Relazione Tecnica

DELFRATE RICCARDO

5B Informatica e telecomunicazione articolazione informatica

A.s 2020/2021

La seguente relazione illustrerà il funzionamento, le procedure e gli strumenti utilizzati nel mio elaborato.
https://drive.google.com/drive/folders/1JoT5XMdPn-guzbhA4v29DzD5XdWTGbjLBnCP-SUK9Gj6tHYGb640lFwD1D6PmEhWlzu7vExn

INDICE

Richieste:

Un'azienda di e-Commerce richiede la realizzazione di un sito dinamico che permetta la navigazione del sito senza alcun tipo di registrazione.
Si preveda la registrazione, attraverso l'inserimento di mail e di password, per i clienti che vorranno effettuare degli acquisti.

Il sito dovrà prevedere la gestione del carrello.
Dopo aver effettuato il login il cliente dovrà aver inserito nel carrello almeno un elemento in modo tale che, per gli accessi successivi, ritroverà le scelte precedentemente effettuate pronte per un eventuale acquisto, che avverrà con una carta di credito con una modalità protetta.
Il candidato, con riferimento a quanto descritto, facendo le opportune ipotesi aggiuntive, produca un elaborato che:

Sistemi

- La metodologia e i protocolli utilizzati per lo scambio delle informazioni sulla carta di credito in modo criptato tramite algoritmo RSA.
- Approfondisca anche in ambito matematico l'algoritmo RSA.

Informatica

- Realizzi il progetto della base di dati per la gestione di quanto descritto, in particolare si richiedono: il modello concettuale, il corrispondente modello logico e fisico.
- Disegni e realizzi un sito web che consenta la realizzazione delle funzionalità sopra descritte.

Progettazione

Requisiti funzionali:

- Un utente per poter procedere all'acquisto dovrà essere registrato.
- Saranno presenti 2 barre poste una lateralmente e l'altra sovrastante la visualizzazione dei prodotti che permetteranno di raggiunge le funzionalità del sito.
- Sarà presente un form di registrazione per permettere di registrarsi.
- Ogni utente una volta registrato avrà la possibilità di accedere alla sua area riservata.
- L'area riservata conterrà un carrello con il riepilogo di tutti i prodotti selezionati.
- La parte al di fuori dell'area riservata sarà visibile a tutti.
- Sarà presente un form per l'inserimento dei dati sulla carta di credito.
- Sarà presente un form per l'inserimento dei dati di spedizione.
- Sarà presente un bottone di logout per effettuare la disconnessione.

Requisiti non funzionali:

- In fase di registrazione sarà presente un controllo per verificare che le due password inserite corrispondano.
- Prima di avanzare alla fase di pagamento sarà obbligatorio accettare i termini e le condizioni.
- Una volta effettuato l'accesso verranno visualizzati bottoni per utilizzare funzionalità extra.
- In caso di eventuali errori/operazioni avvenute verranno comunicati/e all'utente tramite alert.
- Tutti i dati dell'utente che ha effettuato la registrazione saranno salvati nel database.
- La password dovrà necessariamente essere di almeno 8 caratteri e contenere almeno un numero, almeno una lettera maiuscola e almeno una lettera minuscola.
- La password prima di venire inserita verrà criptata utilizzando due metodologie hash differenti.
- Ogni articolo apparterrà ad una categoria.

Work Breakdown Structure:

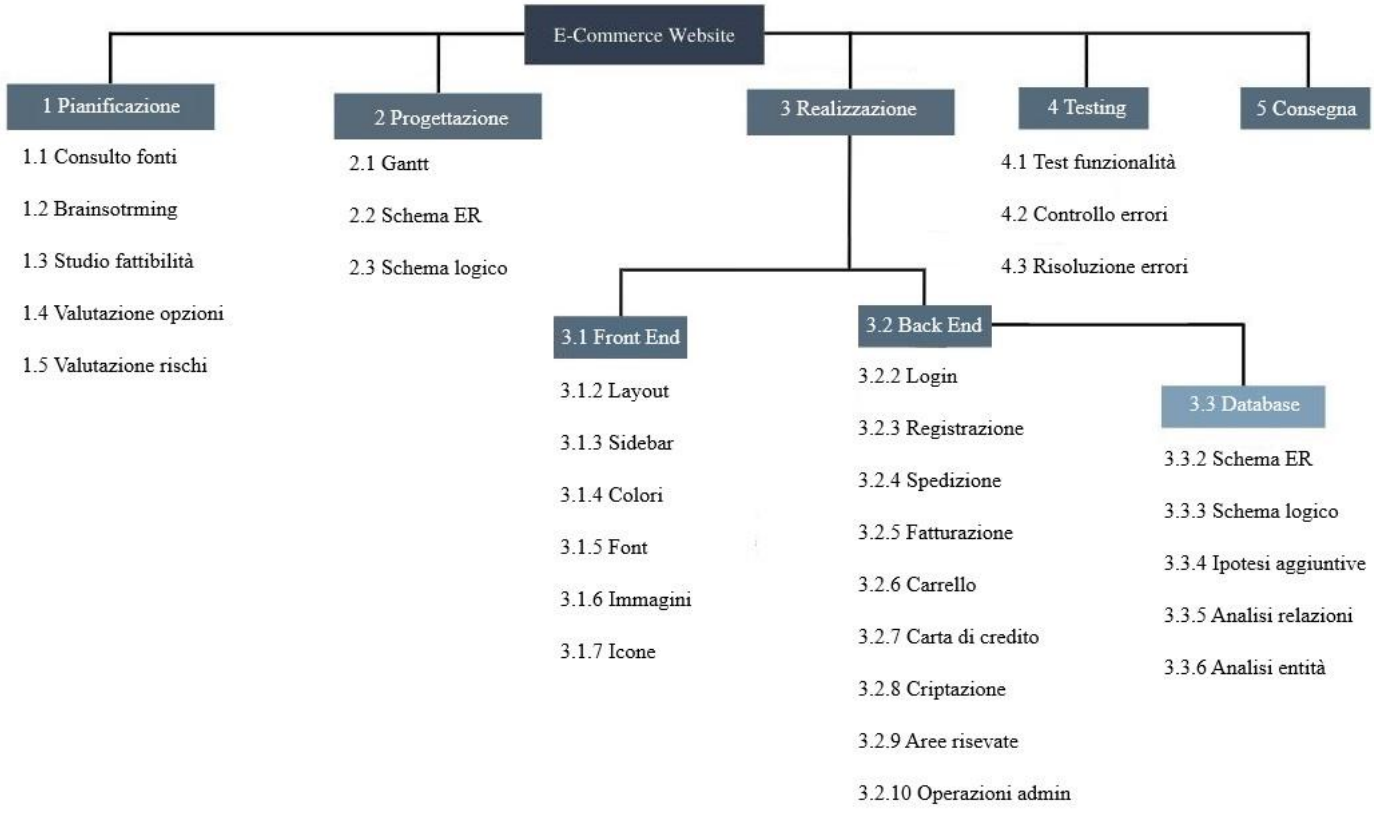
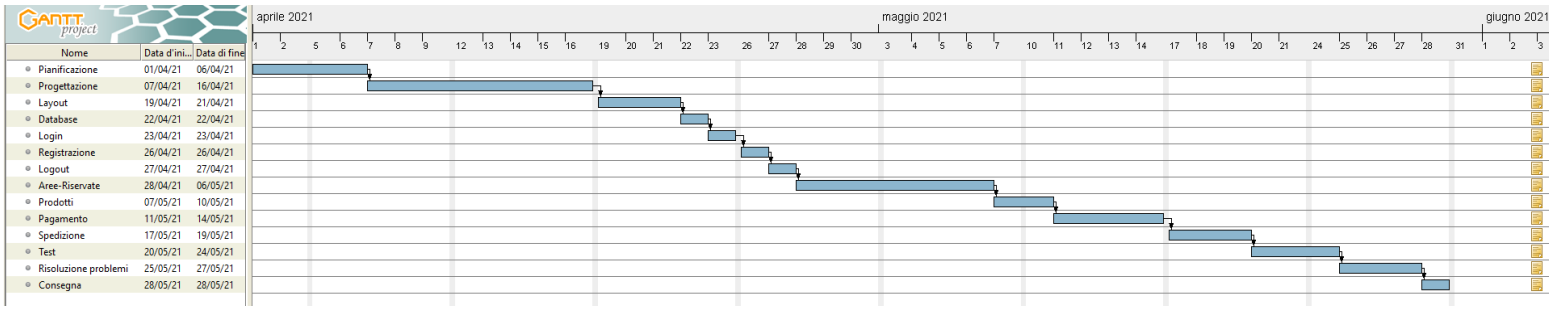


Diagramma di Gantt:

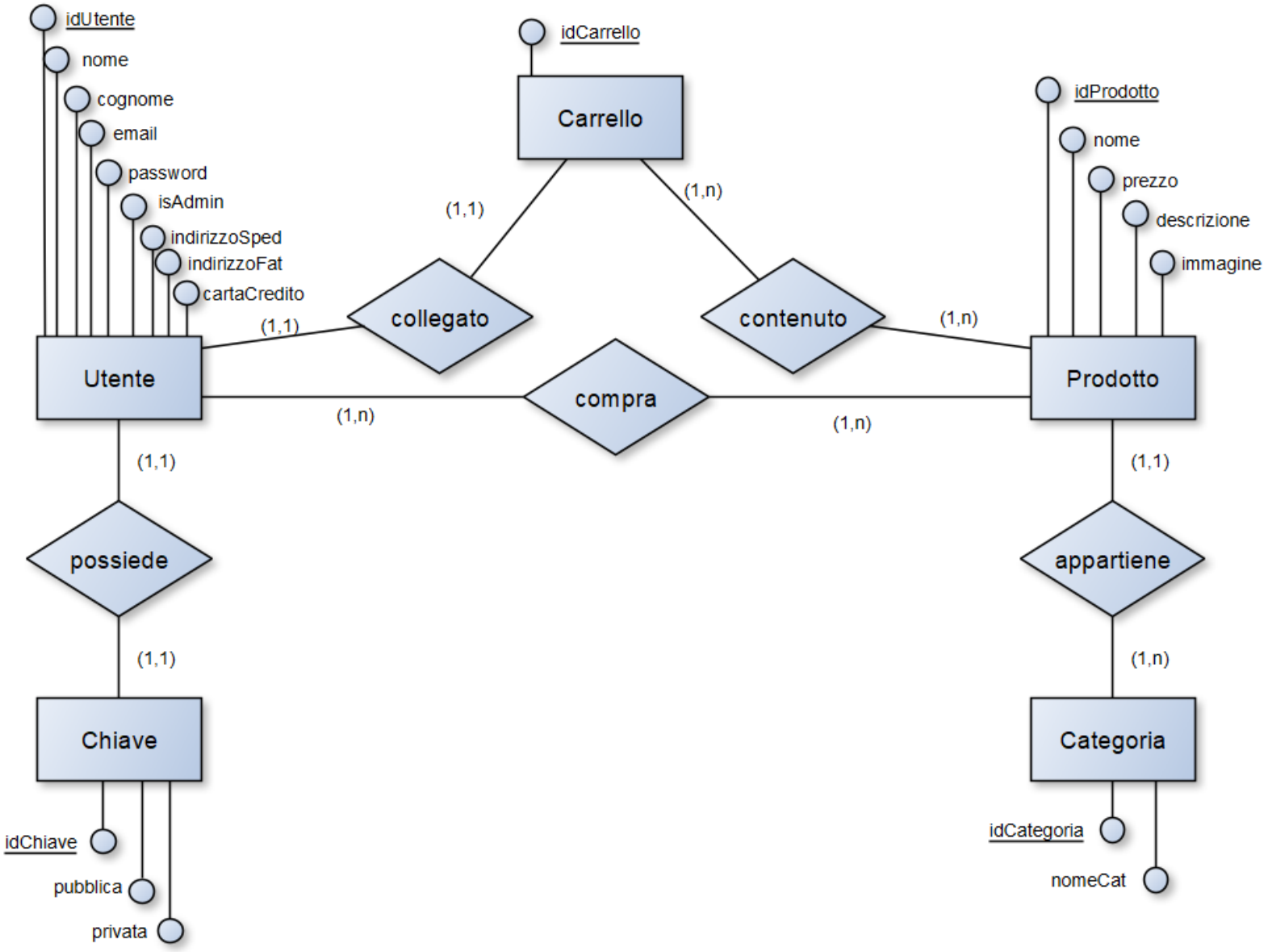


- **Pianificazione:**
  Pianificazioni di tutte le varie attività successive, brainstorming, valutazioni delle opzioni, valutazione dei rischi, studio fattibilità del progetto, consulto fonti.
- **Progettazione:**
  Realizzazione analisi del problema, realizzazione schema ER, realizzazione schema logico.
- **Layout:**
  Realizzazione del layout del sito, scelta dei colori, del font e del template.
- **Database:**
  Realizzazione schema ER schema logico e creazione fisica del database in MySQL.
  Identificazione delle entità e delle relazioni.
- **Login:**
  Creazione del form di login per permettere all'utente di accedere all'area riservata.
- **Registrazione:**
  Creazione del form di registrazione che permetterà di essere aggiunti nel database con relative operazioni per garantirne la sicurezza.
- **Logout:**
  Implementazione pagina di logout la quale permetterà di disconnettersi chiudendo la sessione ed eliminando tutti i dati contenuti in essa.
- **Aree-Riservate:**
  Implementazione aree riservata per amministratore e utente standard.
  L'amministratore avrà una panoramica di tutte le operazioni che potranno essere eseguite nel database e visualizzare un riepilogo dei prodotti, le categorie e gli utenti presenti.
  L'utente standard una volta avvenuto il login verrà riportato nella home page dove compariranno il bottone per la visualizzazione del carrello il quale potrà essere usato per il completamento dell'ordine e il bottone di logout.
- **Prodotti:**
  Realizzazione della parte di visualizzazione dei prodotti presenti nel database.
- **Pagamento:**
  Implementazione form di inserimento dei dati relativi alla carta di credito e implementazione dell'algoritmo RSA per la sicurezza di essi.
- **Spedizione:**
  Realizzazione form per l'inserimento dei dati relativi all'indirizzo di spedizione e di fatturazione.
- **Test:**
  Verifica di tutte le funzionalità offerte e individuazione di eventuali errori.
- **Risoluzione problemi:**
  Risoluzione dei problemi emersi dalla fase di test.
- **Consegna:**
  Consegna finale progetto.

Ipotesi aggiuntive:

- Amministratore può effettuare operazioni di inserimento.
- Per fare acquisti bisogna registrarsi in modo da essere inseriti in un database.
- La password verrà criptata in modo adeguato prima di essere inserita nel database
- Sarà presente un campo nella tabella utente per distinguere l'amministratore dall'utente standard.
- Sarà presente un carrello contenente tutti i prodotti con costo finale e bottone per procedere all'acquisto.
- Il form fi login sarà unico, con reindirizzamelo in pagine diverse a seconda dei permessi

Schema ER:



Schema logico:

Chiave: (idChiave(PK), pubblica, privata);
Utente: (idUtente(PK), nome, cognome, email, password, isAdmin, via, cap, nCiv, comune, provincia, FragioneSociale, FpecCod, FpartitaIva, Fvia, Fcap, FnCiv, Fcomune, Fprovincia, FcodFiscale, cartaCredito idChiave(FK), idCarrello(FK));
Compra(idCompra(PK), idUtente(FK), idProdotto(FK));
Prodotto: (idProdotto(PK), nome, prezzo, descrizione, immagine, idCategoria(FK));
Carrello: (idCarrello(PK));
Contenuto(idContenuto(PK), idCarrello(FK), idProdotto(FK));
Categoria: (idCategoria(PK), nomeCat);

Analisi associazioni:

Chiave-Utente:

Tra l'entità chiave e l'entità utente è presente un'associazione "possiede" uno a uno in quanto un utente possiede a una sola coppia di chiavi e una coppia di chiavi è posseduta da un solo utente.

Utente-Carrello:

Tra l'entità Utente e l'entità Carrello è presente un'associazione "collegato" uno a uno in quanto un utente è collegato ad un solo carrello e un carrello è collegato ad un solo utente.

Utente-Prodotto:

Tra l'entità Utente e l'entità Prodotto è presente un'associazione "compra" molti a molti in quanto un utente può comprare più prodotti e un prodotto può essere acquistato da più utenti.

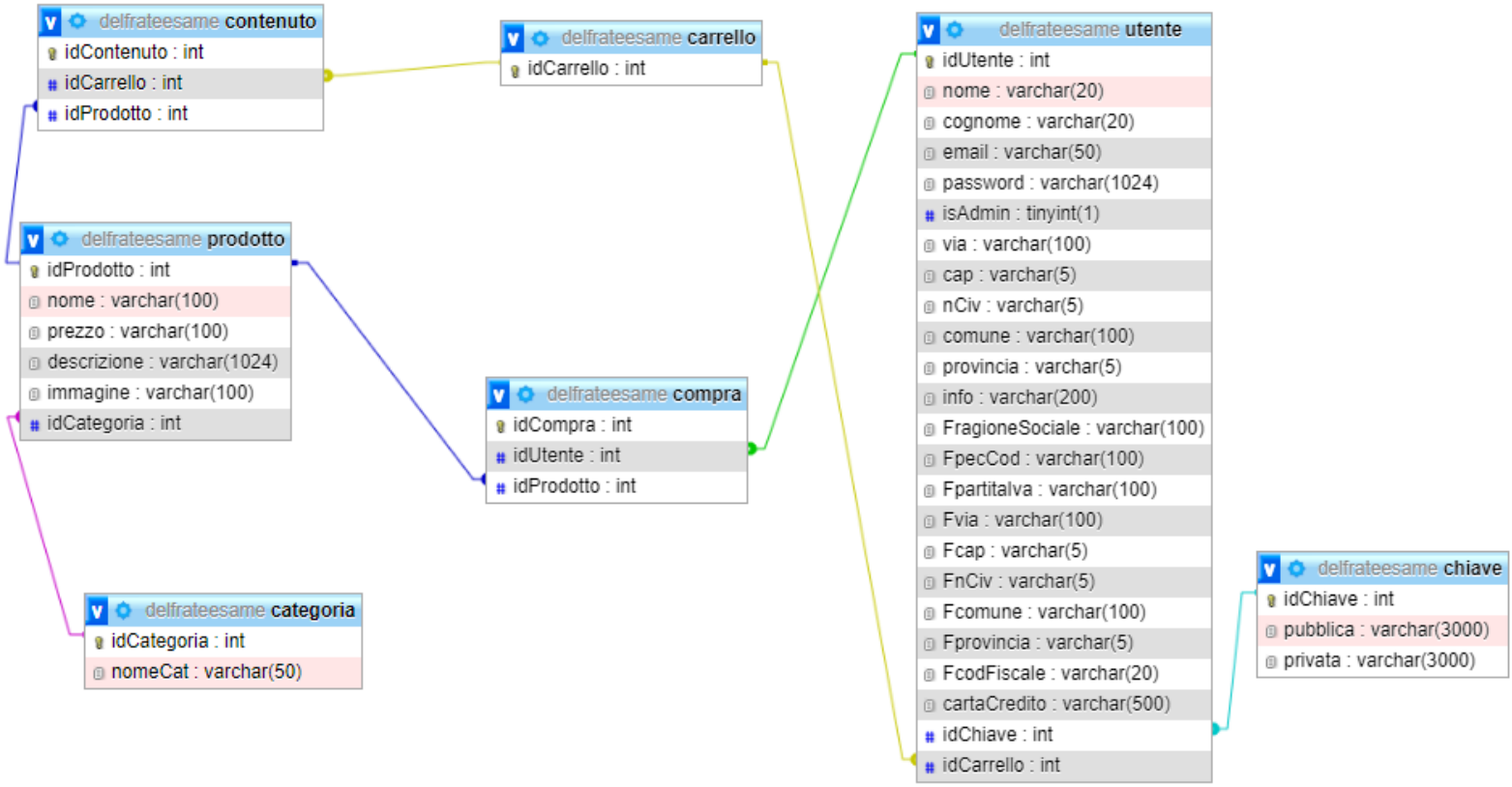
Carrello-Prodotto:

Tra l'entità Carrello e l'entità Prodotto è presente un'associazione "contenuto" molti a molti in quanto un carrello può contenere più prodotti e un prodotto può essere contenuto in più carrelli.

Prodotto-Categoria:

Tra l'entità Prodotto e l'entità Categoria è presente un'associazione "appartiene" uno a molti in quanto un prodotto appartiene a una sola categoria e a una categoria appartengono più prodotti.

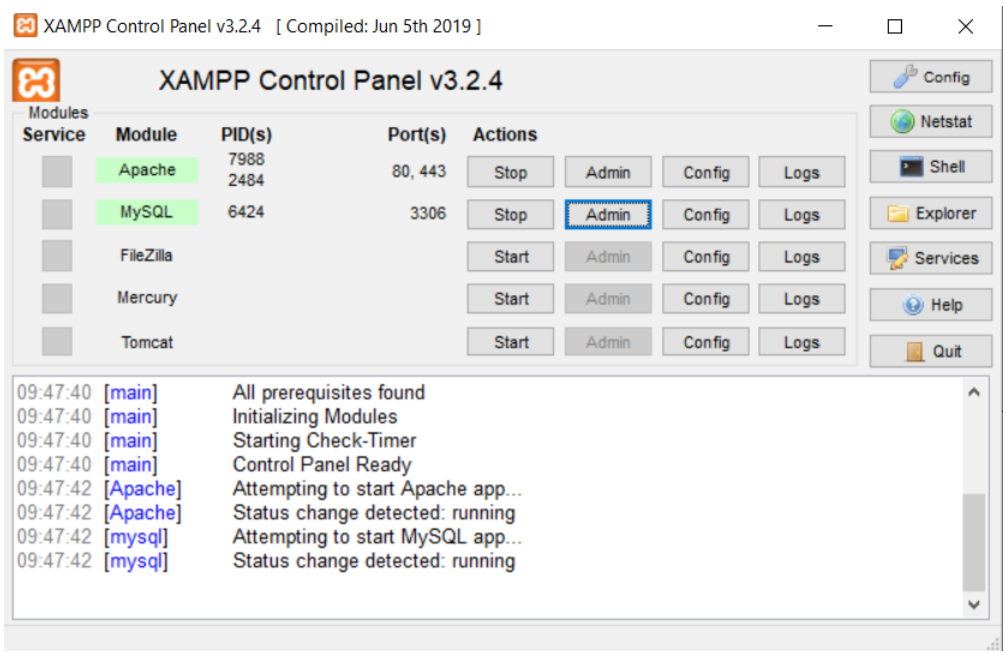
Realizzazione fisica



Strumenti utilizzati:

XAMPP:

Ho deciso di utilizzare XAMPP in quanto offre la possibilità di gestire un Application server capace di interpretare pagine web dinamiche e un modulo MySQL il quale permette di avere a disposizione un database.



Linguaggi di programmazione utilizzati

PHP:

PHP è un linguaggio di scripting lato server interpretato, concepito per la programmazione di pagine web dinamiche. L'interprete PHP è un software libero distribuito sotto la PHP License.

Ho deciso di utilizzare questo linguaggio in quanto è in grado di interfacciarsi a innumerevoli DBMS tra cui MySQL per il quale è ottimamente integrato, inoltre fornisce un'API specifica per interagire con il web-server Apache.

Javascript:

JavaScript è un linguaggio di programmazione orientato agli oggetti e agli eventi lato client, ho utilizzato questo linguaggio in quanto offre una grande interoperabilità con html e PHP.

Script

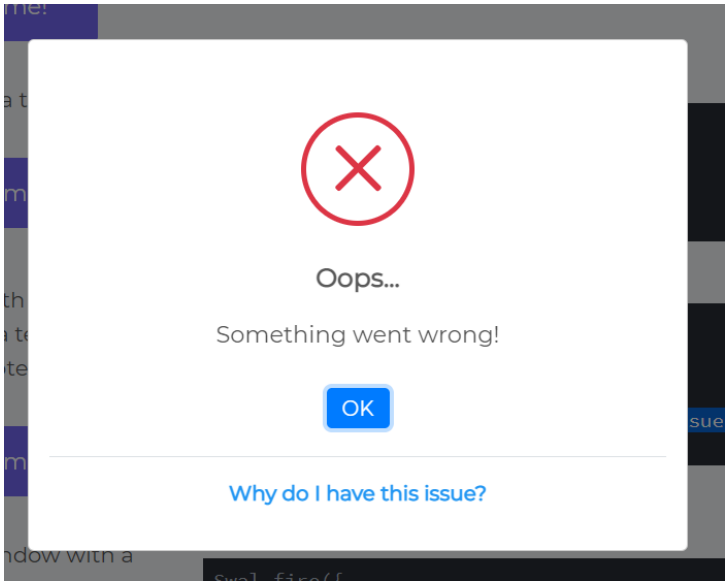
Sweet alert:

Sweet alert è uno script Javasciprt che ho utilizzato per migliorare la visualizzazione degli alert.
Per utilizzare questo script è necessaria la sua incorporazione nel codice della pagina tramite la seguente linea:

```
<script src="https://cdn.jsdelivr.net/npm/sweetalert2@10.16.6/dist/sweetalert2.all.min.js"></script>
```

Alert standard:

Sweet Alert:



Codice:

```
Swal.fire({
  icon: 'error',
  title: 'Oops...',
  text: 'Something went wrong!',
  footer: '<a href>Why do I have this issue?</a>' })
```

Linguaggio di markup utilizzato

Html:

Html “HyperText Markup Language” è un linguaggio di formattazione che descrive le modalità di impaginazione o visualizzazione grafica (layout) del contenuto, testuale e non, di una pagina web attraverso tag di formattazione.
Ogni tag (ad esempio <h1> o <p>) specifica un diverso ruolo dei contenuti che esso contrassegna.
I browser che leggono il codice mostrano all'utente formattazioni predefinite per ogni tag che incontrano (così ad esempio i contenuti marcati con il tag <h1> avranno carattere 18pt e i contenuti marcati da <p> avranno carattere 12pt).

Strumenti grafici

Bootstrap:

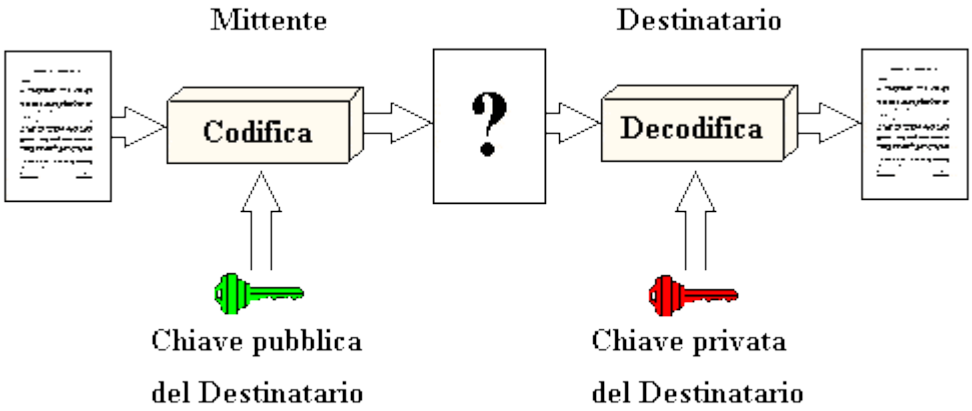
Bootstrap è una raccolta di strumenti liberi per la creazione di siti e applicazioni per il Web, contiene modelli di progettazione basati su HTML e CSS.
L’ho scelto rispetto ad un CMS in quanto offre una maggior libertà di scelta dell’implementazione, inoltre supporta anche il “Responsive web design”, ciò significa che il layout delle pagine web si regola dinamicamente, tenendo conto delle caratteristiche del dispositivo utilizzato, sia esso desktop, tablet o telefono cellulare.
Per utilizzare Bootstrap è necessario includere nel codice i collegamenti ad esse.
Esempio:
<link href="https://stackpath.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css" rel="stylesheet">

Css:

Il CSS “Cascading Style Sheets” è un linguaggio usato per definire la formattazione di diversi documenti, nel mio caso l’ho utilizzato per la formattazione dei documenti Html e PHP.
Necessita di essere incorporato nel file in cui si richiede il suo utilizzo.
Esempio:
<link href="./style.css " rel="stylesheet" media="all">

RSA:

RSA è un algoritmo di criptazione a chiave asimmetrica inventato nel 1977 da Ronald Rivest, Adi Shamir e Leonard Adleman basato sulla fattorizzazione di numeri primi (Problema computazionalmente complesso).
Data la sua natura “Asimmetrica” si basa sull'esistenza di due chiavi distinte, che vengono usate per cifrare e decifrare. Se la prima chiave viene usata per la cifratura, la seconda deve necessariamente essere utilizzata per la decifratura e viceversa.



La questione fondamentale è che, nonostante le due chiavi siano fra loro dipendenti, non è possibile risalire dall'una all'altra, in modo che se anche si è a conoscenza di una delle due chiavi, non si possa risalire all'altra, garantendo in questo modo l'integrità della crittografia.

Esempio elenco telefonico:
Trovare il numero di telefono dal nome di un utente è immediato mentre trovare il nome dato il numero di telefono richiederebbe un tempo x (non infinito ma da considerarsi tale con grandi numeri).
L'RSA usa la moltiplicazione di numeri primi grandi come funzione "Facile da calcolare" n=p*q e la sua funzione opposta ovvero la fattorizzazione come funzione "Infinita".

Funzionamento

- Si scelgono a caso due numeri primi, *p* e *q* abbastanza grandi da garantire la sicurezza dell'algoritmo (hanno più di 300 cifre ciascuno).
- Si calcola *n = p*q*
- Si calcola *φ(n) = (p-1)(q-1)*
- Si considera che la fattorizzazione di n è segreta e solo chi sceglie i due numeri primi, p e q, la conosce
- Si sceglie poi un numero *e* (chiamato esponente pubblico), coprimo con *φ(n)* e più piccolo di *φ(n)*
- Si calcola il numero *d* (chiamato esponente privato) tale che il suo prodotto con *e* sia congruo a *1 Mod(φ(n))* ovvero che *ed=1 Mod(φ(n))*

La chiave pubblica è *(n,e)*, mentre la chiave privata è *(n,d)*

Numeri e caratteri
Dato che l'RSA lavora con i numeri ci si appoggia sulla tabella ASCII per ottenere il valore numerico corrispondente da ciascun carattere.

Esempio algoritmo

Nella seguente tabella è rappresentata la cifratura nel carattere 'A' corrispondente al numero 65 nella tabella ASCII.

Inserimento dati	
p (1° numero primo)	43
q (2° numero primo)	47
m messaggio chiaro ($m < N$)	65
Cifra	

Procedura			
Descrizione	**Variabile**	**Formula**	**Valore**
Alice sceglie due numeri primi $p = 43$ e $q = 47$ e li moltiplica pubblicando il risultato	N *chiave pubblica*	$p \times q = 43 \times 47$	2021
Alice, conoscendo p e q calcola facilmente la funzione di Eulero $\Phi(N)$ che deve restare segreta.	$\Phi(N)$ *chiave privata*	$\Phi(N) = (p-1)(q-1) = \Phi(2021) = 42 \times 46$	1932
Alice calcola la seconda chiave e, primo numero primo con $\Phi(N)$ e la pubblica.	e *chiave pubblica*	primo numero tale che $MCD(e, \Phi(N)) = 1$	5
Alice calcola la chiave segreta d, che è l´inverso di e modulo $\Phi(N)$ e NON la pubblica.	d *chiave privata*	d tale che $e \times d \equiv 1 \pmod{\Phi(N)}$	773
Bruno per inviarle un messaggio m usando le chiavi pubbliche di Alice, N ed e, calcola la potenza $c = m^e \pmod{N}$ ed invia c per un canale pubblico.	c *cifrato*	$c \equiv m^e \pmod{N} = 65^5 \pmod{2021} =$	168
Alice e solo Alice che conosce la chiave segreta d può ora decifrare il messaggio m semplicemente calcolando la potenza inversa.	m *decifrato*	$m \equiv c^d \pmod{N} = 168^{773} \pmod{2021} =$	65

Perché è sicuro e perché si usano i numeri primi?

Premessa

L'aritmetica che serve per implementare l'algoritmo RSA è quella della struttura algebrica Z_n:
L'insieme delle classi di resto modulo n, cioè l'insieme {0, 1, ..., n−1} in cui sono definite la somma e il prodotto nel seguente modo:

- a+b = mod(a+b, n)
- a·b = mod(a·b, n)

dove mod(x,y) è la funzione che restituisce il resto del quoziente tra numeri interi x/y.

13 : 2 = 6 1 13 mod(2)=1

Quoziente Resto

Domanda: esiste una legge generale anche per l'inverso di a in Zn?
Per esempio, qual è l'inverso di 1234 in Z_{1789}?
È possibile calcolarlo direttamente oppure dobbiamo calcolare 1234·1, 1234·2, 1234·3, ... in Z_{1789} fino a che il risultato è 1?
Vediamo le principali operazioni effettuabili: somma prodotto ed elevamento a potenza.

Esempio tabella di somma in Z_9:

+	0	1	2	3	4	5	6	7	8
0	0	1	2	3	4	5	6	7	8
1	1	2	3	4	5	6	7	8	0
2	2	3	4	5	6	7	8	0	1
3	3	4	5	6	7	8	0	1	2
4	4	5	6	7	8	0	1	2	3
5	5	6	7	8	0	1	2	3	4
6	6	7	8	0	1	2	3	4	5
7	7	8	0	1	2	3	4	5	6
8	8	0	1	2	3	4	5	6	7

Nella tabella della somma si osserva che la distribuzione degli opposti (celle scure) segue la seguente legge:
l'opposto di a in Z_n è n-a

Esempio:
Considero a=5 e z=9 → l'opposto di a in Z_9 è → 9-5 → 4 (notare cella scura in coordinate 5-4)

Questa tabella risulta estremamente regolare quindi è da scartare.

Esempio tabella di prodotto in Z_9:

x	1	2	3	4	5	6	7	8
1	1	2	3	4	5	6	7	8
2	2	4	6	8	1	3	5	7
3	3	6	0	3	6	0	3	6
4	4	8	3	7	2	6	1	5
5	5	1	6	2	7	3	8	4
6	6	3	0	6	3	0	6	3
7	7	5	3	1	8	6	4	2
8	8	7	6	5	4	3	2	1

Nella tabella del prodotto si osserva innanzitutto che non tutti gli elementi hanno inverso (solo 2, 4, 5, 7 e 8 sono invertibili); come è noto, in Z_n ammettono inverso solo i numeri a primi con n, cioè tali che MCD(a, n) = 1;

Esempio:
a=10 in Z_9 MCD(10,9)=1 in tabella casella scura con coordinate 5-2

Da questa tabella si vede che non vi è alcune regolarità evidente (Tranne una simmetria ma trascurabile in questo ragionamento) per questo non va bene.

Esempio tabella elevamento a potenza Z_{11}:

^	1	2	3	4	5	6	7	8	9	10
2	2	4	8	5	10	9	7	3	6	1
3	3	9	5	4	1	3	9	5	4	1
4	4	5	9	3	1	4	5	9	3	1
5	5	3	4	9	1	5	3	4	9	1
6	6	3	7	9	10	5	8	4	2	1
7	7	5	2	3	10	4	6	9	8	1
8	8	9	6	4	10	3	2	5	7	1
9	9	4	3	5	1	9	4	3	5	1
10	10	1	10	1	10	1	10	1	10	1

Si osservi l'ultima colonna: è composta di soli 1, cioè per ogni a non nullo risulta:
in Z_{11}: $a^{10}=1$.
Per ogni $a \in Z_n$, $a \neq 0$, risulta $a^{n-1} = 1$ → in Z_{11} 6^{11-1} → 6^{10} → 1
(60466179Mod(11)=1)

Se così fosse, avremmo trovato una legge generale per l'inverso di a in Z_n:
L'inverso di a sarebbe a^{n-2}.

In realtà la congettura è corretta se e solo se n è primo.
Questo è infatti quanto afferma il seguente.

Teorema di Fermat

Se e solo se n è primo, per ogni $a \in Z_n$ non nullo risulta $a^{n-1} = 1$.

Se n non è primo?

Torniamo a prendere in considerazione Z_9 questa volta con l'elevamento a potenza**.** (9 non è un numero primo)
Il numero 1 non viene considerato perché sempre presente.

n = 9	^	1	2	3	4	5	6	7	8
	2	2	4	8	7	5	1	2	4
	3	3	0	0	0	0	0	0	0
	4	4	7	1	4	7	1	4	7
	5	5	7	8	4	2	1	5	7
	6	6	0	0	0	0	0	0	0
	7	7	4	1	7	4	1	7	4
	8	8	1	8	1	8	1	8	1

Da questa tabella osserviamo una colonna particolare, sembra che esista un esponente (6) tale che ogni elemento invertibile (2-4-5-7-8) elevato ad esso sia uguale ad 1.

Teorema di Eulero

Per ogni $n \geq 2$ e per ogni $a \in Z_n$ invertibile risulta $a^{\varphi(n)}=1$, dove $\varphi(n)$ è il numero di naturali compresi tra 1 e n che sono primi con n.

Funzione di Eulero φ(n)

Associa ad ogni numero naturale n il numero di numeri a∈{1, 2, ..., n} tali che MCD(a, n) = 1.
Tabella contenente le prime φ(n) con 2≤n≤15.

n	2	3	4	5	6	7	8	9	10	11	12	13	14	15
φ(*n*)	1	2	2	4	2	6	4	6	4	10	4	12	6	8

In questo caso osserviamo che da entrambe le tabelle emerge che φ(9)=6.
Questa funzione possiede una proprietà particolare ovvero:
Presi 2 numeri p e q primi distinti, allora φ (pq) = (p−1) · (q−1).
Esempio:
φ (3*5) = 8 → (3-1)*(5-1) = 8

Le proprietà della funzione φ mostrano che il calcolo di φ (n) è immediato se si conosce la fattorizzazione di n;
ma la fattorizzazione di n è esattamente ciò che verrà tenuto nascosto e che costituisce il motivo della inattaccabilità di RSA; le difficoltà di calcolare φ (n) e di fattorizzare n sono equivalenti.
Infatti:

- Dalla fattorizzazione di n (n = pq) si ricava immediatamente φ(n): φ(n) = (p−1)(q−1);
- Dalla conoscenza di φ (n) e n si ricava la fattorizzazione di n: $x^2 - (p+q)x + pq = 0$.

Fortunatamente non è necessario conoscere φ(n) per calcolare l'inverso di a in Z_n possiamo usare un'applicazione dell'algoritmo di Euclide.

Algoritmo di Euclide

L'algoritmo di Euclide è un algoritmo molto efficiente per il calcolo del MCD tra due numeri naturali a e b in tempi molto ragionevoli.

Phpseclib

Secure Communications Library offre una serie di file in formato PHP per l'implementazione di diverse soluzioni per gestire la criptazione e non solo.
Nel mio caso l'ho scelta per l'implementazione della criptazione RSA.

Criptazione

Per prima cosa includo il file RSA.php fornitomi dalla libreria, mi salvo i valori passati tramite metodo post in delle variabili.
Successivamente inizializzo un oggetto **$rsa** legato alla classe **Crypt_RSA** ed utilizzo la funzione **extract()** e il metodo **createKey()**.

extract():
Questa funzione utilizza chiavi dell'array come nomi di variabile ei valori come valori variabili.
Per ogni elemento si creerà una variabile nella tabella dei simboli.

createKey():
Ritorna un array con i seguenti elementi:
'privatekey': Chiave privata.
'publickey': Chiave pubblica.
Passo come parametro 2048 in quanto identifica la lunghezza in bit della chiave (2048 bit) in quanto sono lo standard di criptazione odierno.

```php
<?php
include 'phpseclib/Crypt/RSA.php';

$numero = $_POST['numero'];
$nome = $_POST['nome'];
$mese = $_POST['mese'];
$anno = $_POST['anno'];
$cvc = $_POST['cvc'];


$rsa = new Crypt_RSA();
extract($rsa->createKey(2048));
```

Per criptare il messaggio gli accodo le variabili separate da un carattere '-' in un'altra variabile chiamata **$plaintext** che conterrà il testo in chiaro.

Attraverso il metodo **loadKey()** a cui passo la chiave pubblica "carico" la chiave per la criptazione.
Cripto il testo in chiaro con il metodo **encrypt()** passandoglielo come parametro ed ottengo il testo criptato **$ciphertext**.
Una volta ottenuto il testo criptato ho la necessita di codificarlo in base 64 (**base64_encode()**) in quanto il database non riesce a salvare il ciphertext nella sua versione originale binaria.
Successivamente eseguo la query per inserire i dati criptati e codificati (**$ciphertext64**) del database.

```php
//----- CRIPTAZIONE -----
$plaintext = "$numero-$nome-$mese-$anno-$cvc";

$rsa->loadKey($publickey);
$ciphertext = $rsa->encrypt($plaintext);
$ciphertext64 = base64_encode($ciphertext);

$update3 = "UPDATE utente SET cartaCredito = '$ciphertext64' WHERE idUtente = $idUtente";
if (!mysqli_query($conn, $update3)) {
    ?>
    <script>
        Swal.fire({
            icon: 'error',
            title: 'Ops...',
            text: 'Cè stato un problema in fase di registrazione',
            allowEscape: false,
            allowOutsideClick: false,
            confirmButtonText: "<a href='../login/login.php'>Riprova</a>"
        });
    </script>
<?php
}
//----- FINE CRIPTAZIONE -----
```

Decriptazione

```php
//----- DECRIPTAZIONE -----
$rsa->loadKey($privatekey);
$cript64Dec = base64_decode($ciphertext64);
$decriptato = $rsa->decrypt($cript64Dec);
//----- FINE DECRIPTAZIONE -----
```

Per decriptare i dati utilizzo sempre il metodo **loadKey()** passandogli questa volta la chiave privata (**$privatekey**).
Riporto da base 64 a binario (**base64_decode()**) e mi salvo il risultato nella variabile **$cript64Dec()**, infine decripto attraverso il metodo **decrypt()**.

Registrazione

In fase di registrazione ho implementato la criptazione della password nel seguente modo:

```php
//----CRIPTAZIONE PASSWORD-----
function cryptPassword($password)
{
    $hash1 = crypt($password, '$1$shfpersh$');
    $hash2 = crypt($hash1, '$6$rounds=5000$K8d6sRt2$');
    return $hash2;
}
//---- FINE CRIPTAZIONE PASSWORD-----
```

La funzione riceve come parametro la password in chiaro inserita dall'utente, la passa alla funzione crypt().
Crypt() necessita di due parametri la stringa da criptare e il metodo di criptazione con il salt ovvero dei caratteri aggiuntivi per migliorarne la sicurezza.

(La criptazione con SHA-512 ha bisogno di aver specificato l' "Hashing loop" indicato con rounds=500 ovvero il numero di volte che il ciclo Hashing va ripetuto)

Queste funzioni Hash hanno 2 caratteristiche:

- Sono irreversibili, infatti una volta criptata la stringa non può più tornare alla sua forma "Normale".
- Se si usa la medesima funzione per criptare 2 stringe identiche si ottiene lo stesso risultato.

Nel primo caso 1 identifica l'algoritmo MD5 con salt "shfpersh".
Il risultato viene passato alla seconda funzione crypt la quale usa l'algorimo SHA-512 6 con salt "K8d6sRt2".

Controllo sulla password

In fase di inserimento della password ho verificato che venissero rispettate determinate caratteristiche:
Lunghezza minima di 8 caratteri
Minimo una lettera maiuscola
Minimo una lettera minuscola
Minimo un numero.
<input type="password" name="password" pattern="(?=.*\d)(?=.*[a-z])(?=.*[A-Z]).{8,}" title="Almeno un numero, una lettara minuscola o Maiuscola e almeno 8 o più caratteri" placeholder="Password" required>

Gestione carrello

Come da richiesta è stato messo a disposizioni di ogni utente un carrello identificato da un id il quale conterrà tutti i prodotti che l'utente ha selezionato.
Quest'area del sito è accessibile solo se prima è stato effettuato il login.
Per la visualizzazione dei prodotti presenti ho usato la seguente query:
$select = "SELECT * FROM prodotto P INNER JOIN contenuto C ON P.idProdotto=C.idProdotto WHERE c.idCarrello=$idCarrello";
In cui unisco tramite INNER JOIN la tabella contenuto in cui è contenuto l'id del carrello associato all'utente e l'id dei prodotti contenuti in modo che seleziono tutti i dati in cui l'idCarrello è quello dell'utente loggato in questo momento ottenuto mediante la seguente Query.
$select = "SELECT idCarrello FROM utente WHERE idUtente=$idUtenteLoggato";
Successivamente tramite array associativo visualizzo i prodotti formattandoli graficamente.

```php
$totale = 0;
while ($riga2) {
    $idProdotto = $riga2['idProdotto'];
    $nome = $riga2['nome'];
    $prezzo = $riga2['prezzo'];
    $totale = $totale + $prezzo;

    $categoria = $riga2['idCategoria'];
    switch ($categoria) {
        case '1':
            $icona = "../../img/icone/telefono.png";
            break;
        case '2':
            $icona = "../../img/icone/cuffia.png";
            break;
        case '3':
            $icona = "../../img/icone/zaino.jpg";
            break;
    }

    echo "
    <div class='product ux-card'>
        <form action='./eliminaProdotto.php' method='POST'>
            <img src='$icona' height='50' width='50' />
            <span class='title'>$nome</span>
            <span class='price'>€ $prezzo</span>

            <input type='hidden' name='idCarrello' value='$idCarrello'>
            <input type='hidden' name='idProdotto' value='$idProdotto'>
            <button class='btn btn-canvas btn-xs remove' type='submit'><i class='material-icons'>delete</i> Rimuovi</button>
        </form>
    </div>
    ";
    $riga2 = mysqli_fetch_array($ris, MYSQLI_ASSOC);
};
```

Inserimento prodotti

Una volta premuto il bottone acquista tramite codice PHP i prodotti vengono inseriti nel seguente modo:
Per prima cosa ottengo l'idCarrello dell'utente loggato:
$idUtenteLoggato = $_SESSION["idUtenteLoggato"];
$ottieniIdCarrello = "SELECT idCarrello FROM utente WHERE idUtente='$idUtenteLoggato'";

Ora posso inserire nella tabella contenuto l'idCarrello appena ottenuto e l'idProdotto passato dal form
$insert = "INSERT INTO contenuto (idCarrello, idProdotto) VALUES ('$idCarrello', '$idProdotto')";

e nella tabella compra l'id dell'utente che ha effettuato l'acquisto e l'id del prodotto acquistato

$insert = "INSERT INTO compra (idUtente, idProdotto) VALUES ('$idUtenteLoggato', '$idProdotto')";

Eliminazione dei prodotti

Per eliminare un prodotto dal carrello mi salvo il suo id e l'id del carrello in cui è contenuto in due variabili tramite array super globale POST:

$idProdotto = $_POST['idProdotto'];
$idCarrello = $_POST['idCarrello'];

Ed esegue le seguenti 2 query per eliminare i riferimenti alle tabelle compra e contenuto:

$delete = "DELETE FROM contenuto where idProdotto=$idProdotto AND idCarrello=$idCarrello";
$delete2 = "DELETE FROM compra where idProdotto=$idProdotto AND idUtente=$idUtente";

Login

In fase di login una volta ricevuti i valori passati tramite metodo POST è stata effettuata la criptazione della password per effettuare il controllo con quella salvata nel database tramite la seguente Query.
$select = "SELECT * FROM utente WHERE email='$mail' and password='$passCript'";
utente identifica la tabella presa in considerazione, email e password identificano i campi di essa su cui verrà effettuato il controllo e $mail, $passCript rappresentano i valori inseriti dall'utente su cui va effettuata la verifica.
Successivamente ho controllato se l'utente fosse un admin attraverso il campo isAdmin, in questo caso l'ho reindirizzato nell'area riservata all'amministratore.

```php
$mail = $_POST['mail'];
$password = $_POST['password'];

$passCript = cryptPassword($password);

$select = "SELECT * FROM utente WHERE email='$mail' and password='$passCript'";

$ris = mysqli_query($conn, $select);
if (!$ris) {
?>
    <script>
        Swal.fire({
            icon: 'error',
            title: 'Ops...',
            text: 'Cè stato un problema in fase di login',
            allowEscape: false,
            allowOutsideClick: false,
            confirmButtonText: "<a href='../login/login.php'>Riprova</a>"
        });
    </script>
    <?php
} else {
    $riga = mysqli_fetch_row($ris);
    $admin = $riga[5];
    $conta = mysqli_num_rows($ris);
    if ($conta == 1) {
        $_SESSION['nome'] = $riga[1];
        $_SESSION['idUtenteLoggato'] = $riga[0];
        if ($admin == 1) {

            $_SESSION['admin'] = 1;
            $_SESSION['connessione'] = 1;
            header("Location: ./admin/admin.php");
        } else {
            $_SESSION['connessione'] = 1;
            header("Location: ../../index.php");
        }
    } else {
```

Logout

Dato che gestisco le sessioni in fase di logout dovrò terminarle, per far ciò uso le seguenti istruzioni:

```php
session_start();
// elimina le variabili di sessione impostate
$_SESSION = array();
// elimina la sessione
session_destroy();
header('Location: ../../index.php');
```

$_SESSION = array(); mi permette di sovrascrivere l'array di sessione con uno vuoto in modo da ripulirlo.
Fatto ciò termino la sessione con session_destroy();

Inserimento prodotti admin

Per inserire i prodotti passo tutti i dati tramite form, per indicare che si vuole passare un file, nel mio caso l'immagine del prodotto, bisogna specificarlo nel form nel seguente modo: enctype="multipart/form-data".
Una volta passati devo specificare la Path in cui voglio che venga caricata l'immagine $upload_path, mi devo salvare il nome del file $filename e accodarlo alla Path.
Successivamente eseguo un controllo per vedere se è già presente o se è troppo grande.

```php
$nome = $_POST['nome'];
$prezzo = $_POST['prezzo'];
$descrizione = $_POST['descrizione'];
$categoria = $_POST['categoria'];

$upload_path = "../../../img/prodotti/";
$filename = basename($_FILES['file']['name']);
$target_File = $upload_path . $filename;
$check = true;
$output = "";

echo $filename;
echo "<br>";
echo $target_File;

if (file_exists($target_File)) {
    $check = false;
    $output = "Il file esiste gia";
}

if ($_FILES['file']['size'] > 5000000) {
    $check = false;
    $output = "Il file è troppo grande";
}
```

Una volta passati i primi controlli verifico che sia stato effettivamente inserito ed eseguo la query per salvarlo nel database

```php
if ($check) {
    if (move_uploaded_file($_FILES['file']['tmp_name'], $target_File)) {
        $immagine = "./img/prodotti/" . $filename;

        include('../../database/config.php');
        $conn = mysqli_connect(DB_HOST, DB_USER, DB_PASS, DB_NAME);
        $ins = "INSERT INTO prodotto (nome, prezzo, descrizione, immagine, idCategoria) VALUES ('$nome','$prezzo','$descrizione','$immagine','$categoria

        $ris = mysqli_query($conn, $ins);
        if (!$ris) {
?>
            <script>
                Swal.fire({
                    icon: 'error',
                    title: 'Ops...',
                    text: 'Cè stato un problema in fase di visualizzazione',
                    allowEscape: false,
                    allowOutsideClick: false,
                    confirmButtonText: "<a href='../../../index.php'> Riprova</a>"
                })
            </script>
        <?php
            exit();
        } else {
            $_SESSION['inserito'] = 1;
            header('Location: ./admin.php');
        }
    } else {
        ?>
        <script>
            Swal.fire({
                icon: 'error',
                title: 'Ops...',
                text: 'Errore in fase di caricamento del file',
                allowEscape: false,
                allowOutsideClick: false,
                confirmButtonText: "<a href='./admin.php'> Riprova </a>"
            })
        </script>
    <?php
```

Categorie

Essendo i prodotti già divisi in categorie in fase di visualizzazione recupero l'id della categoria passandolo tramite array super globale POST ed eseguo la query per la visualizzazione.

```php
$idCategoria = $_POST['idCat'];

$select = "SELECT * FROM prodotto where idCategoria=$idCategoria order by 'id'";

$ris = mysqli_query($conn, $select);
if (!$ris) {
?>
    <script>
        Swal.fire({
            icon: 'error',
            title: 'Ops...',
            text: 'Cè stato un problema in fase di visualizzazione',
            allowEscape: false,
            allowOutsideClick: false,
            confirmButtonText: "<a href='./index.php'>Riprova</a>"
        })
    </script>
<?php
    exit();
}
```

Per la visualizzazione in maniera formattata uso il metodo mysqli_fetch_array() con un controllo finché il valore restituito è true.

```php
$i = 0;
while ($riga) {
    $id = $riga['idProdotto'];
    $nome = $riga['nome'];
    $prezzo = $riga['prezzo'];
    $descrizione = $riga['descrizione'];
    $path = $riga['immagine'];
    $immagine = "../.";
    $immagine = $immagine . $path;


    $riga = mysqli_fetch_array($ris, MYSQLI_ASSOC);



    echo "
    <div class='grid__item'>

    <form action='../carrello/insCarrello.php' method='POST'>
      <div class='Pcontainer'>
        <div class='Pimg'>
          <img src='$immagine'>
          <input type='hidden' name='idProdotto' value='$id' >
        </div>

        <div class='Player-Pcontainer'>
          <div class='Ptext-Pcontent'>
            <div>
              <p id='Pname'>$nome</p>
              <p class='Pcollection'>$descrizione</p>
            </div>
            <h3 id='Pprice'>€ $prezzo</h3>
            <input type='submit' class='Pcart-Pbtn' name='submitButton' value='Compra'>
            </a>
          </div>
        </div>
      </div>
    </form>
    </div>
```

Indirizzo

Avendo da trattare due diversi indirizzi (spedizione e fatturazione), non per forza presenti in contempera, ho diviso la query in due effettuando un controllo se il valore passato in fase di fatturazione fosse NULL.
In questo caso l'indirizzo di fatturazione non verrà inserito nel database.

```
$insert = "UPDATE utente SET via = '$via', cap = '$cap', nCiv = '$nCiv', comune = '$comune', provincia = '$provincia', info = '$info' WHERE idUtente = $
if (!mysqli_query($conn, $insert)) {
?>
  <script>
    Swal.fire({
      icon: 'error',
      title: 'Ops...',
      text: 'Cè stato un problema in fase di registrazione',
      allowEscape: false,
      allowOutsideClick: false,
      confirmButtonText: "<a href='../login/login.php'>Riprova</a>"
    });
  </script>
  <?php
}

if ($FragioneSociale != NULL) {
  $insert2 = "UPDATE utente SET FragioneSociale = '$FragioneSociale', FpecCod = '$FpecCod', FpartitaIva = '$FpartitaIva', Fvia = '$Fvia', Fcap = '$Fcap'
  if (!mysqli_query($conn, $insert2)) {
  ?>
    <script>
      Swal.fire({
        icon: 'error',
        title: 'Ops...',
        text: 'Cè stato un problema in fase di registrazione',
        allowEscape: false,
        allowOutsideClick: false,
        confirmButtonText: "<a href='../login/login.php'>Riprova</a>"
      });
    </script>
<?php
  }
```

```
$insert = "UPDATE utente SET via = '$via', cap = '$cap', nCiv = '$nCiv', comune = '$comune',
provincia = '$provincia', info = '$info' WHERE idUtente = $idUtente";

$insert2 = "UPDATE utente SET FragioneSociale = '$FragioneSociale', FpecCod = '$FpecCod',
FpartitaIva = '$FpartitaIva', Fvia = '$Fvia', Fcap = '$Fcap', FnCiv = '$FnCiv', Fcomune =
'$Fcomune', Fprovincia = '$Fprovincia', FcodFiscale = '$FcodFiscale'WHERE idUtente =
$idUtente";
```

Termini e condizioni

Per gestire le privacy policy mi sono appoggiato a Iubenda, un sito che permette di generare una privacy policy e una cookie policy per adeguare il sito a quelle che sono le normative di legge in materia di protezione dei dati personali degli utenti, durante la loro navigazione.

Conclusioni

Il progetto, nonostante alcune complicazioni, ha raggiunto la sua forma finale conforme a quanto richiesto.
I problemi principali sono emersi nella fase di realizzazione soprattutto legati alla richiesta di sistemi, nonostante ciò sono riuscito a trovare soluzioni e strade alternative per completare la richiesta.
Giunti alla fine posso constatare di aver imparato molto non solo dal punto di vista della realizzazione ma anche dal punto di vista della progettazione, sono sicuro che questa esperienza mi tornerà utile sia nel continuo del mio percorsi di studi sia in un successivo ambito lavorativo.