito da un framework. [3]

Oltre ai controlli sull'autenticazione, autorizzazione e gestione delle sessioni, tale framework permette di evitare attacchi tra i quali:

- **SQL Injection** : Attacco volto all'inserimento di codice SQL in input non controllati, in modo da poter accedere ai dati del database. Django permette di evitare tale attacco in quanto permette di utilizzare un linguaggio di programmazione orientato agli oggetti come Python per la gestione dei dati, applicando ciò che viene definito *input sanitizing*, il quale permette di evitare l'inserimento di codice SQL in input non controllati.
- **Cross Site Scripting(XSS)** : Tale attacco permette all'utente di inserire codice HTML e JavaScript malevolo che verrà poi visualizzato dagli altri utenti dell'applicazione mediante la pressione di un link o bottone. Tale problema viene risolto mediante input sanitizing e l'escape automatico di caratteri non sicuri.
- **Cross Site Request Forgery(CSRF)**: L'attacco CSRF sfrutta il fatto che molte applicazioni web autenticano gli utenti utilizzando sessioni o token di autenticazione, ma non validano in modo appropriato la fonte delle richieste che vengono inviate. In un attacco CSRF, un malintenzionato crea una pagina web malevola o un messaggio e lo fa visualizzare all'utente target mentre è autenticato in un'altra applicazione web.

Quando l'utente visualizza questa pagina o questo messaggio, il codice maligno presente al suo interno genera una richiesta HTTP verso l'applicazione web bersaglio, sfruttando la sessione o il token di autenticazione dell'utente. Poiché l'applicazione web bersaglio considera la richiesta legittima, esegue l'azione richiesta, che potrebbe essere, ad esempio, un trasferimento di fondi, una modifica delle impostazioni dell'account o l'invio di un messaggio. Django permette di evitare tale attacco mediante l'utilizzo di un token di sicurezza.

- **Clickjacking** : Attacco che mira ad ingannare l'utente mediante il click di zone nascoste o trasparenti. Django permette di evitare tale attacco mediante l'utilizzo di un middleware che permette di inserire un'intestazione HTTP.

3 Conclusioni

Conclusioni

Bibliografia

- [1] Disi industry. <https://industry.disi.unitn.it>.
- [2] Django models. <https://docs.djangoproject.com/en/4.2/topics/db/models/>.
- [3] Django security. <https://docs.djangoproject.com/en/4.2/topics/security/>.
- [4] Django template. <https://docs.djangoproject.com/en/4.2/topics/templates/>.
- [5] Django views. <https://docs.djangoproject.com/en/4.2/topics/class-based-views/>.

Allegato A Titolo primo allegato

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec sed nunc orci. Aliquam nec nisl vitae sapien pulvinar dictum quis non urna. Suspendisse at dui a erat aliquam vestibulum. Quisque ultrices pellentesque pellentesque. Pellentesque egestas quam sed blandit tempus. Sed congue nec risus posuere euismod. Maecenas ut lacus id mauris sagittis egestas a eu dui. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Pellentesque at ultrices tellus. Ut eu purus eget sem iaculis ultricies sed non lorem. Curabitur gravida dui eget ex vestibulum venenatis. Phasellus gravida tellus velit, non eleifend justo lobortis eget.

A.1 Titolo

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec sed nunc orci. Aliquam nec nisl vitae sapien pulvinar dictum quis non urna. Suspendisse at dui a erat aliquam vestibulum. Quisque ultrices pellentesque pellentesque. Pellentesque egestas quam sed blandit tempus. Sed congue nec risus posuere euismod. Maecenas ut lacus id mauris sagittis egestas a eu dui. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Pellentesque at ultrices tellus. Ut eu purus eget sem iaculis ultricies sed non lorem. Curabitur gravida dui eget ex vestibulum venenatis. Phasellus gravida tellus velit, non eleifend justo lobortis eget.

A.1.1 Sottotitolo

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec sed nunc orci. Aliquam nec nisl vitae sapien pulvinar dictum quis non urna. Suspendisse at dui a erat aliquam vestibulum. Quisque ultrices pellentesque pellentesque. Pellentesque egestas quam sed blandit tempus. Sed congue nec risus posuere euismod. Maecenas ut lacus id mauris sagittis egestas a eu dui. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Pellentesque at ultrices tellus. Ut eu purus eget sem iaculis ultricies sed non lorem. Curabitur gravida dui eget ex vestibulum venenatis. Phasellus gravida tellus velit, non eleifend justo lobortis eget.

Allegato B Titolo secondo allegato

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec sed nunc orci. Aliquam nec nisl vitae sapien pulvinar dictum quis non urna. Suspendisse at dui a erat aliquam vestibulum. Quisque ultrices pellentesque pellentesque. Pellentesque egestas quam sed blandit tempus. Sed congue nec risus posuere euismod. Maecenas ut lacus id mauris sagittis egestas a eu dui. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Pellentesque at ultrices tellus. Ut eu purus eget sem iaculis ultricies sed non lorem. Curabitur gravida dui eget ex vestibulum venenatis. Phasellus gravida tellus velit, non eleifend justo lobortis eget.

B.1 Titolo

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec sed nunc orci. Aliquam nec nisl vitae sapien pulvinar dictum quis non urna. Suspendisse at dui a erat aliquam vestibulum. Quisque ultrices pellentesque pellentesque. Pellentesque egestas quam sed blandit tempus. Sed congue nec risus posuere euismod. Maecenas ut lacus id mauris sagittis egestas a eu dui. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Pellentesque at ultrices tellus. Ut eu purus eget sem iaculis ultricies sed non lorem. Curabitur gravida dui eget ex vestibulum venenatis. Phasellus gravida tellus velit, non eleifend justo lobortis eget.

B.1.1 Sottotitolo

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec sed nunc orci. Aliquam nec nisl vitae sapien pulvinar dictum quis non urna. Suspendisse at dui a erat aliquam vestibulum. Quisque ultrices pellentesque pellentesque. Pellentesque egestas quam sed blandit tempus. Sed congue nec risus posuere euismod. Maecenas ut lacus id mauris sagittis egestas a eu dui. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Pellentesque at ultrices tellus. Ut eu purus eget sem iaculis ultricies sed non lorem. Curabitur gravida dui eget ex vestibulum venenatis. Phasellus gravida tellus velit, non eleifend justo lobortis eget.