

Università Politecnica delle Marche



Facoltà di Ingegneria
Corso di Laurea in Ingegneria Informatica e dell'Automazione

**Sviluppo del modulo di accesso ai contenuti in una
piattaforma di e-learning**
**Development of the contents access module in an e-
learning platform**

Relatore:
Prof. Alessandro Cucchiarelli

Candidato:
Lorenzo Di Alessandro

Anno accademico 2022/2023

INDICE

1 - Introduzione

2 - Contesto applicativo

3 - Obiettivi del progetto

4 - Strumenti utilizzati

5 - Applicazione sviluppata

6 - Conclusioni e Sviluppi futuri

Bibliografia

Capitolo 1

Introduzione

Il presente lavoro di tesi illustra lo sviluppo di una sezione di accesso ai contenuti di una piattaforma e-learning, svolto durante l'attività di tirocinio presso la Partner Revolution. Quest'ultima è una società che si occupa di promuovere partnership tra diverse aziende attraverso l'uso di tecnologie e strumenti web.

Prima di affrontare lo sviluppo pratico della sezione di accesso ai contenuti di un e-learning, è opportuno studiarne il contesto applicativo. Con il termine e-learning si intende apprendimento online, ovvero la divulgazione di contenuti formativi sfruttando le tecnologie web. Le piattaforme e-learning che vantano maggior fama presentano caratteristiche simili. Tutte hanno una dashboard personale, un catalogo

dei corsi offerti e uno di quelli attivi, con un motore di ricerca avanzato compreso di filtri per facilitare la ricerca. Inoltre, dispongono di contenuti interattivi, come quiz, esercizi e progetti, per monitorare le attività dell'utente.

Le persone con un dispositivo dotato di accesso alla rete Internet possono usufruire di tutti i servizi di una sezione di accesso ai contenuti di una piattaforma e-learning. Durante lo sviluppo di quest'ultima si deve tener conto degli ostacoli da affrontare per renderla adattabile ad ogni necessità. In particolare, è necessario garantire un adeguato coinvolgimento degli studenti, la personalizzazione dell'apprendimento di ogni singolo utente, la valutazione delle competenze, la gestione dei dati e dei costi per il mantenimento e l'aggiornamento della piattaforma.

La richiesta della società PartnerRevolution è stata quella di sviluppare, sfruttando il framework Larvel 10, la sezione utente, di tipo studente, dell'interfaccia e del back-end per l'accesso ai contenuti di una piattaforma e-learning. A termine dello sviluppo, il software prodotto può essere ceduto o venduto ad altre aziende, le quali integrano la sezione di accesso al proprio sito web per divulgare contenuti formativi ai loro dipendenti. L'interfaccia utente di tale sezione è strutturata in due viste, quali: "Catalogo dei corsi" e "Lezioni". La prima offre all'utente la visualizzazione dei corsi disponibili o che sono in fase di pubblicazione. Se l'utente ha corsi attivi, quest'ultimi compariranno in cima all'elenco dei corsi. Nella seconda vista, l'utente ha la possibilità di visualizzare il contenuto di tutti i moduli delle lezioni o solo di quelli gratuiti. Il contenuto dei moduli può essere sia multimediale, come video YouTube, video locali in formato mp4, immagini, presentazioni in ppt e documenti in pdf, sia hyperlink, ovvero riferimenti ad altri siti. Inoltre, la società presso cui ho svolto l'attività di tirocinio, ha richiesto di implementare il lavoro aggiungendo particolari funzionalità,

che assicurassero ad esempio che l'ordine di visualizzazione dei corsi, delle lezioni e dei moduli fosse rispettato, che permettesse di ottenere informazioni sulla progressione di un utente nell'affrontare un corso a lui associato e, infine, garantissero la possibilità di personalizzare un corso attivo disabilitando i moduli per un singolo utente.

Per lo sviluppo del progetto è stato utilizzato Laravel Sail con ambiente Docker e avendo a disposizione Windows è stato necessario ricorrere all'utilizzo di WLS.

L'attività di tirocinio si è svolta in remoto. A tal proposito, GitHub si è rivelata una piattaforma essenziale per la realizzazione del lavoro. Essa ha consentito la condivisione del codice sorgente e la possibilità di lavorare sul codice contemporaneamente da diversi dispositivi, ha permesso di gestire in maniera organizzata le missioni assegnate dalla società, di segnalare i bug da correggere e revisionare i lavori terminati. Per lo sviluppo dell'applicazione web sono stati utilizzati i linguaggi HTML per la struttura e CSS per lo stile. Per migliorare l'esperienza utente e renderla dinamica, ci si è avvalsi della libreria jQuery di Javascript e della tecnologia Ajax. L'applicazione segue l'architettura MVC, in particolare il "UserController" gestisce il flusso di dati e la navigazione degli utenti richiamando il modello principale "Catalog" nel quale vi sono i metodi per l'acquisizione dei corsi e dei loro contenuti. Nei metodi è presente anche la logica per l'autorizzazione di visualizzazione di un contenuto da parte di un determinato utente. Le funzioni jQuery gestiscono le interazioni dell'utente con la vista "Lezioni" in modo dinamico, senza aggiornamento della pagina. Esse cambiano la struttura HTML in base al tipo di contenuto del modulo selezionato dall'utente, per la sua corretta riproduzione o visualizzazione, e impostano il valore dei bottoni di navigazione.

Capitolo 2

Contesto applicativo

In un'epoca caratterizzata da una rapida evoluzione tecnologica e dalla globalizzazione, la formazione e l'aggiornamento continuo hanno assunto un ruolo sempre più centrale nella vita di ogni individuo. In questo contesto, l'e-learning emerge come una soluzione flessibile e accessibile che può soddisfare diverse esigenze formative. Il presente capitolo mira a fornire un quadro complessivo del settore dell'e-learning, illustrando le sue caratteristiche principali ed il profilo degli utenti; verranno analizzate alcune delle piattaforme di e-learning più note e utilizzate ponendo particolare attenzione alle interazioni che gli utenti effettuano all'interno di esse. Successivamente, verranno discusse le sfide e le opportunità che caratterizzano

questo settore in crescita, offrendo una panoramica equilibrata dei suoi vantaggi e svantaggi.

L'e-learning, o apprendimento elettronico, rappresenta una modalità di formazione che sfrutta le tecnologie digitali per fornire contenuti educativi; si è evoluto da semplici corsi online a un ecosistema complesso che include una varietà di strumenti, piattaforme e metodologie di apprendimento. In questo contesto, il termine "e-learning" abbraccia una vasta gamma di applicazioni e processi, come Web-based learning¹, aule virtuali, e piattaforme di gestione dell'apprendimento (Learning Management Systems, LMS²).

Nel panorama dell'e-learning, numerose piattaforme si sono distinte per la loro qualità e offerta formativa, ma anche per la loro interfaccia utente e le funzionalità che propongono. Coursera è un esempio emblematico di collaborazione stretta con università e istituti accademici di rilievo globale. La sua interfaccia è intuitiva e permette agli studenti di accedere facilmente ai corsi accademici, tecnici o persino a interi percorsi di laurea online. La dashboard personale mostra all'utente un quadro chiaro del proprio percorso formativo, facilitando la gestione e il monitoraggio dei corsi [1]. Udemy, invece, presenta un modello più aperto, in cui chiunque può diventare istruttore. La sua interfaccia è funzionale, con una ricerca avanzata che permette di esplorare una vasta gamma di argomenti, dai più tecnici ai più creativi.

¹ si riferisce all'uso di risorse online e ambienti digitali per insegnare e apprendere

² è una piattaforma software utilizzata per pianificare, implementare, gestire e valutare processi di apprendimento. Queste piattaforme sono ampiamente utilizzate in contesti educativi e aziendali per la somministrazione di corsi online, la gestione di programmi di formazione, e la tracciabilità dei progressi degli studenti o dei dipendenti.

L'accento è posto sull'individualizzazione dell'esperienza, con suggerimenti personalizzati basati sugli interessi dell'utente [2]. EdX combina l'approccio accademico di Coursera con una varietà di offerte formative, tra cui corsi professionali e "MicroMasters". Quest'ultima piattaforma si distingue per una navigazione semplice e una dashboard chiara, nella quale è possibile avere una panoramica dei corsi in cui si è iscritti e del progresso compiuto [3]. Khan Academy, con la sua missione no-profit, offre una piattaforma user-friendly con corsi gratuiti su una vasta gamma di materie. L'interfaccia è pulita e diretta, con risorse didattiche facilmente accessibili e una forte attenzione alla community e all'interazione tra gli utenti [4]. Oltre alla varietà dei contenuti formativi, queste piattaforme condividono alcune funzionalità chiave. Tutte presentano una dashboard personale, un motore di ricerca avanzato e filtri per facilitare la selezione dei corsi. La presenza di elementi interattivi, come quiz, esercizi e progetti, è centrale e mira a promuovere un apprendimento attivo.

La componente sociale è altresì essenziale: forum, spazi comunitari e discussioni animano la vita della piattaforma. Inoltre, la compatibilità mobile, tramite app native o interfacce ottimizzate, rende l'accesso al materiale didattico comodo e immediato. Infine, la possibilità di ottenere certificati valorizza ulteriormente l'esperienza formativa, rendendo questi strumenti utili anche nel mondo professionale e accademico.

L'e-learning riesce a soddisfare le esigenze formative di un pubblico estremamente variegato: da studenti universitari a professionisti, passando per lifelong learners e individui in contesti meno favoriti. È possibile distinguere diverse categorie di utenti alcune delle quali tendono a emergere con maggiore frequenza. Tra queste, quella degli studenti universitari rappresenta una delle più significative, che sfrutta tali

piattaforme per ampliare le proprie competenze accademiche o acquisire abilità che non sono fornite dal classico percorso formativo. Inoltre, i professionisti utilizzano le piattaforme di e-learning per rimanere aggiornati su nuove tecnologie, normative o metodologie di lavoro. Per giunta, le piattaforme di e-learning sono una risorsa preziosa per gli appassionati di lifelong learning, che considerano l'istruzione come un percorso continuo e sono sempre alla ricerca di nuovi argomenti da esplorare. Non da meno, queste piattaforme rappresentano una fonte di formazione per persone in paesi in via di sviluppo o in situazioni in cui l'accesso all'istruzione è limitato per ragioni economiche, geografiche o sociali. L'e-learning fornisce un'opzione di formazione più flessibile rispetto ai tradizionali modelli educativi permettendo così anche agli individui con impegni familiari o lavorativi di seguire i corsi a loro ritmo. In generale, gli utenti di piattaforme di e-learning sono motivati da una serie di fattori che vanno dal desiderio di avanzamento professionale alla pura curiosità individuale. Tuttavia, tutti gli utenti condividono la necessità di un ambiente di apprendimento che sia duttile nonché strutturato, in grado di adattarsi alle proprie esigenze specifiche pur fornendo un percorso educativo coerente e ben definito.

Nonostante i numerosi vantaggi dell'e-learning, questo settore è soggetto a diverse problematiche che possono influenzare l'efficacia complessiva dell'apprendimento online, come ad esempio: la necessità di disporre di un dispositivo con accesso alla rete Internet, l'accessibilità, il coinvolgimento degli studenti, la personalizzazione dell'apprendimento, la valutazione delle competenze, la gestione dei dati e i costi. Il superamento di tali sfide richiede un approccio multidisciplinare e una collaborazione continua tra diverse aree di competenza per garantire un'esperienza di apprendimento efficace ed inclusiva. Tra le problematiche sopra citate una delle più significative è

l'accessibilità digitale per le persone con disabilità. Poiché l'e-learning si basa su piattaforme digitali, è fondamentale garantire che queste siano progettate in modo tale da essere accessibili a tutti, inclusi coloro che hanno bisogni speciali. Ciò può includere la necessità di fornire opzioni di navigazione alternative per utenti con limitazioni visive o uditive, nonché garantire che i contenuti siano presentati in formati compatibili con gli strumenti di assistenza per la disabilità. Un'altra sfida chiave riguarda l'interazione e il coinvolgimento degli studenti. Poiché l'e-learning si realizza principalmente in un contesto virtuale, può essere più difficile creare un ambiente di apprendimento coinvolgente e interattivo rispetto all'insegnamento in aula. È importante sviluppare strategie didattiche e strumenti che stimolino la partecipazione attiva degli studenti, incoraggiando la collaborazione, la discussione e la risoluzione di problemi. La personalizzazione dell'apprendimento rappresenta un'altra problematica significativa nell'e-learning. Gli utenti hanno esigenze, abilità e stili di apprendimento diversi, e ciò richiede un approccio educativo che consenta di adattare i materiali didattici e le attività per soddisfare le esigenze individuali. La personalizzazione può essere ottenuta attraverso l'utilizzo di strumenti tecnologici, l'analisi dei dati sugli studenti e l'implementazione di piani di studio su misura. La valutazione delle competenze è un aspetto cruciale dell'e-learning. Mentre in un contesto tradizionale è relativamente semplice valutare il progresso degli studenti attraverso esami in aula, l'e-learning richiede approcci valutativi diversi. La sfida consiste nel progettare metodi di valutazione efficaci che siano adeguati alla modalità di apprendimento online e che misurino in modo accurato le competenze acquisite dagli studenti. La gestione dei dati rappresenta un elemento essenziale per garantire il corretto funzionamento di una piattaforma di e-learning. Con un numero crescente di

utenti, corsi e materiali didattici, la gestione efficiente dei dati diventa fondamentale. Ciò include la conservazione sicura dei dati sensibili, la loro integrità e la capacità di monitorare i progressi e le performance degli utenti. Infine, i costi associati all'implementazione e alla gestione delle piattaforme di e-learning possono costituire un ulteriore ostacolo. La creazione e il mantenimento di un ambiente virtuale di apprendimento possono richiedere investimenti significativi, sia a livello di infrastrutture tecnologiche che di risorse umane.

Per affrontare queste problematiche e garantire un apprendimento efficace e di alta qualità, è necessario un approccio integrato e multidisciplinare. La collaborazione tra esperti di educazione, tecnologi dell'informazione, designer didattici e altri professionisti è fondamentale per sviluppare soluzioni innovative e adattabili alle sfide in continua mutazione dell'e-learning.

In conclusione, l'e-learning rappresenta un formidabile veicolo di trasformazione educativa e sociale, capace di rompere le barriere tradizionali e di adattarsi ai cambiamenti rapidi e alle diverse esigenze che caratterizzano il mondo contemporaneo.

Capitolo 3

Obiettivi del progetto

Il progetto si concentra sulla progettazione e realizzazione di pagine web per l'accesso ai contenuti di una piattaforma e-learning. Queste pagine, destinate ad essere integrate in siti web di aziende, sono finalizzate alla divulgazione di corsi di formazione specificamente rivolti alle diverse categorie di personale, come operai e rappresentanti.

La richiesta principale è stata quella di fornire un ambiente in cui fosse possibile caricare, organizzare e visualizzare un'ampia gamma di corsi, da esporre in modo chiaro e strutturato. Dunque, le principali funzionalità sviluppate riguardano il livello utente, per permettere agli utenti finali l'interazione con il catalogo dei corsi, le lezioni

e i moduli associati. Un amministratore del sistema ha il compito di caricare i corsi sulla piattaforma e di organizzarli in lezioni e moduli. Questi moduli possono includere una varietà di elementi multimediali, come video integrati da YouTube, video in formato .mp4, immagini, nonché link di siti web o file come presentazioni in PowerPoint o documenti PDF. Ogniqualvolta un utente registrato sceglie un corso deve richiederne l'attivazione all'amministratore contattandolo personalmente, dopodiché otterrà l'accesso completo a tutti i contenuti del corso. Nel caso in cui un corso non sia attivo per un utente, questo avrà l'accesso limitato ai soli moduli etichettati come "GRATUITI" del corso ed alle relative informazioni.

Oltre agli obiettivi primari che costituiscono il cuore della piattaforma di e-learning, il progetto mira a realizzare una serie di obiettivi secondari. Questi aspetti, sebbene non siano fondamentali come i requisiti principali, giocano un ruolo cruciale nell'ottimizzazione dell'esperienza utente. Gli obiettivi secondari puntano a migliorare vari aspetti, come la strutturazione dei contenuti, la personalizzazione, l'usabilità e il monitoraggio, che, se realizzati con successo, contribuiscono a rendere la piattaforma più robusta, intuitiva e coinvolgente.

Un obiettivo secondario fondamentale consiste nel fornire una struttura di contenuti ben organizzata che serva sia agli utenti che all'amministratore del sistema. Per gli utenti, un'organizzazione chiara e logica nel design della piattaforma favorisce una navigazione immediata, incoraggiando così un percorso di apprendimento lineare e progressivo. Parallelamente, per l'amministratore, è essenziale avere la capacità di personalizzare l'ordine di visualizzazione dei corsi nel catalogo. Tale funzionalità gli permette di evidenziare i corsi di maggiore rilevanza o quelli che ricevono più interazioni, come clic o visualizzazioni, posizionandoli in cima alla lista. In questo

modo, l'amministratore può guidare l'attenzione degli utenti verso contenuti specifici, migliorando così l'efficacia complessiva della piattaforma.

Un altro obiettivo secondario di rilievo è la creazione di meccanismi efficaci per monitorare il livello di coinvolgimento e partecipazione degli utenti durante il loro percorso formativo. Il sistema prevede la raccolta e l'analisi di dati comportamentali, come il tempo trascorso su ciascun modulo, la frequenza con cui vengono visualizzate le lezioni e le interazioni con i contenuti multimediali. Questo insieme di metriche fornirà all'amministratore una visione dettagliata del grado di interesse e dell'efficacia dei corsi offerti. Il monitoraggio attivo permetterà di individuare eventuali ostacoli nell'apprendimento o nel coinvolgimento, dando così la possibilità di intervenire tempestivamente con miglioramenti o supporto personalizzato. Per esempio, se un utente trascorre un tempo eccessivamente lungo su una specifica lezione, ciò potrebbe segnalare una sua difficoltà e dunque, la necessità di fornire una spiegazione più chiara o ulteriore materiale didattico. Questo livello di analisi sull'attività dell'utente non solo migliora la qualità dell'esperienza di apprendimento, ma fornisce anche strumenti preziosi per l'ottimizzazione continua dei contenuti e delle strategie formative.

Oltre a garantire un'interfaccia user-friendly e un percorso intuitivo, un'altra richiesta secondaria è quella di facilitare la navigazione tra diversi livelli di contenuti: tra i corsi, tra le lezioni e tra i moduli. Gli utenti dovrebbero essere in grado di passare agevolmente da un modulo all'altro all'interno di una stessa lezione, così come tra diverse lezioni all'interno di un corso, o persino tra diversi corsi. Questa fluidità nella navigazione è fondamentale per mantenere elevati livelli di coinvolgimento e rendere l'esperienza di apprendimento più piacevole e meno frustrante. Elementi di design

come breadcrumb, pulsanti di navigazione evidenti e un menù strutturato hanno contribuito alla realizzazione di questa finalità, permettendo agli utenti di concentrarsi sul contenuto dei corsi piuttosto che sul modo per accedervi.

Un altro obiettivo secondario fondamentale è la possibilità di fornire un percorso formativo personalizzato in base al profilo e alle esigenze di ciascun utente. L'amministratore del sistema avrà l'opportunità di attivare specifici corsi, o addirittura singole lezioni all'interno di un corso, per singoli utenti o gruppi target. Ad esempio, se un utente necessita di formazione solo in determinate aree, l'amministratore può attivare esclusivamente le lezioni pertinenti a quelle aree, rendendo il percorso formativo più focalizzato e meno oneroso in termini di tempo per l'utente. Questa personalizzazione va oltre la semplice attivazione o disattivazione di corsi. Infatti, essa consente una gestione granulare che può essere modulata in base a diversi fattori come il ruolo dell'utente, le competenze pregresse o le esigenze specifiche del momento. In questo modo, si crea un ambiente di apprendimento più adattativo e responsivo, che non solo migliora l'efficacia formativa ma aumenta anche l'interazione e la soddisfazione degli utenti.

Dopo aver delineato complessivamente gli obiettivi specifici del progetto, è opportuno fornire una panoramica del materiale e della documentazione che è stata messa a disposizione da PartnerRevolution. Questo materiale è servito come guida fondamentale nello sviluppo delle due pagine web principali che mi sono state commissionate, quali: "Catalogo dei Corsi" e "Lezioni". Entrambe queste pagine sono state sviluppate con un occhio di riguardo ai dettagli e agli obiettivi sia primari che secondari precedentemente descritti, al fine di assicurare che l'interfaccia finale risultasse consistente con le aspettative e le necessità dell'azienda.

L'immagine rappresentativa del "Catalogo dei Corsi", si osservi la figura 1, serve come mock-up visuale per la pagina principale del sistema e-learning. Dunque, al suo ingresso nella piattaforma, l'utente registrato si trova davanti ad un'interfaccia caratterizzata da una specifica struttura come è possibile vedere nella figura citata.

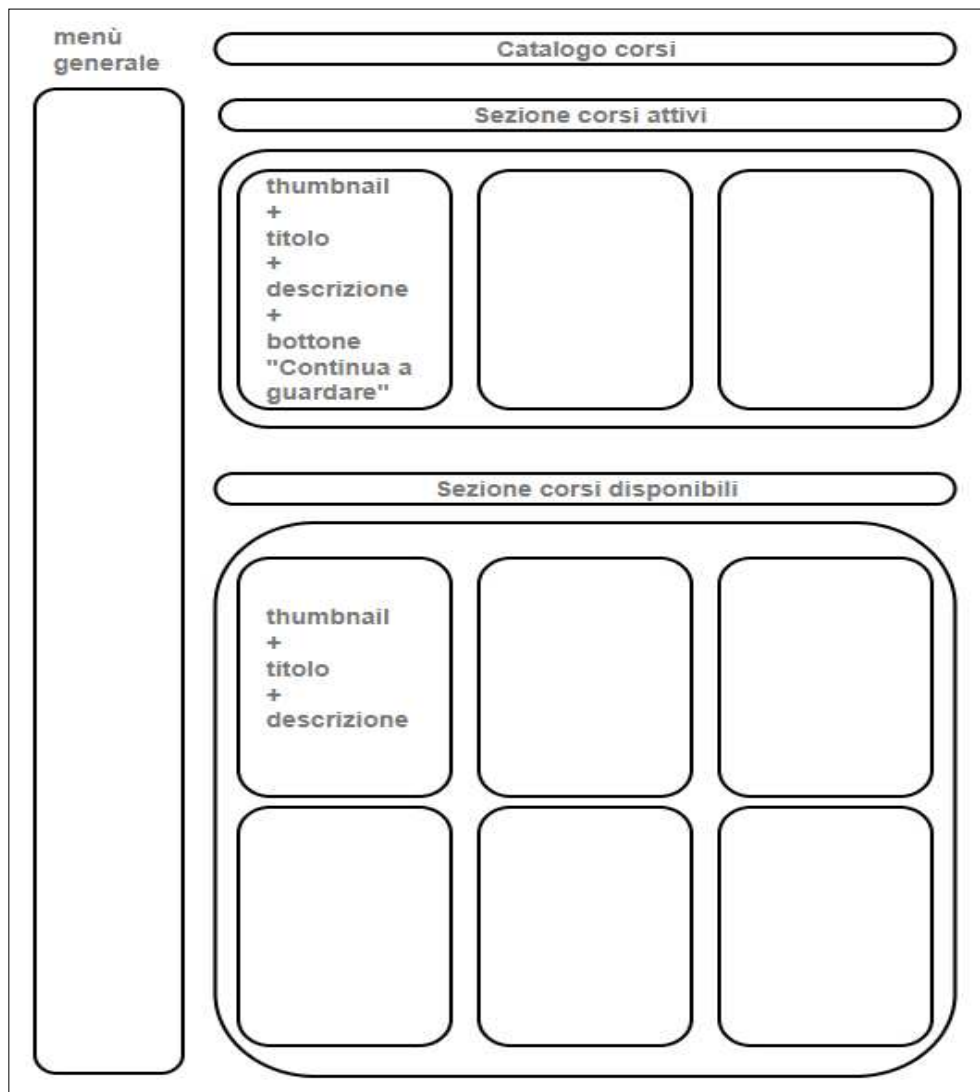


Figura 1- Mock-up visuale della pagina "Catalogo dei Corsi"

La pagina è divisa in due sezioni principali: la prima dedicata ai corsi attivi per l'utente, e la seconda che mostra l'elenco completo dei corsi disponibili pubblicati dall'amministratore. Nella sezione dei corsi attivi, ciascun corso è identificato da un riquadro che include un'immagine evocativa, un titolo e una breve descrizione. È presente un pulsante etichettato con la dicitura "Continua a guardare" che permette all'utente di riprendere direttamente l'ultimo modulo visualizzato. Nella sezione dei corsi disponibili, l'utente può visualizzare un'anteprima di tutti i corsi attivabili, inclusi quelli in fase di progettazione contrassegnati come "in arrivo". Ciascun corso è presentato con un'immagine, un titolo e una breve descrizione, similmente alla sezione dei corsi attivi. Tale disposizione permette immediatamente all'utente di comprendere quali corsi sono attivi e quali sono disponibili o in arrivo, garantendo così un percorso fluido e coinvolgente.

La pagina delle "Lezioni", è una sorta di "hub" per l'interazione dell'utente con il materiale didattico, e rappresenta il nucleo dell'esperienza di apprendimento e-learning offerta dalla piattaforma. Una volta selezionato un corso dal catalogo, l'utente viene reindirizzato nella pagina delle relative lezioni, che è organizzata in modo da fornire un quadro completo e immersivo del corso. Da come si può vedere nella figura 2 l'utente è accolto da un titolo accattivante e da una descrizione sintetica che ha lo scopo di facilitare il suo approccio al corso, anticipandone il contenuto. Successivamente, si apre una sezione dedicata ai contenuti multimediali, dove elementi come video, immagini e documenti vengono riprodotti o mostrati. Un aspetto cruciale di questa pagina è la sezione delle lezioni, strutturata come una tabella facile da leggere e da navigare. Ogni lezione è un elemento cliccabile che, al tocco, rivela la

lista dei moduli associati. Cliccando su un modulo, questo viene eseguito o visualizzato nella sezione multimediale, occupando una posizione di rilievo sulla pagina. Una funzionalità particolarmente notevole è la differenziazione tra contenuti a cui l'utente ha effettivamente accesso. Se un corso è stato attivato per un utente da

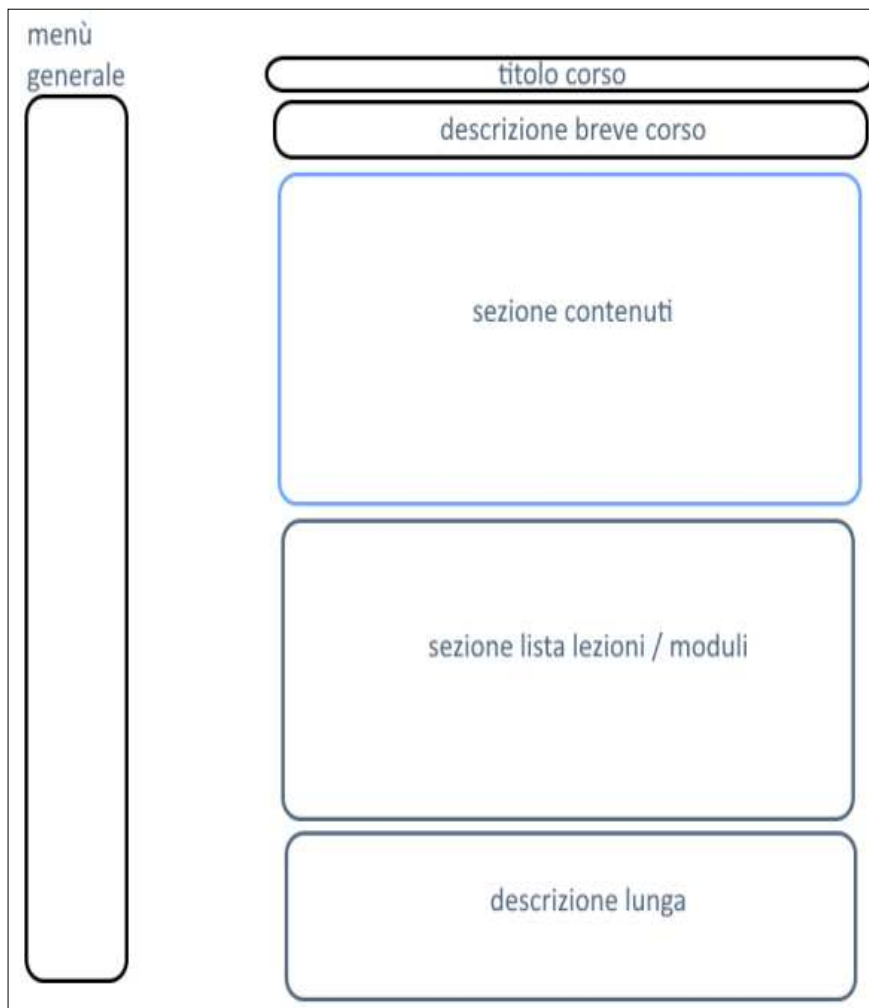


Figura 2 - Mock-up visuale pagina "Lezioni"

parte dell'amministratore, tutti i moduli e lezioni associati sono disponibili per la

visualizzazione. In caso contrario, l'utente può accedere solo ai moduli designati come “gratuiti”, mentre gli altri moduli appaiono in forma grigia e non sono cliccabili.

Una descrizione dettagliata chiude il layout della pagina, offrendo una vista d'insieme che completa l'esperienza formativa e informativa dell'utente.

Questo capitolo ha delineato in dettaglio gli obiettivi principali e secondari del progetto di tirocinio, focalizzandosi sullo sviluppo della piattaforma e-learning. Ha esplorato le funzionalità fondamentali richieste, come la memorizzazione delle informazioni relative alla visualizzazione dei corsi e la strutturazione gerarchica dei contenuti in corsi, lezioni e moduli. Gli obiettivi secondari hanno conferito ulteriori complessità e funzionalità al progetto. Tra questi, il monitoraggio dell'attività dell'utente, l'ottimizzazione dell'usabilità e della navigabilità e la possibilità di personalizzare i corsi per utenti specifici non solo elevano l'esperienza dell'utente finale, ma forniscono anche all'amministratore gli strumenti per gestire efficacemente la piattaforma. Le immagini rappresentative delle pagine “Catalogo dei Corsi” e “Lezioni” servono come un quadro visuale degli obiettivi preposti, offrendo una visione concreta di come la teoria si traduce in pratica.

Capitolo 4

Strumenti utilizzati

Per lo sviluppo della piattaforma di e-learning, è stata impiegata una serie di strumenti software e tecnologie per garantire un ambiente di sviluppo stabile, performante e completo. Di seguito, verranno presentati i principali strumenti utilizzati e fornita una descrizione dettagliata delle loro funzionalità e utilità.

Laravel 10 è un framework PHP che ha notevolmente semplificato lo sviluppo web, offrendo una sintassi intuitiva e ricca di funzionalità. Uno tra i principali vantaggi che ho tratto nell'utilizzare Laravel è stata la sua architettura MVC (Model-View-Controller). Tale pattern di progettazione non solo separa la logica dell'applicazione dalla sua presentazione, ma facilita anche la manutenzione e la scalabilità del software

[5]. Nell'ambito del progetto, i modelli hanno gestito la logica di business e l'interazione con il database, i controller hanno gestito la logica dell'applicazione e direzionato i dati tra i modelli e le viste, e infine, quest'ultime hanno gestito la presentazione dei dati. Questa separazione di compiti ha reso molto più semplice gestire ed estendere le diverse parti dell'applicazione e-learning. Tra le numerose funzionalità di Laravel 10, è importante sottolineare anche l'uso di Eloquent ORM (Object-Relational Mapping) come strumento chiave per la gestione dei dati. Eloquent ORM fornisce un'interfaccia elegante e semplice per interagire con il database, permettendo la creazione, la lettura, l'aggiornamento e la cancellazione di record attraverso oggetti, eliminando così la necessità di dover scrivere query SQL. Questa astrazione dal database non solo aumenta la produttività durante lo sviluppo, ma riduce anche la probabilità di errori in quanto le query vengono gestite dal framework stesso. Con Eloquent, i modelli rappresentano le tabelle del database e le loro relazioni, fornendo un modo intuitivo di accedere e manipolare i dati direttamente dal codice [6]. Un'altra caratteristica fondamentale di Laravel 10 è il suo sistema di routing avanzato. Questo sistema permette di definire con estrema facilità le URL e le rotte dell'applicazione. Nel progetto, la gestione delle rotte è stata essenziale per rendere chiara l'esperienza utente, permettendo una navigazione fluida attraverso le diverse sezioni del portale e-learning [7].

Oltre alla sua architettura MVC, al sistema di routing, e all'uso di Eloquent ORM, Laravel offre un'ampia gamma di funzionalità integrate per la gestione delle sessioni, l'autenticazione e l'interazione con diversi tipi di database. Queste caratteristiche hanno facilitato notevolmente l'implementazione della sezione di accesso di una piattaforma e-learning.

La scelta di utilizzare Laravel 10 come base tecnologica per questo progetto è stata dettata dalla necessità di avere un framework solido, scalabile e con una comprovata affidabilità. Laravel ha brillantemente risposto a queste esigenze, fornendo gli strumenti necessari per la realizzazione di un progetto che ha soddisfatto tutti gli obiettivi preposti.

Nel corso dello sviluppo del progetto e-learning per PartnerRevolution, anche Laravel Sail ha rivestito un ruolo cruciale agevolando la gestione dell'ambiente di sviluppo locale. Attraverso l'utilizzo di Sail, è stato possibile imitare le condizioni del server di produzione in un ambiente locale, garantendo in tal modo un'esperienza di sviluppo uniforme e riducendo i potenziali problemi di compatibilità. Sail si integra strettamente con Docker, fornendo un set di servizi containerizzati essenziali per lo sviluppo, come database, cache e altri servizi back-end [8]. Questa combinazione ha permesso di isolare vari componenti del progetto, semplificando la replicazione e la condivisione di configurazioni di sviluppo tra diversi membri del gruppo di lavoro.

Tra le funzionalità più utilizzate di Sail, vi sono una serie di comandi specifici per la gestione dell'ambiente di sviluppo. Il comando “sail up” è utilizzato per inizializzare e avviare tutti i servizi definiti nel file Docker Compose, configurando automaticamente le variabili d'ambiente e le dipendenze necessarie. Al contrario, il comando “sail down” permette di fermare tutti i servizi e, se necessario, rimuovere i container e le risorse associate. Un altro comando vantaggioso è “sail shell”, che offre un accesso diretto al sistema operativo del contenitore Docker, permettendo di effettuare una serie di operazioni avanzate come la gestione delle dipendenze o la risoluzione di problemi specifici all'interno del container [9].

Laravel Sail non solo ha semplificato la configurazione e la gestione dell'ambiente di sviluppo locale, ma ha anche fornito una gamma di strumenti che hanno reso l'intero processo di sviluppo più efficiente e meno incline a errori.

Nell'ambiente di sviluppo del progetto e-learning, Docker ha occupato una posizione centrale come piattaforma per la containerizzazione di applicazioni. Grazie alla sua capacità di impacchettare software in unità standardizzate chiamate container, Docker ha fornito un livello di flessibilità e portabilità che si è rivelato indispensabile per l'omogeneità dell'ambiente di sviluppo e di produzione. Come anticipato Docker opera basandosi sul principio della containerizzazione, metodo che incapsula un'applicazione e le sue dipendenze in un 'container'. Questo permette all'applicazione di essere eseguita in modo uniforme, indipendentemente dalle peculiarità del sistema operativo o dell'hardware su cui viene eseguita [10]. L'utilizzo di Docker ha agevolato significativamente la realizzazione di un ambiente di sviluppo replicabile e attendibile, contribuendo alla coerenza tra sviluppo e produzione e riducendo la complessità associata alla gestione delle infrastrutture software.

La Windows Subsystem for Linux (WSL2) ha permesso di eseguire un vero sistema Linux direttamente su una macchina Windows, eliminando così molti dei tradizionali problemi di compatibilità tra sistemi operativi diversi [11]. Nel contesto del progetto, l'accesso al sistema Linux tramite Windows Subsystem for Linux 2 (WSL2) è stato semplificato dall'integrazione nativa offerta da Windows. Per avviare il sistema Linux, è stato sufficiente aprire una nuova finestra del terminale e inserire il comando `wsl` per accedere direttamente alla distribuzione Linux preinstallata. Questo ha permesso di eseguire comandi Linux e applicazioni direttamente da Windows, senza la necessità di avviare una macchina virtuale o connettersi a un server remoto. L'integrazione di WSL2 con Visual Studio Code ha ulteriormente agevolato questa transizione,

permettendo di aprire progetti direttamente nell'editor con un ambiente di sviluppo completamente Linux. Quindi, WSL2 ha fornito un ambiente Linux completo, altamente performante e perfettamente integrabile con altri strumenti come Docker, creando così un ambiente di sviluppo robusto e versatile. La sua adozione ha rappresentato un notevole vantaggio nell'ambito del progetto, consentendo un agile sviluppo.

Nel contesto di sviluppo dell'applicazione e-learning, Google Chrome è stato utilizzato come browser principale per testare e visualizzare le funzionalità del sistema. La scelta di Chrome non è stata casuale; il browser offre una serie di strumenti di sviluppo integrati che sono stati essenziali per il debug, l'ispezione del DOM, e la visualizzazione delle performance dell'applicazione.

Uno degli aspetti più rilevanti di Chrome è la sua console di sviluppo, che fornisce informazioni dettagliate sul comportamento del codice, come avvisi, errori e log. Inoltre, la scheda "Network" ha permesso di analizzare le richieste HTTP.

Inoltre, la possibilità di simulare diverse dimensioni dello schermo e dispositivi attraverso la modalità 'Responsive' ha consentito di testare l'applicazione in vari scenari, assicurando che l'interfaccia risultasse ottimale sia per dispositivi mobile sia per dispositivi desktop.

Dopo aver descritto gli strumenti impiegati per lo sviluppo locale e l'ambiente di test, è importante spostare l'attenzione sugli aspetti più tecnici e concreti del progetto come la realizzazione del front-end. Questa fase ha richiesto un attento lavoro di scelta e personalizzazione, specialmente in termini di linguaggi da utilizzare per la struttura e la presentazione del sito.

HTML (HyperText Markup Language) è stato il linguaggio di markup fondamentale utilizzato per strutturare i contenuti sul web. Ai fini della realizzazione del progetto,

HTML ha fornito la struttura di base per le diverse pagine e componenti della sezione di accesso ai contenuti di una piattaforma e-learning. Grazie alla sua flessibilità e alla sua integrazione naturale con altri linguaggi e tecnologie, come CSS e JavaScript, HTML ha permesso di costruire un'interfaccia utente funzionale e immediata, garantendo al contempo l'accessibilità e l'usabilità.

Il Cascading Style Sheets (CSS) è stato utilizzato per gestire la presentazione e il layout del progetto. Grazie all'uso di fogli di stile separati e modulari, è stato possibile mantenere una coerenza visiva in tutto il sistema, semplificando al contempo le modifiche e gli aggiornamenti del design. La possibilità di utilizzare classi e identificatori ha reso facile implementare stili specifici per diversi elementi, mentre le funzionalità avanzate come le media query hanno permesso di rendere il progetto responsivo e adattabile a diverse dimensioni di schermo.

Entrambi i linguaggi sono risultati essenziali per realizzare un progetto che non solo fosse funzionale, ma anche esteticamente piacevole e accessibile. La combinazione di HTML per la struttura e CSS per lo stile ha consentito di creare un'esperienza utente agevole e coinvolgente, rispettando gli standard del design moderno e ha contribuito a raggiungere gli obiettivi del progetto.

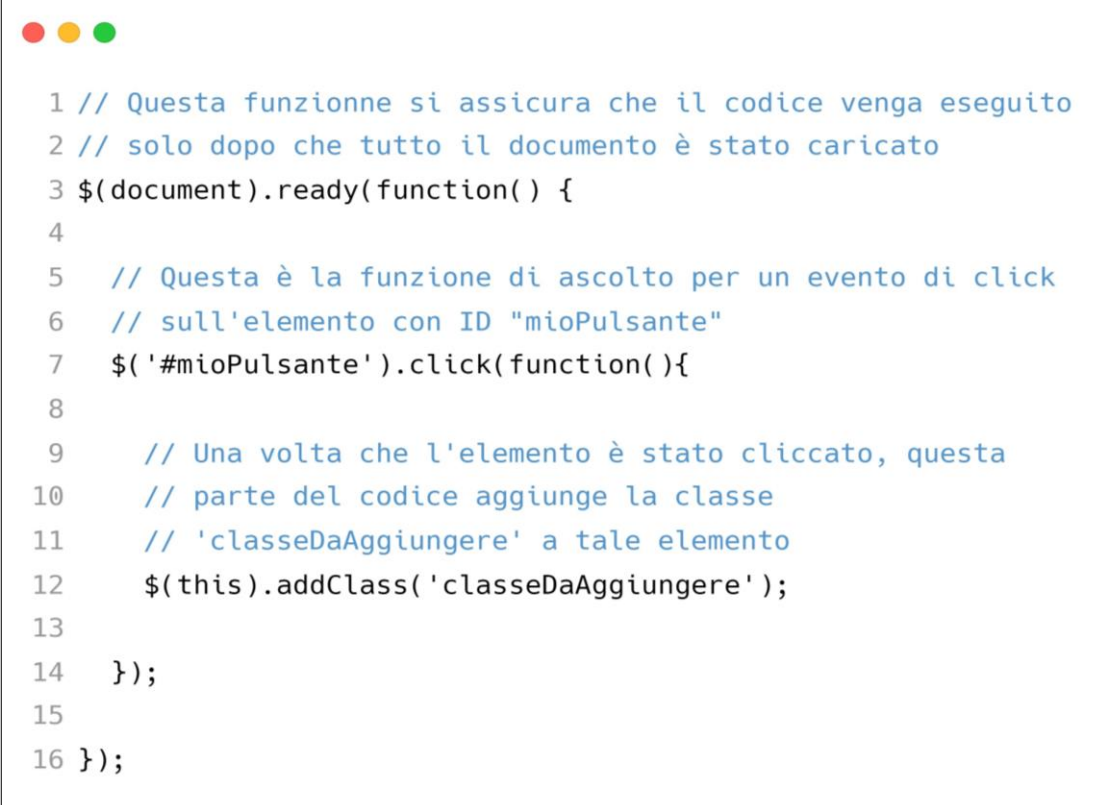
In un ecosistema digitale sempre più esigente e dinamico, JavaScript emerge come uno strumento indispensabile per lo sviluppo di soluzioni web avanzate. Essendo uno dei tre pilastri del web insieme a HTML e CSS, JavaScript si occupa principalmente di gestire l'interazione e la dinamicità sul lato client. Sebbene HTML fornisca la struttura e CSS l'estetica, è JavaScript che trasforma queste pagine statiche in esperienze dinamiche e interattive, aggiungendo un nuovo livello di dinamicità all'interfaccia utente. Nel contesto del progetto e-learning, JavaScript ha funto da ponte tra la rappresentazione grafica delle informazioni e le operazioni di back-end. JavaScript ha

facilitato la gestione degli eventi, dall'apertura di un modulo didattico al tracciamento del progresso dell'utente. Questo ha permesso di creare un ambiente di apprendimento non solo informativo, ma anche interattivo.

Per massimizzare l'efficienza e la scalabilità del progetto, il linguaggio è stato utilizzato in combinazione con altre tecnologie come jQuery e AJAX. Questi strumenti hanno ulteriormente ampliato le capacità di JavaScript, offrendo una sintassi semplificata, maggiore compatibilità tra browser e metodi più efficienti per la manipolazione del DOM e la gestione delle richieste asincrone.

jQuery è una libreria JavaScript di rapida esecuzione focalizzata sulla semplicità della manipolazione del DOM e dell'evento-handling. Essendo un'astrazione di alto livello del JavaScript nativo, jQuery consente di eseguire operazioni complesse con poche righe di codice. Questo è particolarmente utile quando si tratta di manipolare il DOM, gestire eventi, animare elementi e semplificare una varietà di interazioni tra frontend e backend [12]. Ad esempio, la funzionalità di aggiungere una classe CSS a un elemento specifico una volta cliccato è resa notevolmente più facile grazie a jQuery. Nella figura 3 è possibile osservare un frammento di codice che illustra questa operazione. In questo esempio, il codice attende che il documento sia completamente caricato tramite “\$(document).ready” nella riga 3 della figura. Successivamente, ascolta per un evento di click sull'elemento con l'ID “mioPulsante”. Una volta che questo evento si verifica, la funzione associata al click viene eseguita aggiungendo la classe “classeDaAggiungere” all'elemento cliccato, grazie al metodo “addClass(‘classeDaAggiungere);” nella riga 12. Questa capacità di condensare operazioni complesse in un codice semplice e leggibile è una delle ragioni per cui jQuery è stato ampiamente utilizzato nel progetto. L'uso di jQuery ha non solo

accelerato il processo di sviluppo, ma ha anche migliorato la manutenibilità del codice, rendendo più agevole apportare modifiche e aggiornamenti nel tempo.



```
1 // Questa funzione si assicura che il codice venga eseguito
2 // solo dopo che tutto il documento è stato caricato
3 $(document).ready(function() {
4
5 // Questa è la funzione di ascolto per un evento di click
6 // sull'elemento con ID "mioPulsante"
7 $('#mioPulsante').click(function(){
8
9 // Una volta che l'elemento è stato cliccato, questa
10 // parte del codice aggiunge la classe
11 // 'classeDaAggiungere' a tale elemento
12 $(this).addClass('classeDaAggiungere');
13
14 });
15
16 });
```

Figura 3 - Frammento di codice in jQuery di una funzione

AJAX, che sta per Asynchronous JavaScript and XML, è una tecnica che permette di eseguire richieste HTTP asincrone al server senza dover ricaricare l'intera pagina web e di integrare i dati forniti nella risposta direttamente nella pagina corrente, in modo incrementale. Questa tecnica ha rivoluzionato il modo in cui le applicazioni web interagiscono con i server, rendendo le interazioni più fluide e reattive [16].

Uno dei vantaggi più evidenti di AJAX è la sua capacità di comunicare con il server utilizzando vari metodi HTTP, come GET e POST. Il metodo GET è comunemente utilizzato per richiedere dati da un server senza effettuare alcuna modifica ai dati esistenti. D'altro canto, il metodo POST è utilizzato quando è necessario inviare dati al server, come per esempio durante la sottomissione di un modulo.

```
1 $.ajax({
2
3   // Url del servizio a cui fare la richiesta
4   url:"https:example.com/api/data",
5
6   // Metodo HTTP utilizzato
7   type: "GET",
8
9   // Tipo di dati atteso in risposta
10  dataType: "json",
11
12  success: function(response) {
13    // Codice eseguito in caso di successo
14    // "response" conterrà i dati restituiti dal server
15    console.log(response);
16  },
17
18  error: function(error) {
19    // Codice da eseguire in caso di errore
20    console.log("Si è verificato un errore:", error);
21  }
22 });
```

Figura 4 – Frammento di codice jQuery per una richiesta Ajax

Un altro aspetto da considerare è il tipo di dati che AJAX può gestire. Il formato JSON (JavaScript Object Notation) è uno dei formati più comuni utilizzati nelle interazioni AJAX. Quest'ultimo è leggero, di facile lettura e scrittura, e può essere facilmente convertito in un oggetto JavaScript, rendendolo ideale per il trasporto di dati tra client e server [13]. In figura 4 troviamo un esempio pratico di come effettuare una richiesta AJAX utilizzando jQuery, esso esegue una richiesta GET e aspetta una risposta in formato JSON.

In questo esempio, la funzione “\$.ajax()” è utilizzata per eseguire una richiesta GET. Se la richiesta è completata con successo, il blocco di codice all'interno di “success” nella riga 12 della figura 4, verrà eseguito e la variabile “response” conterrà i dati restituiti dal server in formato JSON. La combinazione di jQuery e AJAX offre un insieme di strumenti che possono rendere qualsiasi applicazione web più interattiva e user-friendly.

GitHub è stato adottato come sistema di controllo delle versioni e repository remoto per il progetto di sviluppo della piattaforma di e-learning. Essenziale per il successo del progetto, GitHub ha offerto un'ampia gamma di strumenti e funzionalità che hanno consentito al team di sviluppo di lavorare in modo efficiente e coordinato.

Uno dei vantaggi principali di GitHub è la possibilità di tenere traccia delle modifiche apportate al codice sorgente durante lo sviluppo. Ogni modifica, nota come "commit", è stata registrata in modo dettagliato nel repository, fornendo una cronologia chiara e accurata delle modifiche effettuate da ogni membro del team. Questa funzionalità è stata di fondamentale importanza per monitorare l'evoluzione del progetto e garantire che ogni modifica fosse documentata e facilmente accessibile.

Oltre alla registrazione delle modifiche, grazie a GitHub i membri del team hanno potuto lavorare contemporaneamente sullo stesso progetto, gestendo in modo

efficiente le diverse versioni del codice. La funzione di "branching" di GitHub ha permesso ai membri del team di lavorare su funzionalità specifiche o correzioni di bug in modo separato, senza interferire con il lavoro degli altri. Successivamente, tali modifiche vengono integrate nel ramo principale del progetto, garantendo un flusso di lavoro ordinato e una corretta integrazione delle diverse funzionalità.

La natura remota di GitHub ha consentito al team di sviluppo di condividere facilmente il codice sorgente e di effettuare il monitoraggio del progetto anche da diverse località geografiche. Questa flessibilità ha reso possibile la collaborazione tra membri del team situati in luoghi diversi, agevolando la comunicazione e la gestione delle attività.

Le funzionalità di problemi (issues) e richieste di pull (pull requests) hanno permesso al team di segnalare e risolvere i bug in modo sistematico e organizzato [14]. I commenti e le discussioni in merito a ciascun problema hanno agevolato la comunicazione tra i membri del team e hanno contribuito a garantire una rapida risoluzione delle problematiche.

GitHub ha inoltre fornito funzionalità avanzate di sicurezza e controllo degli accessi, consentendo di gestire in modo accurato i permessi di accesso al repository e proteggendo il codice sorgente da accessi non autorizzati.

Complessivamente, l'utilizzo di GitHub ha rappresentato un pilastro essenziale per il successo del progetto, fornendo uno strumento efficace e versatile per il controllo delle versioni, la collaborazione e la gestione del codice sorgente. Questo processo è stato seguito insieme al "workflow" adottato, un flusso di lavoro ben strutturato che ha permesso al team di gestire in modo efficace le attività, garantendo di operare in maniera coerente e di migliorare la collaborazione tra i membri.

Il "workflow" adottato durante lo sviluppo del progetto ha seguito un processo ben strutturato per garantire una gestione efficace dei task e delle attività. Inizialmente, il team ha consultato la documentazione generale del progetto per comprendere le linee guida e gli obiettivi globali. Successivamente, i task sono stati assegnati ai membri del team in base alle priorità e alle competenze di ciascuno. Per ogni task assegnato, il membro del team ha creato un branch separato nel repository, utilizzando il proprio nome utente e l'ID del task per identificare in modo univoco il branch. Una volta creato il branch di lavoro, il membro del team ha proceduto con l'implementazione del task assegnato, effettuando commit regolari con commenti chiari per descrivere il tipo di modifica apportata e il numero di riferimento del task. Dopo aver completato lo sviluppo del task, il codice è stato sottoposto a un controllo di qualità interno per verificare che rispettasse gli standard e le specifiche del progetto. Successivamente, il membro del team ha pubblicato il proprio branch sul ramo principale di sviluppo, combinando il lavoro svolto con quello degli altri membri del team. Ogni volta che, durante il controllo di qualità, sono emersi problemi o correzioni necessarie, il task è stato segnalato nella tabella delle attività "incorso" con un commento rappresentativo del bug o della modifica richiesta. Il membro del team ha quindi effettuato le correzioni necessarie, ripetendo i passaggi di sviluppo e controllo di qualità fino a quando il task non ha superato con successo la revisione.

Una volta completate tutte le correzioni e superato il controllo di qualità, il task è stato spostato nella tabella delle attività "Fatte (Da revisionare)". Successivamente, un altro membro del team ha effettuato una revisione del codice sorgente, verificando le modifiche apportate e fornendo feedback o suggerimenti per eventuali miglioramenti. Dopo la revisione del codice, il responsabile del progetto ha approvato il task e lo ha spostato nella tabella "Completate" per indicare che è stato completato con successo.

Questo "workflow" ha consentito di gestire in modo efficace le attività, garantendo un flusso di lavoro coerente, una corretta gestione delle modifiche al codice e una migliore collaborazione tra i membri del team.

Questo capitolo ha fornito un'analisi approfondita degli strumenti e delle tecnologie utilizzate nel corso dello sviluppo della piattaforma di e-learning. Attraverso l'utilizzo di HTML e CSS per la strutturazione e la stilizzazione dell'interfaccia utente, passando per JavaScript e le sue librerie annesse come jQuery per la manipolazione dinamica, fino all'uso delle chiamate AJAX per un'interazione asincrona con il server, abbiamo esplorato la complessità e la versatilità delle tecnologie front-end. Allo stesso modo, abbiamo approfondito l'uso di Laravel 10 e Docker, gestiti attraverso Sail, come tecnologie server-side, fornendo un ambiente ottimale per la gestione dei dati e della logica applicativa. WSL2 è stato utilizzato come sottosistema Linux per Windows, offrendo un ambiente di sviluppo unificato e multipiattaforma. Inoltre, altri importanti strumenti utilizzati sono stati Google Chrome, che ha fornito le necessarie funzionalità di debug e test, e Visual Studio, utilizzato come ambiente di sviluppo integrato, che hanno contribuito a rendere il lavoro più efficiente e meno propenso agli errori. Infine, GitHub si è rivelato fondamentale non solo per il controllo delle versioni, ma anche come piattaforma per la collaborazione e la gestione del progetto. Attraverso un "workflow" ben definito e metodico, il team ha potuto lavorare in modo coordinato e sistematico. Tutti questi strumenti e tecnologie non sono stati selezionati a caso, ma sono frutto di decisioni progettuali. Ognuno ha avuto un ruolo fondamentale nel supportare diversi aspetti del ciclo di vita del progetto, dalla concezione iniziale, alla realizzazione, fino alla manutenzione.

Capitolo 5

Applicazione sviluppata

La creazione di un'applicazione che risponda efficacemente alle esigenze specificate rappresenta una sfida di ingegneria e di design. Questo capitolo rappresenta il cuore dell'analisi, poiché si focalizza sull'applicazione software sviluppata allo scopo di raggiungere gli obiettivi del progetto.

Verrà fornita una visione d'insieme dell'architettura dell'applicazione, approfondendo il ruolo che ogni componente riveste nell'attività complessiva. Saranno esaminati il database, l'interfaccia utente e le caratteristiche tecniche che rendono l'applicazione reattiva e funzionale. Verranno mostrate anche alcune porzioni di codice ritenute particolarmente significative, al fine di fornire una visione chiara dell'approccio alla risoluzione dei problemi affrontati.

L'obiettivo di questo capitolo è quello di offrire una panoramica completa ed esaustiva del lavoro svolto durante il progetto, illustrando non solo la teoria ma anche la pratica che ha portato alla realizzazione dell'applicazione.

Il fulcro dell'applicazione sviluppata risiede nel framework Laravel 10, che funge da pilastro architettonico e fornisce una solida base per l'intera soluzione software. La scelta di questo framework è dettata dalla sua capacità di adattarsi perfettamente alle esigenze di un progetto come il nostro. Laravel, grazie al suo approccio Model-View-Controller (MVC), ha permesso una chiara distinzione tra i vari aspetti dell'architettura del sistema, dunque, è importante illustrare in dettaglio come le varie parti interagiscono tra loro attraverso un diagramma architettonico.

Iniziando dall'alto del diagramma della figura 5, troviamo gli "Utenti", che rappresentano gli attori esterni che interagiscono con il sistema. Questi sono direttamente collegati attraverso frecce bidirezionali alle "Viste". Le viste rappresentano ciò che gli utenti vedono e con cui interagiscono. In particolare, abbiamo due viste principali: "catalogCourses.blade.php", che mostra il catalogo dei corsi disponibili, e "lessons.blade.php", che permette la visualizzazione delle lezioni.

Queste viste, a loro volta, sono connesse al "Controller" ed il controller coinvolto è "UserController", che include diversi metodi responsabili della gestione delle richieste degli utenti sviluppati per presentare i dati in modo appropriato. Tra questi metodi troviamo: "showCatalogCourses()" che si occupa di visualizzare il catalogo dei corsi, "showLessons()" che gestisce la visualizzazione delle lezioni del corso selezionato, e "saveClick()", un metodo che registra le interazioni degli utenti con i contenuti. Successivamente, il controller interagisce con una serie di "Modelli". Questi modelli rappresentano la logica e la struttura dati dell'applicazione e comprendono: "User",

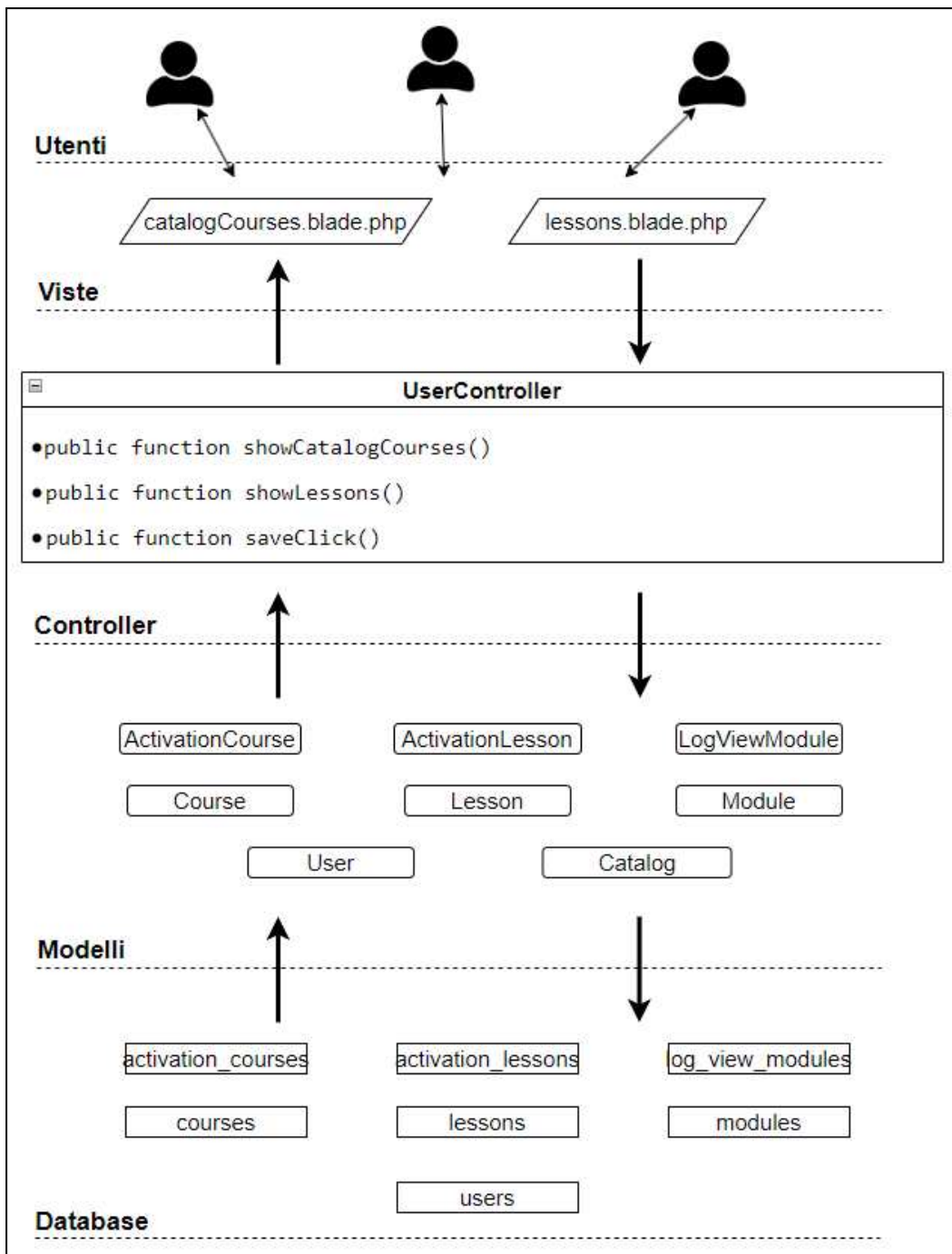


Figura 5 - Diagramma dell'architettura dell'applicazione

“Course”, “Lesson”, “Module”, “ActivationCourse”, “ActivationLesson”, “LogViewModule”, e “Catalog”. Ogni modello è responsabile di un particolare tipo di dato o di funzionalità all'interno dell'applicazione, dal loro nome si intuisce il ruolo che ricoprono.

Infine, in fondo al diagramma, abbiamo il "Database". Questo è la rappresentazione fisica dei dati e include tabelle come “courses”, “lessons”, “modules”, “log_view_modules”, “activation_courses”, “activation_lessons”, e “users”. Queste tabelle memorizzano i dati effettivi, che vengono utilizzati dai modelli per presentare le informazioni agli utenti attraverso le viste.

La piattaforma e-learning che abbiamo progettato si basa su una struttura gerarchica di dati che inizia con i corsi e si ramifica in lezioni e moduli. Questa organizzazione, rappresentata in figura 6 attraverso il relativo schema E-R, permette di offrire un percorso d'apprendimento ben definito e progressivo.

La tabella “courses”, del database su cui appoggia l'applicazione progettata, rappresenta il punto di partenza, fungendo da contenitore principale per il materiale didattico. Ogni corso è univocamente identificato attraverso un codice e include dettagli come il titolo, una breve descrizione, una descrizione estesa, la miniatura del corso, e lo stato, che può assumere vari valori come “abilitato”, “disabilitato”, “bozza” o “preview”. Un elemento fondamentale di questa tabella è la data di creazione, che permette di tener traccia del momento in cui il corso viene introdotto nella piattaforma. Inoltre, tra gli obiettivi secondari rientra l'ordine di visualizzazione dei corsi nel catalogo, dunque, quest'ultimo è indicato dal campo “id_order”.

All'interno di ogni corso, troviamo diverse lezioni. La tabella “lessons” fa riferimento direttamente alla tabella dei corsi attraverso il campo “code_courses”. Ogni lezione,

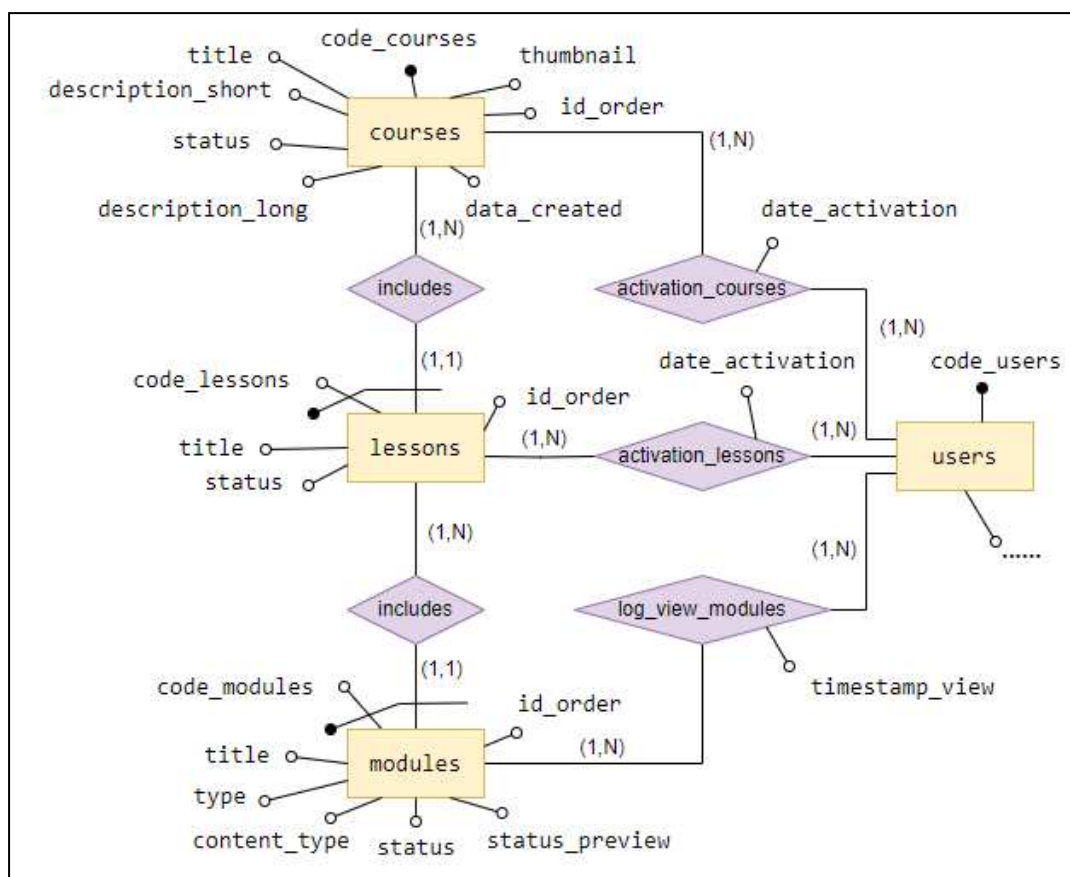


Figura 6 - Schema E-R

quindi, appartiene a un corso specifico e ne rappresenta un segmento o una parte. La lezione ha un suo titolo e uno stato che indica se è attiva o meno. L'ordine delle lezioni all'interno del corso è anch'esso importante e viene gestito attraverso il campo "id_order".

Infine, all'interno di ogni lezione, abbiamo i vari moduli. Questi moduli possono variare in termini di tipo, andando da video incorporati a video embed, presentazioni, link generici e PDF. La tabella dei moduli è collegata alla tabella delle lezioni attraverso il campo "code_lessons", assicurando che ogni modulo sia contestualizzato

all'interno della sua lezione di appartenenza. Anche i moduli hanno un campo "id_order" per definire la sequenza in cui vengono presentati all'utente.

Proseguendo nella definizione delle tabelle, emerge una logica chiara e coerente riguardo le relazioni tra di esse. Le strutture stesse delle tabelle suggeriscono una relazione uno a molti che si manifesta tra i corsi e le lezioni, e analogamente tra le lezioni e i moduli. Questa disposizione rispecchia la gerarchia del materiale didattico: un singolo corso contiene molteplici lezioni, e ogni lezione, a sua volta, può avere diversi moduli. Laravel supporta naturalmente queste relazioni attraverso le sue funzionalità ORM (Object-Relational Mapping). In particolare, è stata sfruttata la funzionalità delle "Eloquent Relationships" di Laravel per stabilire e gestire queste connessioni nel codice. Una trattazione dettagliata su come queste relazioni sono state implementate a livello di model sarà presentata più avanti.

Nella struttura gerarchica del nostro database e-learning, oltre ai corsi, lezioni e moduli, abbiamo anche integrato meccanismi che permettono di monitorare e personalizzare l'esperienza dell'utente. Un elemento chiave in questo sistema è la tabella "activation_courses". Questa tabella tiene traccia di quali corsi sono stati attivati da un particolare utente. Ogni volta che un corso viene attivato per un utente, viene creato un record con un riferimento al corso specifico e all'utente attraverso i campi "code_courses" e "code_users". Questo permette di avere un registro chiaro di quali corsi sono accessibili ad ogni utente e da quando. L'attivazione del corso ha anche uno stato, che può indicare, ad esempio, se il corso è ancora attivo o è stato completato dall'utente.

Per avere una tracciabilità ancora maggiore dell'attività dell'utente, abbiamo creato la tabella "log_view_modules". Questa registra ogni volta che un modulo viene visualizzato da un utente. Ogni record si riferisce al modulo specifico attraverso il

campo “code_modules” e all'utente che l'ha visualizzato attraverso “code_users”. Inoltre, c'è un riferimento al corso a cui appartiene il modulo, e un timestamp che indica esattamente quando il modulo è stato visualizzato. Questo permette di avere una panoramica dettagliata del percorso di apprendimento di ciascun utente e di capire quali parti del corso hanno attirato la loro attenzione.

Infine, la tabella “activation_lessons” funziona in modo simile alla tabella di attivazione dei corsi, ma a livello di lezione. Questa tabella tiene traccia delle lezioni che sono state attivate per ogni utente. Ogni record fa riferimento al modulo specifico della lezione e all'utente attraverso “code_lessons” e “code_users”. Anche in questo caso, la data di attivazione indica quando l'utente ha avuto accesso alla lezione.

Queste tabelle arricchiscono notevolmente la funzionalità della piattaforma, consentendo una personalizzazione dell'esperienza di apprendimento e offrendo strumenti preziosi per monitorare e analizzare l'attività dell'utente. Attraverso questi dati, è possibile offrire contenuti più mirati, comprendere meglio le esigenze degli utenti e ottimizzare l'offerta formativa della piattaforma.

Dopo aver esplorato la struttura del database, è essenziale comprendere come questi dati siano presentati all'utente. La disposizione e la logica delle tabelle nel database sono state concepite non solo per garantire un'ottimale gestione delle informazioni ma anche per assicurare che l'esperienza dell'utente sia intuitiva e gradevole. L'applicazione sviluppata, infatti, non è solo un insieme di dati, ma è un'esperienza formativa completa, e come tale, ogni dettaglio, dalla struttura del database all'interfaccia utente, ha un ruolo cruciale.

La navigazione all'interno della nostra piattaforma didattica è stata progettata per rispondere in modo dinamico alle esigenze degli utenti. Una distinzione fondamentale riguarda gli utenti che hanno corsi attivi e quelli che non ne hanno. Un utente che ha

già attivato corsi, entrando nel Catalogo dei Corsi, si trova di fronte a due distinti elenchi: uno dei corsi che sta attualmente seguendo e l'altro di corsi ancora da esplorare. Cliccando su un corso attivo, l'utente viene guidato direttamente alla pagina delle lezioni del corso, dove ha piena libertà di accedere e interagire con ogni singolo materiale didattico. D'altro canto, un utente che non ha corsi attivi vede solo l'elenco generale dei corsi nel Catalogo. L'utente interessato ad un corso, cliccandoci sopra,

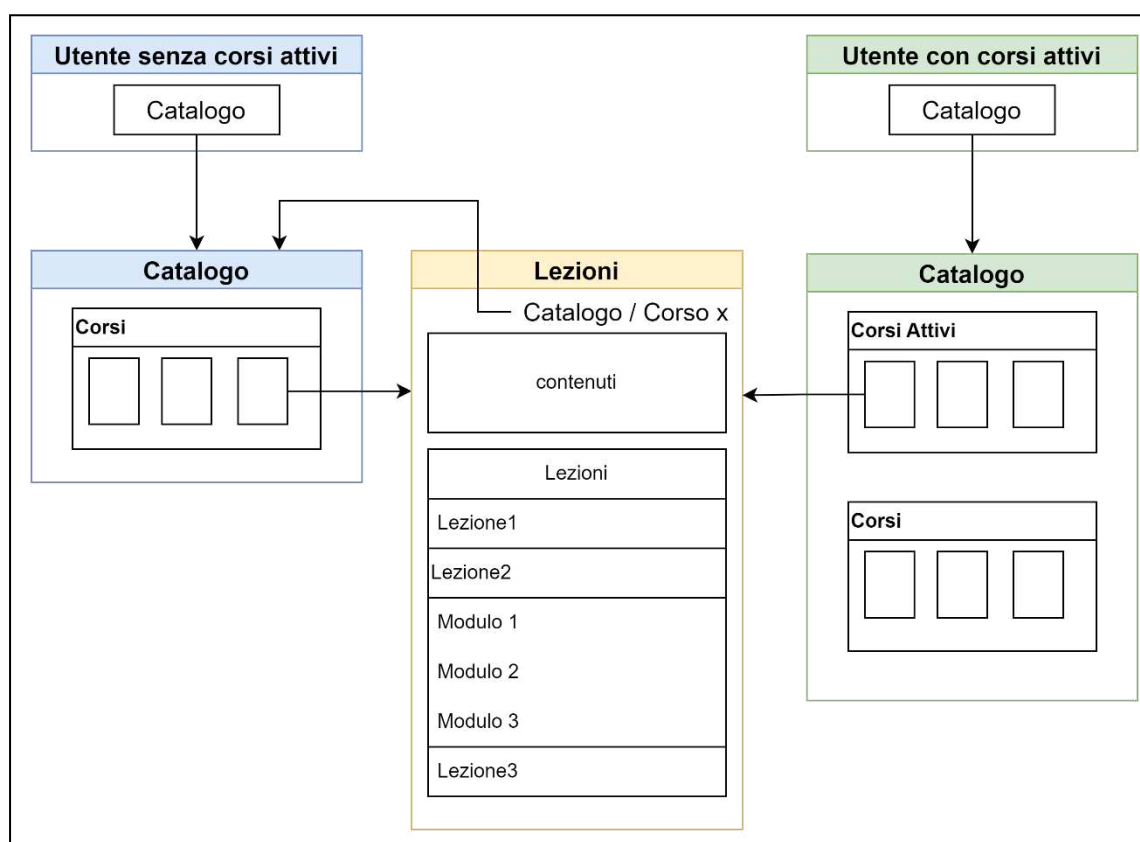


Figura 7 - Diagramma link

viene indirizzato alla pagina delle lezioni di quel corso, ma con una limitazione: potrà interagire solo con i contenuti contrassegnati come "free". In entrambi i percorsi di

navigazione, la presenza di una breadcrumb nella pagina delle lezioni assicura una facile transizione di ritorno al catalogo, offrendo un'esperienza d'uso fluida e senza ostacoli. Per offrire una visione chiara di questa strutturazione, lo schema di figura 7 illustra graficamente i flussi di navigazione e le interazioni degli utenti.

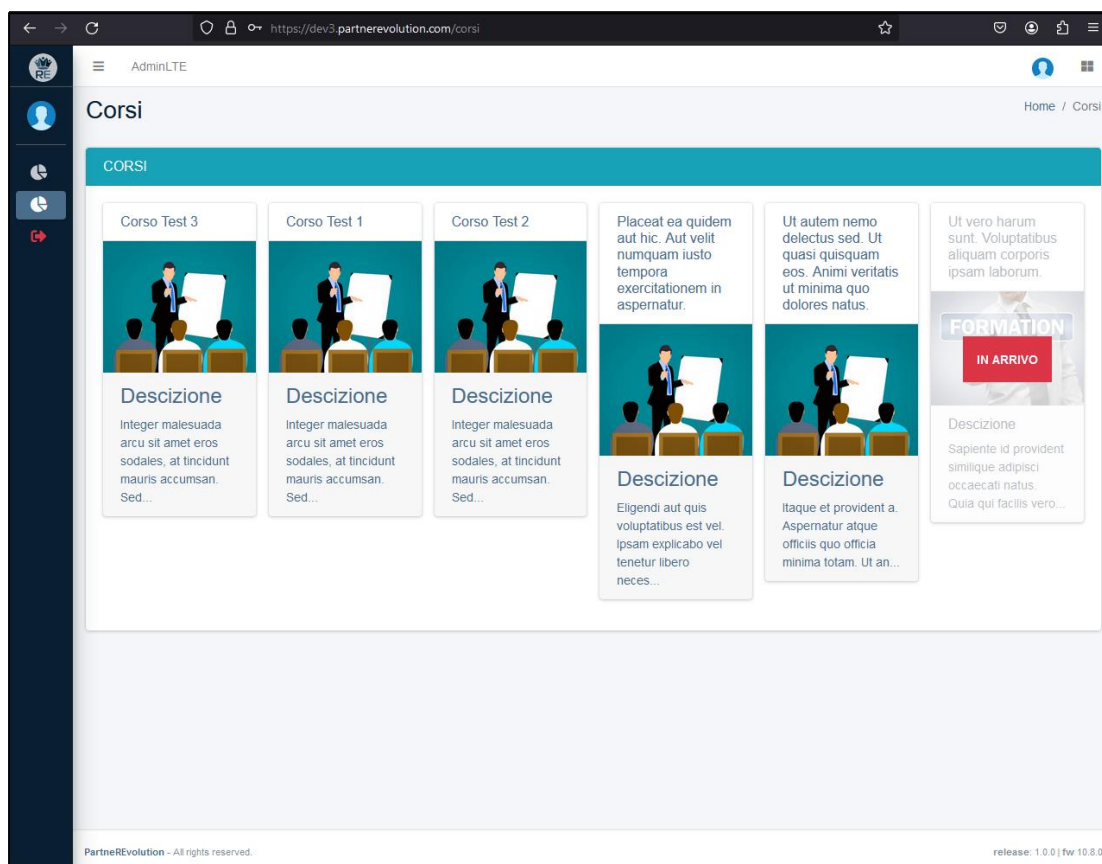


Figura 8 - Catalogo Corsi

Il catalogo dei corsi si adatta dinamicamente all'utente, differenziando la visualizzazione a seconda che questo abbia o meno corsi attivi. All'apertura del catalogo (vedi figura 8), in alto a sinistra, compare il titolo "Corsi", mentre a destra si trova il breadcrumb, che guida l'utente attraverso la struttura del sito.

Se l'utente non ha ancora attivato corsi, da come possiamo vedere in figura, si troverà di fronte a una serie di proposte formative non attive. Ciascun corso è rappresentato come una "carta", che mostra il titolo del corso, una figura illustrativa (o un'immagine predefinita) e una breve descrizione che, per motivi di spazio, termina con tre puntini. Un dettaglio interessante è che alcuni corsi sono presentati in una tonalità grigiastra e sono contrassegnati con un distintivo rosso recante la scritta "in arrivo", segnalando così le future opportunità formative.

Per gli utenti che hanno già iniziato il loro percorso formativo e attivato corsi, la pagina del catalogo, come si può osservare in figura 9, si arricchisce ulteriormente con

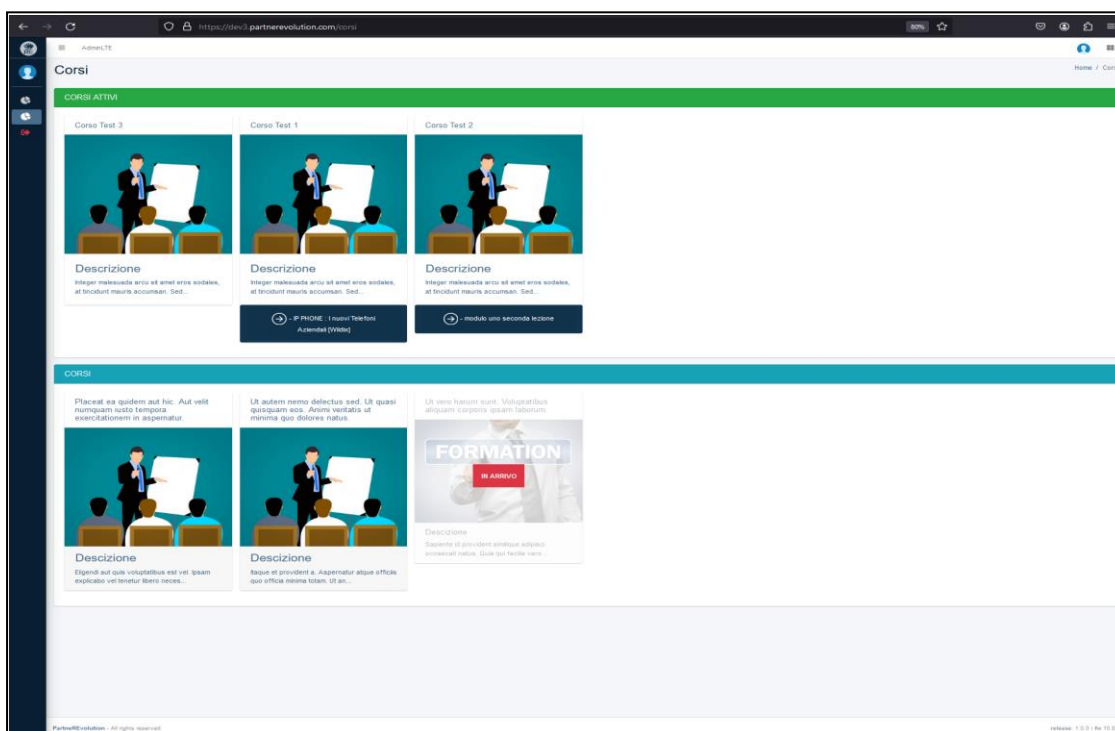


Figura 9 - Interfaccia grafica pagina "Catalogo dei corsi" di un utente con corsi attivi

nuovi elementi. Prima dell'elenco completo dei corsi, infatti, spicca un elenco dedicato

esclusivamente ai "CORSI ATTIVI", contrassegnato in verde chiaro. Questi corsi attivi, presentati con lo stesso stile grafico dell'elenco generale, offrono una funzione aggiuntiva: un bottone che segnala l'ultimo modulo che l'utente ha visualizzato. Cliccando su questo bottone, l'utente può tornare direttamente all'ultima lezione che stava seguendo, ottimizzando il proprio tempo e vivendo un'esperienza di apprendimento fluida e senza interruzioni. Va sottolineato che, per evitare ridondanze, i corsi attivi non vengono riproposti nell'elenco generale, rendendo così la navigazione più chiara e intuitiva.

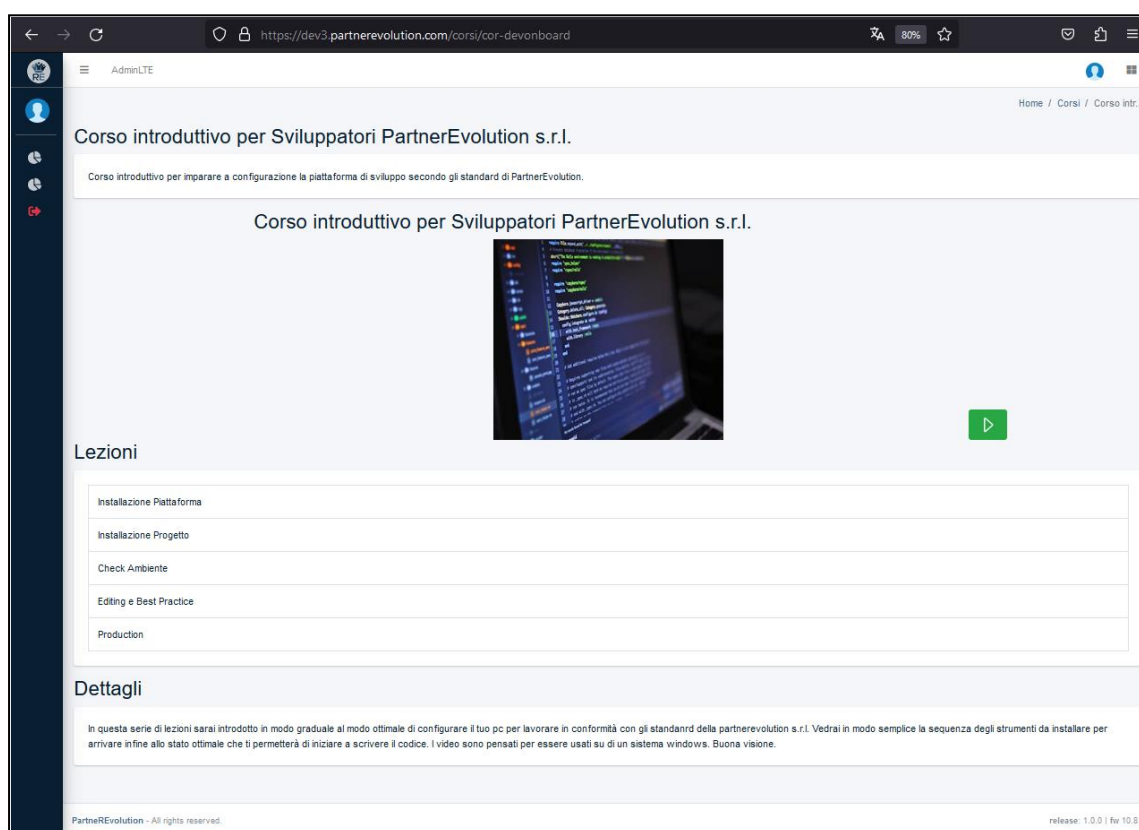


Figura 10 - Interfaccia grafica pagina "Lezioni"

Dopo aver esplorato la pagina “Catalogo dei Corsi”, cliccando su un corso specifico l'utente viene reindirizzato alla pagina “Lezioni”. Questa pagina è il cuore dell'esperienza formativa, dove ogni dettaglio è stato pensato per facilitare l'apprendimento e rendere l'interazione piacevole. Nella parte superiore della pagina, riportata in figura 10, l'occhio cade immediatamente sul titolo del corso, posizionato in alto a sinistra, che serve come punto di riferimento durante la navigazione. A destra, si trova il percorso di navigazione, o breadcrumb, che garantisce all'utente di sapere sempre dove si trova e come tornare indietro. Appena sotto, l'utente può leggere una descrizione breve del corso, ideata per offrire una panoramica immediata degli obiettivi e dei contenuti del corso. Al centro della pagina troviamo la sezione multimediale, come mostrato in figura 11. Quest' ultima è dinamicamente adattabile,



Figura 11 - Sezione multimediale della pagina "Lezioni"

permettendo all'utente, al suo ingresso nella pagina, di visualizzare un video promozionale o osservare un'immagine che caratterizza il corso, per poi interagire con i contenuti multimediali o testuali offerti. Per ogni modulo in visualizzazione, il suo titolo è posizionato sopra al contenuto effettivo, che potrebbe essere un video, un documento in formato PDF, un link di download o un collegamento a risorse esterne. Per facilitare la navigazione tra i contenuti, sono stati inseriti dei bottoni laterali che consentono all'utente di passare facilmente al modulo precedente o a quello successivo, garantendo una fluidità nell'esperienza di apprendimento.

Successivamente, l'utente incontra la sezione dedicata alle lezioni, come mostrato in figura 12. Questa sezione si presenta come una tabella verticale, con una colonna e

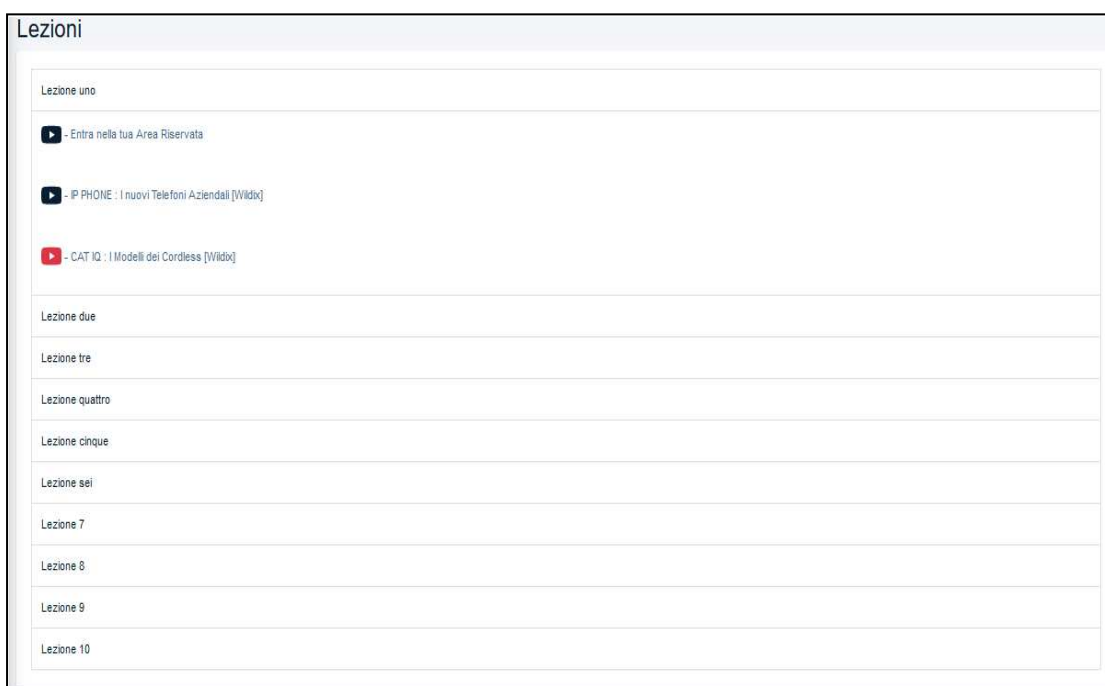


Figura 12 - Sezione elenco delle lezioni della pagina "Lezioni"

tante righe quante sono le lezioni. Ciascuna riga, o lezione, è interattiva: cliccandola,

si espande una sotto-tabella che elenca i moduli disponibili per quella specifica lezione. Ogni modulo è distintamente rappresentato da un'icona che indica il tipo di contenuto (che sia un PDF, un video di YouTube, un link esterno o una presentazione PowerPoint). Una particolarità notevole è l'interazione delle icone. Quando un modulo è in corso di visualizzazione, la sua icona diventa rossa, permettendo all'utente di identificare rapidamente dove si trova nel suo percorso formativo.

The screenshot shows a web interface for a course titled "Corso Test 3". At the top, there is a header with the course name and a block of placeholder text. Below this, a section titled "- modulo tre prima lezione" contains a blue link "Clicca qui per scaricare la presentazione" with a download icon. A green arrow icon is positioned to the left of the "Lezioni" section header. The "Lezioni" section is a table with three main rows: "Lezione uno", "Lezione due", and "Lezione tre". Each row contains a list of modules with icons indicating their type (video, PDF, or presentation) and a "Lezione gratuita" label.

Lezioni	
Lezione uno	
- modulo uno prima lezione	Lezione gratuita
- modulo due prima lezione	Lezione gratuita
- modulo tre prima lezione	Lezione gratuita
Lezione due	
- modulo uno seconda lezione	
- modulo due seconda lezione	
- modulo tre seconda lezione	
Lezione tre	

Figura 13 - Pagina "Lezioni" di un corso non attivo

Gli utenti che visualizzano un corso non attivo, pur avendo la possibilità di vedere tutti i contenuti, le lezioni e i moduli, saranno limitati nell'interazione con essi. Si può osservare nella figura 13 che solo i moduli contrassegnati dalla scritta "Lezione gratuita" in verde saranno interattivi, permettendo un'anteprima del corso prima dell'eventuale attivazione completa.

Concludendo la navigazione della pagina, dopo la sezione delle lezioni, l'utente trova la sezione "Dettagli", che presenta una descrizione estesa del corso, permettendo all'utente di immergersi pienamente nel contesto e negli obiettivi dell'apprendimento proposto.

Dopo aver esplorato l'interfaccia utente, è essenziale comprendere come le azioni dell'utente vengano tradotte in interazioni con il back-end dell'applicazione. Questa traduzione avviene attraverso le "rotte", che in un'applicazione web definiscono la mappa delle possibili interazioni.

```
1 /* Rotte catalogo corsi */
2 Route::get('/corsi', [UserController::class, 'showCatalogCourses'])
3     ->middleware(['auth', 'verified'])->name('corsi');
4
5 /* Rotte lezioni */
6 Route::get('/corsi/{code_courses}/{code_modules?}', [UserController::class, 'showLessons'])
7     ->middleware(['auth', 'verified'])->name('lezioni');
8
9 Route::post('/corsi/saveclick', [UserController::class, 'saveClick'])
10    ->middleware(['auth', 'verified'])->name('lezioni.click');
```

Figura 14 - Rotte per il catalogo dei corsi e per le lezioni

Nella figura 14 sono rappresentate le rotte utilizzate nell'applicazione in esame. Abbiamo tre rotte chiave che agevolano la navigazione dell'utente. Per esempio, la

rotta `"/corsi"` è stata concepita per guidare l'utente attraverso l'elenco dei corsi disponibili. Una volta autenticato e verificato, l'utente può accedere senza problemi a questa pagina e navigare tra i diversi corsi offerti.

Tuttavia, il vero valore aggiunto emerge quando consideriamo rotte più personalizzate come `"/corsi/{code_courses}/{code_modules?}"`. Questa rotta non solo tiene conto del corso specifico scelto dall'utente, ma offre anche la possibilità di visualizzare direttamente un modulo specifico, grazie al parametro opzionale `"{code_modules?}"`. Questo parametro, pensato per migliorare l'esperienza dell'utente, rappresenta una caratteristica user-friendly molto apprezzata. Si immagina di poter tornare direttamente all'ultimo modulo visualizzato con un semplice click, evitando così di dover ricercare manualmente il punto in cui si era interrotta la navigazione. In aggiunta, le interazioni dell'utente non vengono trascurate. La rotta `"/corsi/saveclick"` è stata specificamente ideata per tracciare ogni click dell'utente su un modulo, garantendo così un monitoraggio accurato delle attività dell'utente e il salvataggio dell'ultimo modulo visualizzato. Le rotte agiscono come ponte tra le scelte dell'utente nell'interfaccia e le azioni eseguite dal back-end dell'applicazione. Come vedremo, il Controller è l'elemento chiave che interpreta e gestisce queste azioni. Una volta che l'utente interagisce con l'interfaccia, le azioni compiute vengono tradotte in specifiche richieste al server, che risponde attraverso il Controller. In questa applicazione, lo `"UserController"` rappresenta il centro della gestione delle richieste inerenti all'utente e ai corsi. Si occupa di presentare i corsi, le lezioni, i moduli, e di registrare le interazioni dell'utente con i contenuti. Nel framework Laravel, un Controller è spesso responsabile di definire la logica di business dell'applicazione e di interagire con i modelli, estraendo o manipolando i dati, prima di presentarli all'utente attraverso una vista. Nella figura 15 sono mostrati i metodi dello `"UserController"`.


```
1 /**
2  * Mostra il catalogo dei corsi all'utente.
3  * Il metodo estrae tutti i corsi disponibili, escludendo quelli già attivati dall'utente,
4  * e li invia alla vista 'catalogCourses' per la visualizzazione.
5  */
6 public function showCatalogCourses() {
7     $user_id = Auth::user()->code_users;
8     return view('catalogCourses')
9         ->with('catalogCourses', $this->catalogModel->getAllCourses()->diff($this-
10         >catalogModel->getActivatedCourses($user_id)))
11         ->with('activetedCourses', $this->catalogModel-
12         >getActivatedCourses(Auth::user()->code_users));
13 }
14 /**
15  * Mostra le lezioni e i moduli di un corso specificato dal suo codice.
16  * Se specificato, mostra anche l'ultimo modulo visualizzato dall'utente per quel corso.
17  */
18 public function showLessons($code_courses, $code_lastViewModules = null) {
19     $user_id = Auth::user()->code_users;
20     return view('lessons')
21         ->with('lessons', $this->catalogModel->getLessonsByCourse($code_courses))
22         ->with('modules', $this->catalogModel->getModulesByCourse($code_courses))
23         ->with('course', $this->catalogModel->getCourseById($code_courses))
24         ->with('flag', $this->catalogModel->getCourseIsActive($code_courses,
25         $user_id))
26         ->with('lastmodule', $this->catalogModel-
27         >getModuleById($code_lastViewModules));
28 }
29 /**
30  * Registra un click dell'utente su un modulo.
31  * Se l'utente ha accesso attivo al corso a cui appartiene il modulo,
32  * viene registrato un log di visualizzazione per quel modulo.
33  */
34 public function saveClick(Request $request) {
35     $user_id = Auth::user()->code_users;
36     return response()->json($this->catalogModel
37         ->createClick($request->input('id'), $user_id));
38 }
```

Figura 15 - Metodi dello UserController

All'interno del “UserController”, il metodo “showCatalogCourses”, nella riga 6, ha un ruolo centrale nella presentazione dei corsi all'utente. Quando invocato, estrae tutti i corsi disponibili, ma con una particolare attenzione: esclude quei corsi che l'utente ha già attivato. In pratica, questo significa che l'utente vedrà solamente i corsi che può ancora attivare o acquistare, evitando duplicati o informazioni non pertinenti.

Procedendo, troviamo il metodo “showLessons”, nella riga 17: esso fornisce una panoramica dettagliata dei contenuti di un corso specifico. Quando un utente decide di esplorare un corso dal catalogo, è proprio questo metodo che permette ciò. Esso non si limita a presentare le lezioni e i moduli relativi a quel corso, ma tiene conto di alcuni dettagli importanti per migliorare il controllo e l'esperienza dell'utente.

Infatti, il metodo passa la variabile “\$flag”, nella riga 23, alla vista “lessons” per comunicarle se il corso in questione è attivo o meno per l'utente. Questa distinzione è fondamentale, poiché la vista, in base a questa variabile, determina se mostrare all'utente tutti i contenuti del corso oppure limitarsi a presentare solo quelli gratuiti.

Un altro aspetto rilevante del metodo “showLessons” è la sua capacità di tenere traccia dell'ultimo modulo visualizzato dall'utente. Se al metodo viene passato un valore, diverso da null ed uguale al codice identificativo di un modulo esistente, attraverso il parametro “\$code_lastViewModule”, allora la funzione presenterà, in aggiunta ai contenuti del corso, anche l'ultimo modulo che l'utente aveva esplorato attraverso la variabile “lastmodule”. Questa caratteristica è stata progettata con l'intento di offrire all'utente la possibilità di riprendere il suo percorso di studio esattamente da dove si era interrotto, garantendo una continuità nell'apprendimento.

Infine, il metodo “saveClick” riveste un'importanza strategica nella raccolta di dati sull'interazione dell'utente. Ogni volta che un utente clicca su un modulo, l'azione viene registrata, attraverso questo metodo, nella tabella “log_view_modules”. Ciò è

fondamentale non solo per tracciare i progressi dell'utente, ma anche per raccogliere dati utili per eventuali analisi o miglioramenti futuri dell'interfaccia e dei contenuti.

Dalla precedente discussione sull'architettura del controller, abbiamo evidenziato come il nostro sistema interagisca strettamente con vari modelli. Il modello `Catalog` emerge come uno strumento fondamentale, fungendo da ponte tra le risorse e l'interfaccia utente. Esso facilita operazioni come il recupero e l'aggregazione dei dati. Esaminiamo più da vicino la funzione "getAllCourses", utilizzata dal controller nel metodo "showCatalogCourses" per ottenere tutti i corsi, illustrata nella figura 16.

```
1 public function getAllCourses() {
2     // Recupera tutti i corsi in anteprima
3     $coursePre = $this->courseModel->getAllCoursesPre();
4
5     // Recupera corsi che sono sia "disabilitati" che in "bozza"
6     $courseDisBozz = $this->courseModel->getAllCoursesDisBozz();
7
8     // Ottiene tutti i corsi e rimuove quelli in "bozza", "disabilitati" e "anteprima"
9     $allCourse = $this->courseModel->getAllCourses()->diff($courseDisBozz-
10     >concat($coursePre));
11
12     // Restituisce l'elenco completo dei corsi, mettendo i corsi in anteprima alla fine
13     return $allCourse->concat($coursePre);
14 }
```

Figura 16 - Metodo "getAllCourses()" del modello Catalog

Nel contesto del sistema, la funzione gioca un ruolo chiave nel fornire un elenco completo dei corsi disponibili, organizzandoli in un ordine specifico. Qui, i corsi sono inizialmente segmentati basandosi sul loro stato: se sono in "anteprima" o sono "disabilitati/bozza". La logica del codice si assicura che tutti i corsi, tranne quelli che sono "disabilitati" o in "bozza", siano recuperati e che successivamente, i corsi in "anteprima" vengano aggiunti alla fine dell'elenco. Questa organizzazione può essere utile per mostrare i corsi in modo che gli utenti vedano prima i corsi generalmente disponibili, seguiti da quelli in "anteprima".

Esaminiamo più da vicino la funzione "getCourseById", mostrata in figura 17, che recupera dettagli specifici di un corso utilizzando l'ID fornito. Oltre a fornire informazioni basilari sul corso, si occupa anche di gestire i moduli video promozionali associati. Nel caso in cui un corso disponga di un modulo video promozionale, la funzione recupera i dettagli di quel modulo. Se tale modulo è identificato come un video di YouTube, dunque il suo attributo "type" ha valore "video_embedded", la funzione estrae l'ID del video dal relativo URL, consentendo così una facile integrazione del video all'interno dell'interfaccia utente.

```
1 public function getCourseById($code_courses){
2     // Recupera le informazioni del corso tramite l'ID fornito.
3     $course = $this->courseModel->getCourseById($code_courses);
4
5     // Controlla se il corso ha un modulo video promozionale associato.
6     if($course->promo_code_modules != null){
7
8         // Recupera le informazioni del modulo tramite l'ID del modulo promozionale.
9         $link = $this->moduleModel->getModuleById($course->promo_code_modules);
10
11        // Se il modulo è presente e di tipo 'video_embedded' (un video YouTube nel nostro
12        caso).
13        if($link != null && $link->type == 'video_embedded'){
14            // Estrae l'ID del video da un URL completo di YouTube usando un'espressione
15            regolare.
16            if(preg_match("#(?<=v=)[a-zA-Z0-9-]+(?:=&)|(?<=v\|/)[^&\n]+(?:=?\?)|(?<=v=)[^&\n]+|
17            (?<=youtu.be/)[^&\n]+#", $link->content_type, $matches)){
18                $link->content_type = $matches[0];
19            }
20        }
21
22        // Associa l'ID del video e il tipo di modulo al corso.
23        $course->setAttribute('link', $link->content_type);
24        $course->setAttribute('type', $link->type);
25    }
26    return $course;
27 }
```

Figura 17 - Metodo "getCourseById" di "Catalog"

Proseguendo con l'analisi della classe “Catalog”, ci imbattiamo in altre funzioni importanti che meritano attenzione. La funzione “getActivatedCourses”, rappresentata in figura 18, rappresenta uno degli elementi chiave della classe.

```
1 public function getActivatedCourses($code_users) {
2     // Ottiene l'elenco degli ID dei corsi attivati per un utente specifico.
3     $courseIdList = $this->activeCourseModel->getCoursesByUser($code_users);
4     $actCourseList = new Collection;
5
6     foreach ($courseIdList as $courseId) {
7         // Recupera le informazioni dettagliate del corso utilizzando l'ID.
8         $actCourse = $this->courseModel->getCourseById($courseId);
9
10        // Cerca l'ultimo modulo visualizzato dall'utente per quel corso specifico.
11        $lastLogViewModule = $this->logviewmoduleModel
12            ->where([
13                ['code_users', '=', $code_users],
14                ['code_courses', '=', $courseId],
15            ])->orderBy('timestamp_view', 'DESC')->first();
16
17        if(isset($lastLogViewModule)){
18            // Recupera i dettagli del modulo.
19            $lastModule = $this->moduleModel->getModuleById($lastLogViewModule
20                ->code_modules);
21            // Associa al corso i dettagli sull'ultimo modulo visualizzato.
22            $actCourse->setAttribute('code_lastModuleView', $lastModule->code_modules);
23            $actCourse->setAttribute('title_lastModuleView', $lastModule->title);
24        }
25        // Aggiunge il corso alla collezione.
26        $actCourseList->push($actCourse);
27    }
28
29    return $actCourseList->sortBy('id_order');
30 }
31
```

Figura 18 - Metodo "getActivatedCourses" di "Catalog"

Il suo compito principale è di recuperare l'elenco dei corsi attivi associati a un utente specificato tramite il parametro “\$code_users”. Il processo inizia interagendo con il modello “activeCourseModel”, utilizzando il metodo “getCoursesByUser” per ottenere gli ID dei corsi che sono stati attivati per l'utente in questione.

Una volta ottenuti gli ID, si procede con la creazione di una nuova collezione, denominata “\$actCourseList”, che avrà il compito di contenere i dettagli completi di ogni corso.

Il metodo, quindi, scorre l'elenco degli ID dei corsi con un ciclo for-each. Per ogni ID, recupera le informazioni dettagliate del corso corrispondente tramite il metodo “getCourseById” del modello “courseModel”.

Ma non si limita a questo: per ogni corso, la funzione cerca di determinare qual è stato l'ultimo modulo visualizzato dall'utente. Questo viene fatto attraverso una query al modello “logviewmoduleModel”, che ordina i moduli visualizzati in base al timestamp e seleziona il più recente.

Se viene identificato un modulo visualizzato di recente, la funzione estrae ulteriori dettagli sul modulo tramite il metodo “getModuleById” del modello “moduleModel”. Questi dettagli vengono poi associati al corso, in modo che ogni corso attivo nell'elenco contenga informazioni sull'ultimo modulo consultato dall'utente.

Infine, il corso viene aggiunto alla collezione “\$actCourseList”, e alla fine del ciclo, la collezione completa viene ordinata in base all'attributo “id_order” e restituita come risultato della funzione. In questo modo, la funzione fornisce non solo l'elenco dei corsi attivi dell'utente, ma anche informazioni contestuali su dove l'utente si era fermato, permettendo un'esperienza di apprendimento fluida e senza interruzioni.

La funzione “getModulesByCourse”, come possiamo vedere in figura 19, è un elemento centrale nella gestione dei corsi, con uno specifico focus sul recupero dei moduli ad essi associati.

```
1 public function getModulesByCourse($code_course){
2     // Recupera le lezioni associate a un determinato corso.
3     $lessons = $this->getLessonsByCourse($code_course);
4     $allModules = new Collection;
5
6     foreach($lessons as $lesson){
7         // Per ogni lezione, recupera i moduli associati.
8         $modules = $this->lessonModel->find($lesson->code_lessons)->modules;
9
10        // Estrae l'ID del video da un URL completo di YouTube per i moduli di tipo
11        'video_embedded'.
12        foreach($modules as $module){
13            if($module->type == 'video_embedded'){
14                if(preg_match("#(?<=v=)[a-zA-Z0-9-]+(?:=&)|(?:<=v\/)[^\n]+(?:=?\?)|(?:<=v=)
15                [^\n]+(?:<=youtu.be\/)[^\n]+#", $module->content_type, $matches)){
16                    $module->content_type = $matches[0];
17                }
18            }
19        }
20        $allModules = $allModules->concat($modules);
21    }
22    return $allModules->sortBy('id_order');
23 }
24
```

Figura 19 - Metodo "getModulesByCourse" di "Catalog"

Il processo inizia invocando il metodo “getLessonsByCourse”, che, come suggerisce il nome, è incaricato di recuperare tutte le lezioni relative a un corso specificato dal parametro “\$code_course”. Queste lezioni vengono memorizzate nella variabile “\$lessons”.

Un'ulteriore elaborazione viene eseguita sui moduli di tipo `video_embedded`. In particolare, se un modulo ha questo tipo, la funzione tenta di estrarre l'ID del video incorporato da un URL completo di YouTube, utilizzando un'espressione regolare. Questo passaggio è fondamentale per ridurre l'URL completo a un ID video,

semplificando e ottimizzando ulteriori operazioni o integrazioni che potrebbero aver bisogno solo dell'ID.

Dopo aver eseguito queste operazioni su tutti i moduli di ogni lezione, tutti vengono raccolti in una collezione globale denominata “\$allModules”. Questa collezione viene infine ordinata in base all'attributo “id_order” e restituita.

Una particolare attenzione va alla riga 8 della figura 19: “\$modules = \$this->lessonModel->find(\$lesson->code_lessons)->modules;”. Questa singola riga rappresenta l'integrazione di Laravel con il modello dati, ed utilizza la potenza delle relazioni Eloquent per accedere direttamente ai moduli associati a una specifica lezione. Il metodo “find” recupera una lezione basandosi sul suo “code_lessons”, e subito dopo, accedendo alla proprietà “modules”, Laravel sfrutta la relazione “HasMany”, come si può vedere in figura 20, definita nel modello `Lesson` per recuperare tutti i moduli correlati.

```
1 public function modules(): HasMany
2 {
3     return $this->hasMany(Module::class, 'code_lessons', 'code_lessons');
4 }
5
```

Figura 20 - Relazione "HasMany" del modello "Model"

Dunque, viene definita una relazione "uno a molti" tra una “Lesson” e vari “Module”. La funzione “hasMany” stabilisce questa relazione, indicando che una singola lezione può avere molti moduli associati. I parametri passati alla funzione specificano la classe del modello correlato (in questo caso, “Module::class”) e le chiavi esterne che stabiliscono il collegamento tra i due modelli.

Questo flusso semplificato dimostra quanto sia efficiente e intuitiva la gestione delle relazioni in Laravel. Attraverso poche righe di codice, si riesce a navigare attraverso le relazioni tra i modelli e ad accedere a dati correlati, rendendo le operazioni di recupero dei dati notevolmente più concise e dirette.

In sintesi, il modello “Catalog.php” serve come strumento cruciale nell'architettura dell'applicazione. Esso consolida le diverse risorse e le relazioni tra di esse, permettendo un flusso di dati fluido e organizzato.

Dopo aver delineato le fondamenta dell'applicazione attraverso l'interfaccia utente, le rotte, i controller e i modelli, ci spostiamo verso la creazione di un'esperienza utente dinamica. A tal proposito, jQuery e AJAX giocano un ruolo cruciale.

La vista dedicata alla pagina "Lezioni" applica queste tecnologie attraverso una serie di funzioni definite in un file .js ad essa collegato, che la arricchisce di dinamicità e interattività, in modo da ottenere rapidi aggiornamenti dell'interfaccia senza la necessità di ricaricare completamente la pagina. Questo file JavaScript comprende una serie di funzioni che, agendo in sinergia, contribuiscono a rendere l'esperienza dell'utente fluida e intuitiva. Di seguito vengono descritte le principali funzioni presenti nel file.

La funzione “caricamentoPage”, illustrata in figura 21, si preoccupa di ottimizzare l'esperienza utente, assicurandosi che gli utenti possano riprendere da dove avevano interrotto durante una precedente visita alla pagina. La logica all'interno della funzione è mirata a due possibili scenari.

Nel primo scenario, la variabile “codeModules” è stata precedentemente impostata, il che deriva dal fatto che l'utente ha cliccato il bottone "Ultima visualizzazione" di un corso, nella pagina del catalogo corsi.

Quando ciò si verifica, la funzione simula clic sugli elementi corrispondenti ai moduli e alle lezioni, facendo sì che l'utente ritorni esattamente al punto in cui si era interrotto.

```
1 function caricamentoPage() {
2   // La variabile codeModules è settata solo se l'utente nella pagina del catalogo corsi
   ha cliccato il bottone "Ultima visualizzazione"
3   if (typeof(codeModules) !== "undefined" && codeModules !== null) {
4     $("#" + codeModules).click();
5     $("#" + codeLessons).click();
6   } else {
7     // Altrimenti rende interattivo il bottone "successivo" e ne setta il valore =
       code_modules primo modulo
8     $("#playLeft").addClass("d-none");
9     if($(".linkClick").length) $("#playRight").val($(".linkClick").eq(0).attr('id'));
10    else $("#playRight").addClass("d-none");
11  }
12 }
```

Figura 21 - Codice js

Nel secondo scenario, quando la variabile “codeModules” non è definita o è null, l'utente non ha precedentemente selezionato "Ultima visualizzazione" o è la sua prima visita alla pagina. In questa circostanza, il bottone "precedente" viene nascosto e il bottone "successivo" viene impostato per fare riferimento al primo modulo disponibile nella lista. Se non ci sono moduli disponibili, il bottone "successivo" viene anch'esso nascosto.

La funzione “listenerClick()”, mostrata in figura 22, è stata progettata per rispondere alle interazioni specifiche dell'utente con i moduli delle lezioni di un corso. Essa enfatizza due meccanismi chiave di interazione.

```
1 function listnerClick() {
2
3   // Aggiunge un gestore di eventi ai bottoni "playLeft" e "playRight" per gestire il
  click.
4   $("#playLeft, #playRight").click(function(e) {
5     e.preventDefault(); // Previene l'azione di default del click.
6
7     // Utilizza il valore del bottone (che rappresenta l'ID del modulo) per identificare
  // il modulo successivo o precedente
  // e simula un click su di esso.
8     var idModulo = $(this).val();
9     $("#" + idModulo).click();
10
11
12    // Se il modulo target (precedente o successivo) è all'interno di una riga nascosta
  (lezione chiusa)...
13    if($("#" + idModulo).closest('tr').hasClass('d-none')) {
14      // ... chiude la lezione attualmente aperta ...
15      $('tr[aria-expanded="true"]').click();
16      // ... e apre la lezione che contiene il modulo target.
17      $("#" + idModulo).closest('tr').prev().click();
18    }
19  });
20
21  // Aggiunge un gestore di eventi ai link con classe "linkClick" per gestire il click.
22  $(".linkClick").click(function(e) {
23    e.preventDefault(); // Previene l'azione di default del click.
24
25    // Invia una richiesta AJAX utilizzando vari attributi dell'elemento cliccato.
26    sendData(
27      $(this).attr('id'),
28      $(this).attr('href'),
29      $(this).prev(),
30      $(this).attr('class').split(' ')[0],
31      $(this).text(),
32      $(".linkClick").index(this)
33    );
34
35    // Quando un link è cliccato, la pagina effettua uno scroll animato per centrare la
  specifica sezione multimediale(focus).
36    $('html, body').animate({
37      scrollTop: $("#focus").offset().top
38    }, 1000);
39  });
40
41 }
42
```

Figura 22 - Codice js

Il primo riguarda la navigazione tra moduli. Quando un utente interagisce con i bottoni identificati come “#playLeft” o “#playRight”, il sistema identifica il modulo appropriato basandosi sull'ID univoco associato a questi bottoni. Una volta riconosciuto il modulo desiderato, un'azione simulata di click viene effettuata su di esso. Se il modulo selezionato risiede in una sezione di lezione attualmente nascosta, la funzione garantisce che la sezione venga resa visibile, offrendo all'utente una navigazione ininterrotta e contestualizzata.

Il secondo meccanismo si attiva durante l'interazione con specifici collegamenti, distinti dalla classe “.linkClick”. Su un tale intervento, la funzione non solo previene il comportamento predefinito del link, ma invoca anche una richiesta AJAX. Tale richiesta potrebbe avere molteplici finalità, come il recupero di contenuti aggiuntivi o la registrazione di determinate azioni dell'utente. Dopo questa operazione, l'utente viene indirizzato, attraverso uno scroll animato, verso una sezione denominata “#focus”, ovvero la sezione multimediale della pagina “Lezioni”, assicurando che l'attenzione venga rivolta immediatamente al contenuto del modulo.

Nel suo complesso, “listnerClick()” rappresenta uno strumento di navigazione e interazione, progettato per arricchire e semplificare l'esperienza utente.

Infine, la funzione “sendData”, riportata in figura 23, è un componente fondamentale per l'elaborazione interattiva e dinamica dei dati tra il frontend e il backend. Essa utilizza una richiesta AJAX per comunicare con il server, inviando dettagli rilevanti associati ai moduli con cui l'utente ha interagito o che ha visualizzato. Questa richiesta produce una risposta in formato JSON.

```
1 //effettua una richiesta AJAX in formato JSON al server per inviare dati, gestisce la
  risposta del server e esegue azioni specifiche in base al tipo di modulo specificato.
2
3 function sendData(code_modules, link, prec, typeModule, title, index) {
4
5     $.ajaxSetup({
6         headers: {
7             'X-CSRF-TOKEN': $('meta[name="csrf-token"]').attr('content')
8         }
9     });
10
11     $.ajax({
12         url: addRoute,
13         type: 'POST',
14         data: JSON.stringify({ id: code_modules }),
15         dataType: 'json',
16         contentType: 'application/json',
17         success: function(response) {
18             console.log(response);
19
20             //esegue lo switch o assegna il colore rosso alle icone per indicare quella
              corrente
21
22             toggleActive(prec);
23
24             //cambia il titolo del modulo corrente
25
26             changeTitle(title);
27
28             //definisce il valore dei bottoni precedente e successivo in base alla posizione
              del modulo(casi estremi prima e ultima)
29
30             definePreNext(index);
31
32             // in base al tipo di contenuto presente nel modulo, verrà eseguito il relativo
              processo richiesto
33             // (PDF, immagini, video .mp4, video integrati da YouTube e presentazioni .ppt)
              per controllare e, se necessario,
34             //modificare la struttura HTML al fine di garantire il supporto adeguato per i
              contenuti.
35
36             switch(typeModule){
37                 case'needIframe': needIframe(link);
38                 break;
39                 case'needVideo': needVideo(link);
40                 break;
41                 case'needAnchor': needAnchor(link);
42                 break;
43                 case'needButton': needButton(link);
44                 break;
45             }
46
47         },
48         error: function(error) {
49             console.log(error);
50         }
51     });
52 }
53
```

Figura 23 - Richiesta Ajax

Alla ricezione di una risposta positiva dal server, la funzione entra in azione, orchestrando una serie di operazioni destinate a migliorare l'esperienza dell'utente finale. Queste operazioni comprendono l'aggiornamento dinamico degli elementi visivi, come l'attivazione o la disattivazione di icone, la modifica dei titoli o l'aggiornamento dei pulsanti di navigazione a seconda della posizione attuale del modulo.

Oltre a ciò, “sendData” si assicura di personalizzare l'interfaccia in base al tipo di contenuto del modulo, che può variare tra diversi formati come video, immagini, documenti PDF e altro. Questo viene realizzato attraverso l'istruzione PHP switch che, a seconda del tipo di modulo, invoca diverse funzioni, ognuna delle quali è dedicata alla gestione e presentazione di un particolare tipo di contenuto. In questo modo, la funzione assicura che ogni tipo di contenuto sia non solo visualizzato, ma anche ottimizzato per l'interazione e la comprensione dell'utente.

All'interno della “success”, è presente “definePreNext(indexLinkClick)” che opera come un pilota di navigazione: inizializza i pulsanti di navigazione “precedente” e “successivo” dandogli come valore l'identificatore dei relativi moduli e, decide quali mostrare e quali nascondere, gestendo così l'esperienza dell'utente durante la navigazione tra i contenuti.

D'altra parte, “changeTitle(title)” ha il compito specifico di aggiornare il titolo della sezione multimediale, prendendo il titolo fornito come parametro e rendendolo visibile all'utente, mentre le funzioni “needIframe(link)”, “needVideo(link)”, “needButton(link)” e “needAnchor(link)” si occupano di manipolare la sezione multimediale in risposta alle esigenze dei contenuti in visualizzazione. Ad esempio, “needIframe” integra un iframe nella sezione, basandosi sull'URL fornito, “needVideo” fa lo stesso con un elemento video, mentre “needButton” introduce un

pulsante intuitivo per il download, accompagnato da un'icona, e infine, “needAnchor” implementa un link esterno, segnalato da un'icona specifica. Questa serie di funzioni assicura che la piattaforma offra un'esperienza d'uso flessibile e user-friendly, adeguandosi dinamicamente ai diversi tipi di contenuti.

In sintesi, l'uso combinato di jQuery e AJAX fornisce all'applicazione un livello di interattività e reattività che migliorano notevolmente l'esperienza complessiva dell'utente. Attraverso queste funzioni, l'applicazione è in grado di mantenere l'utente al centro dell'attenzione, garantendo che i contenuti siano presentati in modo fluido.

Capitolo 6

Conclusioni e sviluppi futuri

La realizzazione della sezione di accesso ai contenuti di una piattaforma e-learning per la società PartnerRevolution è stata una sfida ardua, nella quale ho maturato esperienza attraverso il confronto con il team di lavoro e rispondendo alle loro richieste. Questa sezione ha una finalità ben precisa: essere ceduta o venduta ad aziende che, integrandola nei loro siti internet, possono formare efficacemente il proprio personale. Ecco perché ogni aspetto del design, dall'interfaccia alle funzionalità, è stato pensato per soddisfare l'esigenza di avere un singolo amministratore che gestisce, in maniera intuitiva e centralizzata, operazioni come il caricamento, la modifica, l'attivazione e la disattivazione dei corsi e dei loro contenuti.

La sua progettazione e implementazione ha richiesto una profonda comprensione delle necessità degli utenti, nonché delle sfide e delle opportunità del settore e-learning. L'applicazione sviluppata non si limita a fornire una soluzione per l'accesso ai contenuti formativi, ma punta a trasformare l'esperienza di apprendimento online in un percorso intuitivo, personalizzato e coinvolgente. Le due pagine principali, "Catalogo dei Corsi" e "Lezioni", servono come fulcro dell'esperienza utente, fornendo un accesso immediato e strutturato ai contenuti. L'integrazione di funzionalità chiave, come la personalizzazione dei corsi unito al monitoraggio dell'attività dell'utente, permette un'ampia flessibilità nella gestione dei contenuti. Infatti, con "personalizzazione" si intende la possibilità di disabilitare singole lezioni di corsi attivati a specifici utenti, per adattare il corso in base al profilo di quest'ultimo. Questa funzionalità è stata progettata anche per garantire che un unico corso possa essere modulato e reso adatto a diverse tipologie di utenti, senza la necessità di creare molteplici versioni del medesimo. Questo riflette l'ambizione del progetto di andare oltre la semplice presentazione dei contenuti, mirando a creare un ambiente di apprendimento che risponde in maniera attiva alle esigenze dell'utente. La sezione di accesso ai contenuti di una piattaforma e-learning sviluppata durante il tirocinio risponde appieno a molti degli obiettivi inizialmente delineati.

In futuro, sono previste diverse opportunità per ampliare e arricchire la sezione di accesso ai contenuti di una piattaforma di e-learning. Uno sviluppo potenziale riguarda la possibilità per gli utenti di richiedere l'attivazione di un corso attraverso un apposito bottone. Questa funzionalità avrebbe il duplice vantaggio di permettere agli utenti di esprimere interesse per particolari argomenti e, allo stesso tempo, fornire feedback sulla direzione futura del catalogo dei corsi.

Un'altra opzione futura riguarda l'introduzione di una pagina per un forum di discussione, che potrebbe fornire una piattaforma interattiva per gli utenti,

permettendo loro di scambiarsi idee, risolvere dubbi e consolidare una comunità d'apprendimento. Tale forum potrebbe potenziare l'apprendimento collaborativo e creare un ambiente d'istruzione ancor più coinvolgente e stimolante.

Infine, l'introduzione di esercizi di valutazione dell'apprendimento, o quiz, alla fine di ogni corso potrebbe rafforzare il processo di apprendimento. Questi esercizi fornirebbero una valutazione immediata sulle competenze acquisite, garantendo una comprensione approfondita degli argomenti trattati. Inoltre, potrebbero servire come strumento di validazione per attestare il completamento di un corso e l'effettiva acquisizione delle competenze proposte.

La combinazione di questi sviluppi con il solido lavoro già svolto potrebbe rendere la piattaforma e-learning di PartnerRevolution come un punto di riferimento nel panorama formativo online.

Bibliografia:

- [1] www.coursera.org - consultato a Luglio 2023
- [2] www.udemy.com - consultato a Luglio 2023
- [3] www.edx.org - consultato a Luglio 2023
- [4] it.khanacademy.org - consultato a Luglio 2023
- [5] <https://laravel.com/docs/10.x> - consultato ad Agosto 2023
- [6] laravel.com/docs/10.x/eloquent-serialization#main-content - consultato ad Agosto 2023
- [7] laravel.com/docs/10.x/routing#main-content - consultato ad Agosto 2023
- [8] laravel.com/docs/10.x/sail#main-content - consultato ad Agosto 2023
- [9] laravel.com/docs/10.x/sail#configuring-a-shell-alias - consultato ad Agosto 2023
- [10] docs.docker.com/get-started - consultato ad Agosto 2023
- [11] learn.microsoft.com/en-us/windows/wsl/ - consultato ad Agosto 2023
- [12] api.jquery.com/ - consultato ad Agosto 2023
- [13] api.jquery.com/category/ajax - consultato ad Agosto 2023
- [14] docs.github.com/en/get-started - consultato ad Agosto 2023