

Relazione del progetto di tecnologie internet

Zeudjio Tchiofo Simon: 310314

Carattin Daniel: 311340

**Indice**

1. Introduzione.………………………………………………………………………………………..…[3](#_bookmark0)

1.1 Contestualizzazione del progetto, scopo, obbiettivi e breve descrizione del contenuto………..…3

1. Metodologia…………………………………………………………………………………………...4

2.1. Descrizione delle tecnologie utilizzate……………………………………………………………………4

2.2. Integrazione per soddisfare le esigenze del progetto………………………………………………………5

1. Implementazione delle funzioni [6](#_bookmark4)

3.1 Utenti……………………………………………………………………………………………………6

3.1.1. HomeScreen…………………………………………………………………………………………..6

3.1.2. login…………………………………………………………………………………………………..7

3.1.3. Registrazione………………………………………………………………………………………….8

3.1.4. UserScreen……………………………………………………………………………………………9

3.1.5. AddtoCart……………………………………………………………………………………………10

3.1.6. CartScreen.…………………………………………………………………………………….……..11

3.1.7. Payment…………………………………………………………………………………….................12

3.1.8. GetOrders……………………………………………………………………………………………12

3.1.9. UserProfile………………………………………………………………………………….………..13

3.2 Admin……………………………………………………………………………………………….…14

3.2.1. AdminDashboard……………………………………………………………………………………14

3.2.1.1 UTENTI……………………………………………………………………………………………15

3.2.1.1.1 GetAllUsers………………………………………………………………………………………15

3.2.1.1.2 AddUser………………………………………………………………………………………….16

3.2.1.1.3AddAdmin………………………………………………………………………………………...16

3.2.1.1.4 GetUser…………………………………………………………………………………………..17

3.2.1.1.5 UpdateUser……………………………………………………………………………………….18

3.2.1.1.6 AdminOrders……………………………………………………………………………………..18

3.2.1.1.7 DeleteUser………………………………………………………………………………………..18

3.2.1.2 PRODOTTI...…...………………………………………………………………………………….19

3.2.1.2.1 GetAllProducts…………………………………………………………………………………....19

3.2.1.2.2 AddProduct…………………………………………………………………………………….…20

3.2.1.2.3 GetProduct………………………………………………………………………………………..20

3.2.1.2.4 UpdateProduct………………………………………………………………………………….....21

3.2.1.2.5 DeleteProduct…………………………………………………………………………………..…21

1. [Configurazione](#_bookmark25) 22
   1. [Database](#_bookmark26) 22
   2. [Server](#_bookmark27) 25
   3. [Client](#_bookmark28) 26
2. INTRODUZIONE
   1. CONTESTUALIZZAZIONE DEL PROGETTO, SCOPO, OBBIETTIVI E BREVE DESCRIZIONE DEL CONTENUTO

Il presente progetto rappresenta un’implementazione pratica di un’applicazione web per la gestione degli ordini online, più specificamente sulla vendita online di attrezzature per atletica leggera per la specialità dei lanci (lancio del disco, lancio del peso, lancio del giavellotto, lancio del martello e anche il lancio del vortex).

Lo scopo principale è fornire un’interfaccia utente intuitiva e una piattaforma affidabile per consentire agli utenti di visualizzare, aggiungere, modificare ed eliminare ordini e prodotti. Gli obiettivi del progetto includono la comprensione delle tecnologie di sviluppo web, come il set MERN (MongoDB, Express.js, React.js, Node.js), l’architettura del sistema e la pratica nell’implementazione di funzionalità di base, come l’autenticazione degli utenti e la gestione del database. La relazione fornirà una descrizione dettagliata del processo di sviluppo, inclusa l’architettura del sistema, la selezione e l’utilizzo delle tecnologie, l’implementazione pratica delle funzionalità, nonché considerazioni e riflessioni sulle sfide incontrate e le future direzioni del progetto.

1. METOLOGIE
   1. TECNOLOGIE UTILIZZATE E MOTOLOGIE DI SVILUPPO

Nel contesto del nostro progetto, abbiamo adattato una metodologia di sviluppo agile, che si basa su iterazioni rapide e feedback continuo per adattarsi ai cambiamenti dei requisiti e massimizzare il valore del progetto. Abbiamo scelto di utilizzare il set MERN (mongoDB, Express.js, React.js, Node.js) per il nostro stack tecnologico, poiché offre una serie di vantaggi in linea con le esigenze del nostro progetto.

**MongoDB:** Abbiamo optato per MongoDB come database NoSQL per la sua flessibilità e scalabilità. La struttura dei dati flessibile di MongoDB si adatta bene alle variazioni dei requisiti del progetto e consente una rapida iterazione nello sviluppo dei modelli dati.

**Express.js:** Abbiamo utilizzato Express.js come framework per lo sviluppo del back-end per la sua semplicità e flessibilità. Express.js semplifica la creazione di API RESTful, gestendo facilmente le richieste HTTP e semplificando il routing e la gestione degli endpoint.

**React.js:** Per il front-end, abbiamo scelto Rect.js per la sua modularità e facilità di sviluppo delle interfacce utente dinamiche e interattive. Con Rect.js, abbiamo potuto creare componenti riutilizzabili e gestire lo stato dell’applicazione in modo efficiente.

**Node.js:** Abbiamo utilizzato Node.js come ambiente di runtime per il nostro server back-end. Grazie alla sua architettura asincrona e non bloccante, Node.js è ideale per applicazioni ad alte prestazioni e scalabili.

**Ruolo di JavaScript:** È importante sottolineare che JavaScript è il linguaggio di programmazione principale utilizzato sia lato client che lato server nel nostro progetto. JavaScript è stato fondamentale per implementare la logica di front-end e back-end, consentendoci di sviluppare un’applicazione web full-stack coesa con coerenza tra front-end e back-end. La sua natura asincrona e non bloccante ha favorito lo sviluppo efficiente e scalabile del nostro progetto, consentendoci di gestire facilmente le operazioni di elaborazione e l’accesso ai dati in modo performante.

* 1. INTEGRAZIONE DELLE TECNOLOGIE

Il set MERN si integra in modo naturale, consentendo lo sviluppo di un’applicazione web full-stack coesa e con coerenza tra front-end e back-end. MongoDb funge da database di supporto per la sua persistenza dei dati, mentre Express.js fornisce un’interfaccia per la logica di business e l’esposizione delle API. React.js gestisce l’interfaccia utente dinamica lato client, interagendo con il back-end tramite chiamate API RESTful. Node.js agisce da middleware tra il front-end e il back-end, gestendo le richieste del client e orchestrando le operazioni di elaborazione e di accesso al database.

Dopo aver impostato il set MERN per lo sviluppo dell’applicazione web full-stack, è comune utilizzare **AXIOS** come libreria per gestire le richieste HTTP tra il client Ract.js e il server Node.js. Axios semplifica l’invio e la ricezione di dati dal back-end al front-end e viceversa, consentendo una comunicazione efficiente tramite chiamate API RESTful.

1. IMPLEMENTAZIONE DELLEFUNZIONI
   1. UTENTI

3.1.1 HomeScreen

HomeScreen è la schermata principale appena il programma viene avviato, ci viene mostrata una pagina dove un utente può far l’accesso ed è in comune sia per i clienti sia per gli amministratori.

Come si può notare contiene una Navbar dove un utente può cliccare su login o registrazione e un Footer dove abbiamo qualche informazione per quello che riguarda il sito in generale.

Immagine che contiene testo, schermata, diagramma, cerchio

Descrizione generata automaticamente

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

* + 1. LOGIN

Nel login abbiamo una form che permette sia agli utenti che agli amministratori di accedere. Per distinguere un utente normale da un amministratore abbiamo salvato bel database una variabile isAdmin che di default è false per un utente che fa la registrazione, e al momento del login in base a sé la variabile è true o false ti indirizza alla pagina appropriata.

Immagine che contiene testo, software, Software multimediale, Icona del computer

Descrizione generata automaticamente

Una volta su questa pagina se l’utente non ha mai fatto l’accesso può cliccare su crea il tuo account E-Shop e verrà indirizzato sulla pagina di registrazione.

Il login è stato implementato anche con la gestione degli errori.

Immagine che contiene testo, schermata, software, Pagina Web

Descrizione generata automaticamente

Immagine che contiene testo, schermata, software

Descrizione generata automaticamente

* + 1. REGISTRAZIONE

La Registrazione è riservata solo ai clienti perché come avevamo accennato pocanzi quando un utente viene creato la variabile isAdmin è di default fase. Solo un amministratore già presente nel sistema può assegnare un nuovo amministratore.

Immagine che contiene testo, schermata, software, design

Descrizione generata automaticamente

Immagine che contiene testo, schermata, software, design

Descrizione generata automaticamente

In questa pagina di registrazione c’è una gestione di errori sulla compilazione di ogni campo, sulla verifica che non si inserisca un indirizzo mail già esistente, sul fatto che la password non può assomigliare al nome, sulla lunghezza della password e sul fatto che debba aver lettere e numeri e ovviamente la Password deve essere uguale a Re-enter Password.

Una volta che la registrazione è avvenuta con successo ci riporta sulla pagina di login e possiamo fare l’accesso.

* + 1. UserScreen

Lo UserScreen è la pagina principale che appare quando l’utente fa il login con successo. Abbiamo una Navbar dove l’utente può visualizzare e modificare il proprio profilo, il carrello, gli ordini fatti e il logout e abbiamo anche la Home che praticamente ci porta alla pagina principale dell’utente.

Nello UserScreen abbiamo i prodotti su forma di paginazione e possiamo cliccare su singoli prodotti per la visualizzazione intera dell’immagine, possiamo aggiunge il prodotto al carrello grazie al button “Aggiungi al carrello “, c’è anche una barra di ricerca per vari prodotti.

Immagine che contiene testo, schermata, software

Descrizione generata automaticamente

Immagine che contiene testo, schermata, diagramma

Descrizione generata automaticamente

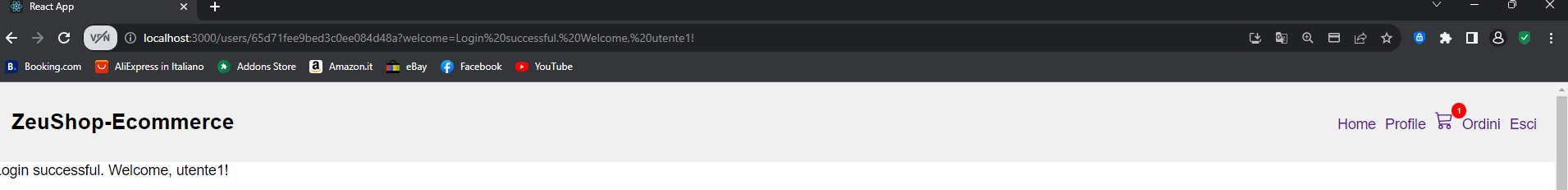
* + 1. AddtoCart

AddtoCart è la funzione che ci permette di aggiunge un prodotto al carrello, quando si aggiunge un prodotto al carrello esce un messaggio di successo da localhost e dopo il clic su ok la pagina si aggiorna.

Immagine che contiene schermata, testo, Software multimediale, software

Descrizione generata automaticamente

Poi la pagina viene aggiornata di conseguenza anche il carrello e ci viene mostrato la quantità di prodotti che abbiamo nel carrello.



* + 1. CartScreen

CartScreen è la pagina del carrello dove possiamo vedere tutti i prodotti che abbiamo aggiunto. Nel CartScreen possiamo aggiornare la quantità dei prodotti aggiunti al carrello, li possiamo anche rimuovere dal carrello e vedere la quantità disponibile.

Immagine che contiene testo, schermata, software, Carattere

Descrizione generata automaticamente

* + 1. Payment

Payment è la funzione per il processo del pagamento, abbiamo una form nella stessa pagina di CartScreen dove chiediamo l’indirizzo di consegna, la modalità di pagamento con la gestione di errore per ogni campo. Ad esempio, il numero di carta deve essere valido, la data di scadenza valida e anche il CVV.

Dopo aver compilato tutti i campi correttamente si procede all’acquista e ci appare un messaggio di successo che sparisce dopo qualche secondo.

Immagine che contiene testo, schermata, Carattere, software

Descrizione generata automaticamente

Quando il pagamento avviene con successo, il carrello si svuota e ovviamente la quantità di prodotti si aggiorna nel database.

* + 1. GetOrders

GetOrders ci permette di vedere i prodotti acquistati, la data e l’ora dell’acquisto ed eventualmente il giorno di consegna che abbiamo impostato a 72 ore dal pagamento però anche in questo caso abbiamo considerato i giorni non lavorativi. Se al calcolo dei tre giorni di consegna se il terzo giorno cade uno dei gironi non lavorativi impostati dal sistema si va avanti di un giorno e così via. Nella pagina degli ordini abbiamo anche una variabile ‘Stato’ salvato nel database che di default è “In attesa consegna” dopo i tre giorni lo stato cambia a “consegnato”.

Immagine che contiene testo, schermata, software, Pagina Web

Descrizione generata automaticamente

Come dicevamo pocanzi, in questo caso si può vedere come lo stato è passato da “In attesa di consegna” a “Consegnato”.

Immagine che contiene testo, schermata, software, Pagina Web

Descrizione generata automaticamente

Nella pagina di ordini abbiamo anche una funzione per eliminare gli ordini e chiede la conferma di eliminazione dell’ordine.

Immagine che contiene testo, Carattere, linea, software

Descrizione generata automaticamente

* + 1. UserProfile

UserProfile permette a un utente di visualizzare e modificare i suoi dati se necessario.

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Come ultima funzione abbiamo il logout che sarebbe il pulsante esci in alto a destra. In questo caso abbiamo usato LocalStorage, una property di javascript che ci permette di salvare i dati dell’utente appena fa l’accesso, quando clicca su logout viene tolto dal LocalStorage e viene indirizzato alla home principale dove può fare il login di nuovo.

* 1. Admin

3.2.1 AdminDashboard

AdminDashboard è la pagina principale dell’amministratore che viene mostrata quando un utente con la variabile isAdmin a true fa l’accesso. In questa pagina l’admin ha l’opportunità di gestire sia utenti sia prodotti.

Immagine che contiene testo, schermata, software, computer

Descrizione generata automaticamente

3.2.1.1 UTENTI

3.2.1.1.1 GetAllUsers

Quando clicchiamo su Utenti ci viene mostrata una pagina di tutti gli utenti compresi anche gli amministratori e in questa pagina si possono compiere diverse operazioni come aggiungere un utente o un admin, vedere le informazioni modificare ed eliminare un utente, si possono anche vedere tutti gli ordini di un utente se quest’ultimo non li ha cancellati. In questa pagina abbiamo anche una barra di ricerca per cercare un utente qualsiasi. Gli utenti vengono mostrati in una tabella dove abbiamo le informazioni essenziali come l’Id, nome, E-mail e password.

Immagine che contiene testo, schermata, software, Icona del computer

Descrizione generata automaticamente

3.2.1.1.2 AddUser

AddUser ci permette di aggiungere un utente. Anche in questo caso abbiamo una gestione di errore ben implementata e vari controlli in tutti i campi. Per l’aggiunta di un utente abbiamo la variabile isAdmin a false.

Immagine che contiene testo, schermata, software, Software multimediale

Descrizione generata automaticamente

3.2.1.1.3 AddAdmin

Per l’aggiunta di un nuovo amministratore abbiamo ovviamente la variabile isAdmin a true.

Immagine che contiene testo, schermata, software, Sistema operativo

Descrizione generata automaticamente

3.2.1.1.4 GetUser

GetUser è la funzione che ci permette di vedere le informazioni di un utente. Ad esempio, in questo caso abbiamo l’Id, Nome, E-mail e Password dell’utente.

Immagine che contiene testo, schermata, software, Software multimediale

Descrizione generata automaticamente

3.2.1.1.5 UpdateUser

Questa è la funzione che ci permette di aggiornare i dati di un utente. Anche in questo caso c’è una gestione accurata degli errori e i controlli necessari.

Immagine che contiene testo, schermata, software, Sistema operativo

Descrizione generata automaticamente

3.2.1.1.6 AdminOrders

AdminOrders permette di vedere gli ordini effettuati da un utente. In questo caso abbiamo possiamo vedere anche cosa contiene l’ordine con alcune informazioni riguardante i prodotti.

Immagine che contiene testo, schermata, software, Icona del computer

Descrizione generata automaticamente

3.2.1.1.7 DeleteUser

Questa è una funzione per eliminare un utente e quando viene cliccato su elimina esce un messaggio che chiede se effettivamente si vuole eliminare l’utente.

Immagine che contiene testo, schermata, software, schermo

Descrizione generata automaticamente

3.2.1.2 PRODOTTI

3.2.1.2.1 GetAllProducts

Come per gli utenti, quando clicchiamo su prodotti ci viene mostrata una pagina di tutti i prodotti presenti e possiamo svolgere operazioni come l’aggiunta di un prodotto, visualizzare modificare oppure eliminare un prodotto. Anche in questo caso abbiamo la barra per la ricerca per nome dei prodotti.

Immagine che contiene testo, schermata, software, Icona del computer

Descrizione generata automaticamente3.2.1.2.2 AddProduct

La funzione che ci permette di aggiungere un prodotto. Abbiamo una gestione di errori appropriata, il controllo sulla compilazione di tutti i campi, il controllo sul nome del prodotto, non deve esistere nel database e il controllo anche sull’url dell’immagine

Immagine che contiene testo, schermata, software, Icona del computer

Descrizione generata automaticamente

3.2.1.2.3 GetProduct

Possiamo visualizzare le informazioni di un singolo prodotto.

Immagine che contiene testo, schermata

Descrizione generata automaticamente

3.2.1.2.4 UpdateProduct

Questa funzione ci permetti di aggiornare un prodotto con la gestione di eventuali errori e i controlli come all’aggiunta di un nuovo prodotto.

Immagine che contiene testo, schermata, software, Icona del computer

Descrizione generata automaticamente

3.2.1.2.5 DeleteProduct

Questa è una funzione per eliminare un prodotto e quando viene cliccato su elimina esce un messaggio che chiede se effettivamente si vuole eliminare il prodotto.

Immagine che contiene testo, schermata, software, Icona del computer

Descrizione generata automaticamente

1. CONFIGURAZIONE

4.1 Database

Per la persistenza dei dati, l’applicazione fa uso di MongoDB, un database non relazionale orientato ai documenti. MongoDB è stato scelto per la sua scalabilità, flessibilità e facilità d’uso, adatto alle esigenze dinamiche dell’applicazione.

**Struttura del Database:**

* Il database è organizzato in quattro collezioni principali corrispondenti ai moduli dell’applicazione:

1. Collezione “users” per la gestione degli utenti registrati.

Immagine che contiene testo, schermata, software, schermo

Descrizione generata automaticamente

1. Collezione “products” per la gestione dei prodotti disponibili.

Immagine che contiene testo, elettronica, schermata, software

Descrizione generata automaticamente

1. Collezione “carts” per la memorizzazione dei carrelli degli utenti.

Immagine che contiene testo, schermata, software

Descrizione generata automaticamente

1. Collezione “orders” per la memorizzazione degli ordini effettuati dagli utenti.

Immagine che contiene testo, schermata

Descrizione generata automaticamente

**Tecnologie Utilizzate:**

* MongoDB Atlas è stato utilizzato come servizio di database cloud per la gestione dei dati.
* Mongoose, una libreria di modellazione dei dati MongoDB in Node,js, è stata impiegata per definire i modelli dei documenti e facilitare le operazioni di accesso al database.

4.2 Server

Il server è stato configurato utilizzando Node.js insieme al framework Express.js per fornire un’interfaccia RESTful per il client e gestire le richieste http.

**Moduli del server:**

Nel server abbiamo dei moduli per la gestione di diverse operazioni. Ad esempio abbiamo:

1. Modulo “user”: Gestisce le operazioni relative agli utenti, tra cui la registrazione, l’autenticazione e la gestione del profilo ect…
2. Modulo “product”: Gestisce le operazioni relative ai prodotti, compresa la visualizzazione dei prodotti disponibili e la ricerca ect…
3. Modulo “cart”: Gestisce le operazioni legate ai carrelli degli utenti, inclusa l’aggiunta, la rimozione e il salvataggio dei prodotti nel carrello e anche l’operazione per il pagamento. In questo modulo abbiamo anche raggiunto le operazioni degli ordini come visualizzare ed eliminare gli ordini degli utenti.

**Tecnologie Utilizzate:**

* **Express.js è stato utilizzato per la gestione delle richieste HTTP e la definizione delle rotte dell’applicazione.**
* **Cors (Cross-Origin Resource Sharing) è stato configurato per consentire le richieste da origini multiple al server.**

4.3 Client

Il client dell’applicazione è stato sviluppato utilizzando React.js, una libreria JavaScript per la creazione di interfacce utente dinamiche. Axios viene impiegato per effettuare richieste HTTP al server e gestire la comunicazione tra il client e il back-end, garantendo un flusso efficiente di dati. Inoltre, React Router è utilizzato per gestire la navigazione e definire rotte all’interno dell’applicazione client, assicurando una navigazione fluida tra le diverse sezioni dell’app.

L’applicazione è progettata per offrire agli utenti un’esperienza completa di shopping online. Include funzionalità di registrazione e login per gli utenti, nonché la gestione dei profili utente. La piattaforma consente agli utenti di visualizzare i prodotti disponibili, ottenere dettagli su di essi e aggiungerli al carrello per l’acquisto. Inoltre, offre un’interfaccia per la gestione e la visualizzazione degli ordini effettuati dagli utenti.

In conclusione, l’applicazione si concentra su un’esperienza utente fluida e intuitiva per esplorare, selezionare e acquistare prodotti online.