



UNIVERSITA' POLITECNICA DELLE MARCHE

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

Corso di Laurea triennale in Ingegneria Biomedica

**UN DATABASE PER SUPPORTARE LA DIAGNOSI DI TUMORE
ALLA CERVICE UTERINA**

A DATABASE TO SUPPORT THE DIAGNOSIS OF CERVICAL CANCER

Relatore: Chiar.mo/a

Prof. Piva Francesco

Tesi di Laurea di:

Lapponi Gabriele

Correlatore:

Prof. Giulietti Matteo

A.A. 2023 / 2024

Indice

Indice delle Figure	III
Abstract	IV
1. INTRODUZIONE.....	1
1.1 Tumore della cervice uterina	1
1.1.1 Anatomia.....	1
1.1.2 Cause e fattori di rischio	2
1.1.3 Screening e Diagnosi.....	4
1.2 Colposcopia	5
1.2.1 Generalità	5
1.2.2 Il Macchinario	6
1.2.3 La Tecnica.....	7
1.2.4 Problematiche.....	9
1.3 Intelligenza Artificiale (AI).....	10
1.3.1 Deep Learning in medicina	10
1.3.2 Importanza del Dataset	12
2. SCOPO DELLA TESI	13
3. MATERIALI E METODI	14
3.1 Linguaggi di front-end	15
3.1.1 HTML.....	15
3.1.2 CSS	16
3.2 Linguaggi di back-end: PHP	19
3.3 SQL e MySQL	19
3.4 XAMPP.....	20
4. RISULTATI.....	22
4.1 Database	22
4.2 Codice.....	25
4.2.1 Style.css.....	27
4.2.2 Connessione al database e configurazione sessione.....	28
4.2.3 Homepage.....	30
4.2.4 Signup, Login e Logout.....	33
4.2.5 Inserimento pazienti.....	42

4.2.6	Inserimento immagini.....	51
4.2.7	Galleria.....	62
4.2.8	Visualizza schede pazienti.....	66
4.2.9	Gestione account.....	70
4.2.10	Modifica.....	71
5.	CONCLUSIONI.....	75
	BIBLIOGRAFIA.....	77

Indice delle Figure

Figura 1. Apparato genitale femminile.....	1
Figura 2. Differenze tra epitelio non alterato (a sinistra) e infetto da HPV (a destra).....	3
Figura 3. Panoramica della progressione del cancro cervicale, da cellule normali a stato di cancro invasivo (Boon et al., 2022).	4
Figura 4. Raccomandazioni secondo il rischio di CIN di terzo grado (Prendiville et al., 2017).	5
Figura 5. Colposcopio Leisegang binoculare con schermo touch integrato (Baliga, 2018).....	6
Figura 6. Cervice dopo l'applicazione di acido acetico. Epitelio squamoso (A), cambiamenti aceto-bianchi con bordo netto (B), canale cervicale e giunzione squamo-colonnare (C), muco cervicale – vischioso (D).....	8
Figura 7. Zona di transizione (TZ) dopo l'applicazione di iodio di Lugol. L'epitelio maturo si colora di marrone indicato dalle frecce bianche) (Baliga, 2018).	8
Figura 8. Rispettivi ruoli di AI, ML e DL nella medicina (Ahmadzadeh Sarhangi et al., 2024).	11
Figura 9. Esempio di pagina scritta in linguaggio HTML.	17
Figura 10. Esempio di linguaggio CSS da utilizzare come foglio di stile per la pagina di esempio a fig.8.	18
Figura 11. Presentazione web del linguaggio HTML e CSS mostrato nelle figure 8 e 9.	18
Figura 12. Pannello di Controllo della piattaforma XAMPP dove sono evidenziati i tasti 'Start' per attivare i servizi Apache e MySQL. La scritta Apache in verde indica lo stato attivo del servizio.	20
Figura 13. Cartella all'interno della quale sono stati organizzati e salvati i file contenenti il codice e le immagini di test.	26

Abstract

Una delle principali cause di mortalità per malattie oncologiche nel sesso femminile è rappresentata dal cancro della cervice uterina. Questo tipo di tumore può essere diagnosticato attraverso l'utilizzo di vari test di screening; qualora uno di questi test dia un risultato positivo, si procede con una colposcopia (svolta da un esperto nel settore) per una valutazione più approfondita. Tuttavia, la diagnosi definitiva e certa di cancro al collo dell'utero può essere ottenuta esclusivamente mediante una biopsia cervicale, che permette di analizzare il tessuto prelevato e confermare la presenza di cellule tumorali.

Essendo la biopsia un esame invasivo e costoso, si cerca di svolgerlo nei soli casi in cui il colposcopista è quasi sicuramente certo, vedendola tramite la colposcopia, della presenza di una neoplasia.

La problematica principale associata a questo tipo di diagnosi risiede nell'interpretazione dell'esame colposcopico; questo, infatti, non è standardizzato e dipende in larga scala dell'esperienza e manualità dell'operatore che lo effettua.

Sempre più di frequente negli ultimi anni, per ovviare a problematiche di questo tipo, in ambito medico si sta cercando di sviluppare software avanzati e reti neurali artificiali per rendere più agevoli i meccanismi di diagnosi precoce. Uno dei fattori principali per la ben riuscita di questi speciali software è sicuramente la quantità e la qualità dei dati a disposizione. Con questo progetto di tesi, si è cercato di creare un database, con sito web annesso, per la raccolta e la categorizzazione di immagini colposcopiche provenienti da alcuni centri ospedalieri italiani, al fine di poter aiutare la ricerca e lo sviluppo nel campo di diagnosi precoce di tumore alla cervice uterina.

Nelle pagine successive sarà presentata un'introduzione sul carcinoma cervicale, esaminandone alcune caratteristiche. Si fornirà inoltre una descrizione della procedura colposcopica, evidenziandone le problematiche. Verranno espone anche nozioni base su cosa sia l'intelligenza artificiale e come questa potrebbe potenzialmente migliorare l'accuratezza e l'efficacia della diagnosi precoce, approfondendo l'importanza di un vasto dataset.

Come ultimi, verranno infine esposti metodi, materiali e il linguaggio di codifica con cui sono stati creati il database e l'interfaccia web per l'inserimento delle immagini relative agli esami di colposcopia, con la speranza che il tutto possa essere di aiuto alla comunità scientifica.

1. INTRODUZIONE

1.1 Tumore della cervice uterina

Il carcinoma della cervice uterina è il quarto tumore più frequentemente diagnosticato e la quarta causa principale di mortalità oncologica tra le donne a livello globale. Nel solo anno 2020, si stima che siano stati registrati più di 600.000 nuovi casi di carcinoma della cervice uterina e all'incirca 342.000 decessi dovuti a questo tipo di cancro (Newaz et al., 2022).

Più dell'85% dei nuovi casi viene riscontrato in individui economicamente svantaggiati e quali il 90% dei decessi per cancro cervicale avviene nelle regioni del mondo con scarse risorse economiche (Bermudez et al., 2015).

1.1.1 Anatomia

L'utero è un organo oblunco composto da fondo, corpo, istmo e cervice. Le tube di Falloppio si inseriscono in ciascun angolo supero-laterale, al di sopra dei quali si trova il fondo. La cervice è avvolta dalla vagina, formando una parte sopra-vaginale e una parte vaginale (Ellis, 2011).

Il canale uterino attraversa l'orifizio interno e si apre come orifizio esterno nella volta vaginale. Il corpo uterino è flesso sulla cervice (ante-flessione), mentre l'intero utero è inclinato in avanti (ante-versione)(Ellis, 2011)(Fig. 1).

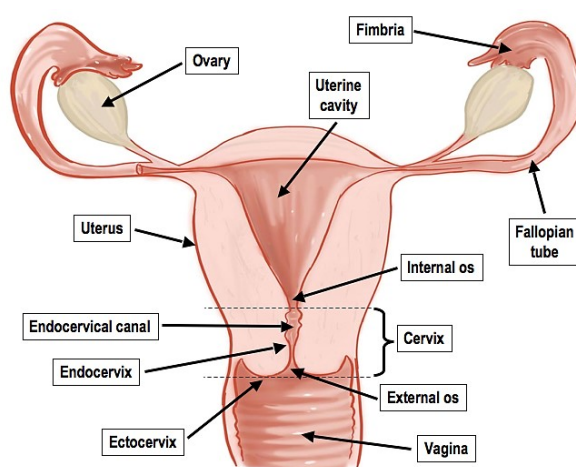


Figura 1. Apparato genitale femminile.

La cervice rappresenta la porzione inferiore dell'utero e si distingue per la sua forma approssimativamente cilindrica; si proietta attraverso la parete superiore-anteriore della vagina, stabilendo una comunicazione con essa mediante il canale endocervicale (Bermudez et al., 2015).

Il cancro della cervice può insorgere in diversi punti: la mucosa che riveste la superficie della cervice stessa oppure svilupparsi all'interno del canale endocervicale. Una volta insediato, il carcinoma della cervice uterina tende a crescere localmente e può estendersi coinvolgendo l'utero, i tessuti paracervicali circostanti e gli organi pelvici. Inoltre, questo tumore, può diffondersi ai linfonodi regionali. Solo in una fase più avanzata della malattia, le cellule cancerogene possono metastatizzare, diffondendosi per via ematogena a strutture anatomiche distanti (Bermudez et al., 2015).

1.1.2 Cause e fattori di rischio

La principale causa del carcinoma della cervice uterina è il Papillomavirus umano (HPV), mentre i principali fattori di rischio includono il fumo, l'infezione da HIV, il trapianto di organi, una storia familiare con tumore al collo dell'utero alle spalle e l'uso continuativo di contraccettivi orali (Newaz et al., 2022).

I virus del papilloma umano costituiscono una famiglia di piccoli virus a DNA che hanno la capacità di replicarsi in siti anatomici specifici all'interno dell'organismo. La loro struttura è caratterizzata da una semplicità sorprendente, essendo composti solamente da un numero limitato di geni. In particolare, si distinguono sei geni precoci e due tardivi (Reich & Regauer, 2023).

Replicazione e trascrizione del virus richiedono proteine che sono codificate nella regione chiamata precoce (Early, E) E1, E2, E4, E5, E6 e E7. Nella regione tardiva (Late, L) sono invece presenti i geni codificanti le proteine strutturali del capsido L1 e L2, che si occupano dell'assemblaggio del virione. La lunga regione di controllo (LCR) o regione regolatoria a monte (URR) è una sezione non codificante che costituisce la terza regione del genoma dell'HPV. Contiene elementi cis che includono una sequenza regolatoria trascrizionale, un'origine di replicazione del DNA e uno o più promotori che regolano l'espressione delle

oncoproteine E6 ed E7. Un'aumentata trascrizione degli oncogeni E6 ed E7 è una caratteristica dei carcinomi cervicali invasivi ed è essenziale per lo sviluppo del cancro causato dall'HPV (Endale et al., 2024) (Fig. 2).

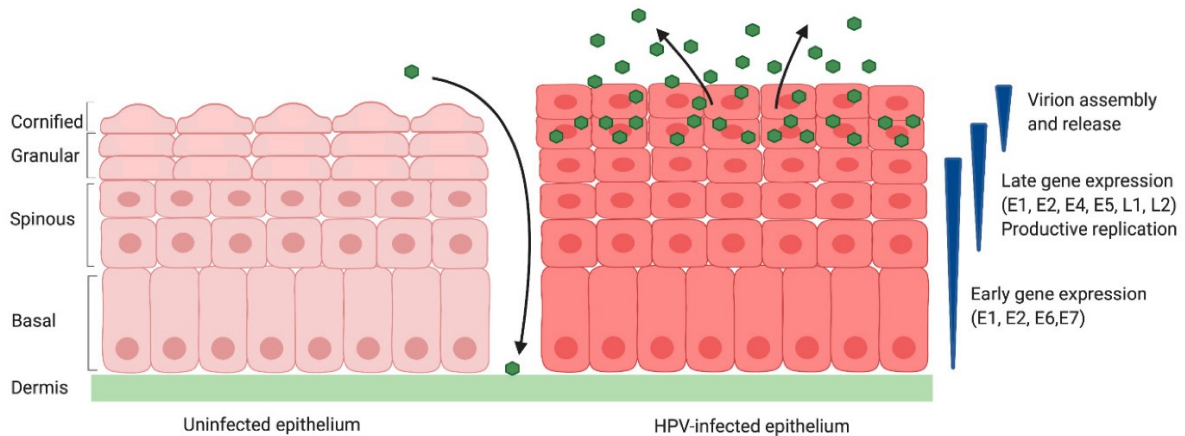


Figura 2. Differenze tra epitelio non alterato (a sinistra) e infetto da HPV (a destra).

Nel processo dello sviluppo del cancro cervicale, le anomalie nelle cellule squamose, conosciute come Neoplasia Intraepiteliale Cervicale (CIN), sono strettamente legate all'infezione da specifici tipi di HPV ad alto rischio (Ahmadzadeh Sarhangi et al., 2024).

Benché esistano più di cento tipi di Papillomavirus umano, i due classificati come ad alto rischio sono gli HPV 16 e 18, riconosciuti come i principali agenti causali del cancro cervicale (Sambyal & Sarwar, 2023).

Sebbene non tutti i casi di CIN progrediscono a cancro cervicale, alcuni di grado elevato non trattato potrebbero portare, nel peggiore dei casi, a un cancro cervicale invasivo (Ahmadzadeh Sarhangi et al., 2024).

Come affermato dall'Organizzazione Mondiale della Sanità (OMS), sono presenti tre diverse tipologie di CIN. La neoplasia CIN1 può essere categorizzata come LSIL (Lesione Intraepiteliale Squamosa di Basso Grado), mentre la CIN2 e CIN3 possono essere categorizzate come HSIL (Lesione Intraepiteliale Squamosa di Alto Grado) (Ahmadzadeh Sarhangi et al., 2024) (Fig. 3).

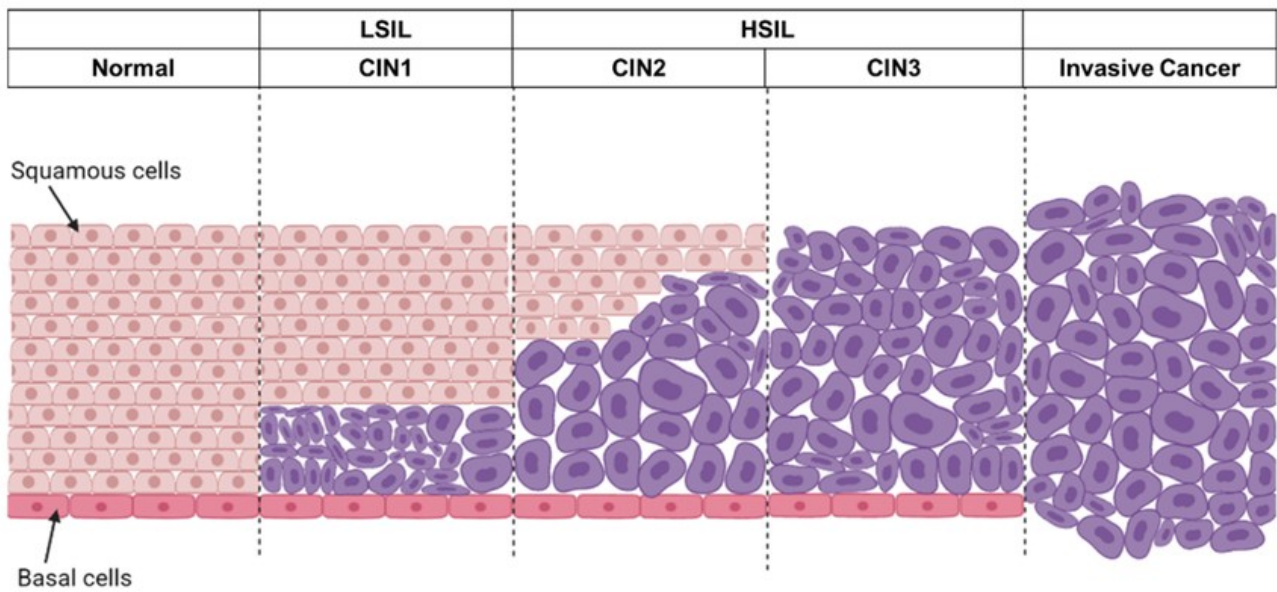


Figura 3. Panoramica della progressione del cancro cervicale, da cellule normali a stato di cancro invasivo (Boon et al., 2022).

1.1.3 Screening e Diagnosi

Un carcinoma di questo tipo può essere rilevato tramite vari test di screening, come l'Ispezione Visiva con Acido Acetico (VIA), la citologia (Pap test e citologia in fase liquida), i test HPV e la colposcopia. Questi test sono progettati per individuare il CIN nella cervice di donne in apparenza sane e asintomatiche, il che può indicare la presenza di lesioni precancerose o altri potenziali rischi per il cancro cervicale (Ahmadzadeh Sarhangi et al., 2024).

Gli attuali test di screening per il pre-cancro cervicale non sono completamente sensibili né assolutamente specifici. Ad esempio, il test per l'HPV ad alto rischio individua quasi tutte le lesioni precancerose alla cervice, ma risulta spesso positivo anche un donne con infezioni HPV ad alto rischio innocue e transitorie (Prendiville et al., 2017).

La citologia, d'altro canto, è molto più specifica rispetto al test discusso sopra, ma non riesce a individuare la totalità delle donne a rischio di pre-cancro. È evidente che il test ideale, non attualmente esistente, sarebbe in grado di individuare solo le donne ad alto rischio di sviluppare questo tipo di carcinoma (Prendiville et al., 2017).

Nel caso in cui il test di screening è positivo per il CIN, allora è necessario confermare la diagnosi con una colposcopia seguita da biopsie mirate, di solito eseguite in ambiente

sanitario da un ginecologo o un altro operatore qualificato, utilizzando un colposcopio (Ahmadzadeh Sarhangi et al., 2024).

1.2 Colposcopia

Nel contesto della tradizionale esame colposcopica, la diagnosi di Neoplasia Intraepiteliale Cervicale (CIN) e di cancro cervicale, si basa principalmente su una valutazione soggettiva, dipendendo fortemente dall'esperienza e dalle competenze dell'operatore. Tuttavia, l'accuratezza e l'efficienza nel rilevare lesioni cervicali e cancro invasivo precoce sono limitate. Per interpretare correttamente le immagini acquisite, è fondamentale che il clinico possieda una solida base di conoscenze e una vasta esperienza, le quali richiedono molto tempo per l'apprendimento e un costante aggiornamento professionale (Dong et al., 2022).

1.2.1 Generalità

Un esame colposcopico può essere utilizzato per esaminare qualsiasi superficie epiteliale del tratto genitale inferiore. Alcune delle situazioni in cui si raccomanda la colposcopia sono illustrate nella figura 4.

Risk of CIN3	Recommended action
> 5%	Immediate referral for colposcopy
2–5%	6–12-month follow-up
0.1–2%	3-year follow-up
< 0.1%	5-year follow-up

CIN3, cervical intraepithelial neoplasia grade 3.

Figura 4. Raccomandazioni secondo il rischio di CIN di terzo grado (Prendiville et al., 2017).

Può essere impiegata come strumento di screening primario, anche se non è molto efficace in questo caso, e per facilitare diverse modalità di trattamento. La colposcopia viene utilizzata per esaminare la vulva, l'ano, la vagina e più recentemente l'orofaringe, così come l'epitelio penieno, poiché ciascuno di questi siti è soggetto allo sviluppo di lesioni precancerose riconoscibili colposcopicamente. Tuttavia, la grande maggioranza degli esami colposcopici riguarda la cervice con sospetto di pre-cancro (Prendiville et al., 2017).

Prima dell'esame colposcopico, è essenziale raccogliere la storia clinica della paziente, preferibilmente utilizzando un modulo specifico per la clinica che effettua l'esame.

Le informazioni da ottenere includono dettagli su mestruazione, contraccezione, gravidanze, abitudine al fumo, precedenti esami citologici cervicali, sintomi, trattamenti precedenti e data dell'ultimo ciclo mestruale. Alla paziente dovrebbe inoltre essere fornita una spiegazione completa riguardo all'esame colposcopico e del materiale informativo scritto prima della visita al reparto di colposcopia (Shafi & Nazeer, 2012).

1.2.2 Il Macchinario

Il colposcopio (Fig. 5) è un dispositivo composto da un microscopio binoculare e una sorgente luminosa, è relativamente semplice e permette l'osservazione della cervice sotto una luce intensa; la variabilità di ingrandimento a bassa potenza consente una valutazione dettagliata delle strutture cervicali. Il macchinario è anche spesso dotato di uno splitter ottico per consentire il collegamento di una fotocamera o videocamera (Prendiville et al., 2017).



Figura 5. Colposcopio Leisegang binoculare con schermo touch integrato (Baliga, 2018).

Esistono diverse tipologie di colposcopi disponibili nel mercato. Per la scelta dell'attrezzatura colposcopica adatta al proprio ambiente specifico bisognerebbe valutare alcuni criteri:

- Costo
- Qualità ottica – Riguarda luminosità, nitidezza e uniformità di illuminazione nonché la possibilità di utilizzare un filtro verde. Più elevata è la qualità dell'ottica, migliore sarà la resa dell'immagine colposcopica
- Ingrandimento dello zoom
- Design – Dovrebbe consentire al braccio colposcopico di muoversi in maniera fluida e stabile
- Oculare – Confortevole nell'uso. Molti hanno un'opzione di correzione diottrica per soggetti con miopia e/o ipermetropia
- Lunghezza focale – Di solito fissa, anche se in alcuni può essere variabile di qualche grado, permettendo all'esperto di posizionarsi ottimamente per la procedura
- Illuminazione
- Tipologie di supporti – Ad esempio autoportanti, da parete o da soffitto
- Ambiente operativo – Assicurarsi che lo strumento funzioni in maniera adatta considerando temperatura e umidità
- Accessori – Facilità di registrazione e visualizzazione con telecamere e monitor (Shafi & Nazeer, 2012)

1.2.3 La Tecnica

Tramite l'esame colposcopico, l'esperto dovrebbe essere in grado di comprendere e documentare diversi aspetti della situazione clinica attuale della paziente:

- 1) Riconoscere se sono presenti anomalie dell'epitelio
- 2) Documentare i risultati degli esami sopra scritti con formato standard di nomenclatura e terminologia IFCPC (International Federation of Cervical Pathology and Colposcopy)
- 3) Se possibile, ottenere più immagini possibile dei risultati degli esami fatti così da registrare

- a) Grandezza e tipologia della zona di transizione (TZ)
- b) Le zone con anomalità
- c) Le zone di prelievo per biopsia, se presenti
- d) Il tipo trattamento utilizzato, se presente (Prendiville et al., 2017)

Per una valutazione colposcopica adatta, è una pratica usuale applicare una soluzione di acido acetico (AA) (3% o 5%) sulla cervice anche se, sorprendentemente, non ci sono ancora indicazioni precise su come eseguire precisamente il test con acido acetico per ottenere risultati ottimali (Hilal et al., 2020).

L'acido acetico aiuta a differenziare le aree con atipicità che dovrebbero essere mappate dal medico tramite apparecchiature per la registrazione di immagini (Fig. 6). Tali immagini dovrebbero essere poi conservate digitalmente o in formato cartaceo.

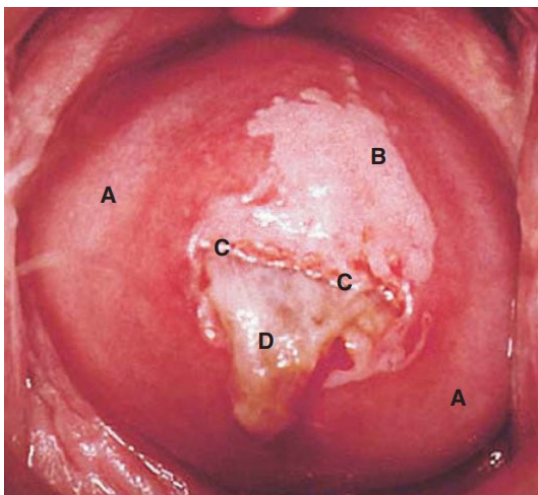


Figura 6. Cervice dopo l'applicazione di acido acetico. Epitelio squamoso (A), cambiamenti aceto-bianchi con bordo netto (B), canale cervicale e giunzione squamo-colonnare (C), muco cervicale – vischioso (D) (Baliga, 2018).

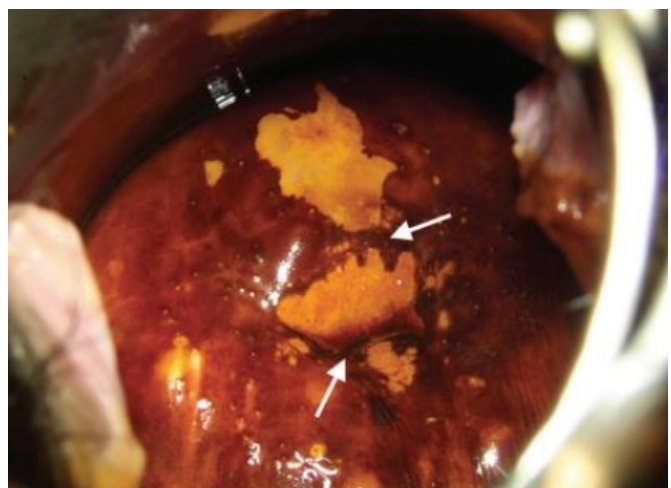


Figura 7. Zona di transizione (TZ) dopo l'applicazione di iodio di Lugol. L'epitelio maturo si colora di marrone indicato dalle frecce bianche) (Baliga, 2018).

Anche lo iodio di Lugol (1% iodio, 2% ioduro di potassio, 97% acqua distillata) può essere utilizzato per delineare ulteriormente l'epitelio atipico che contiene poco o nessun glicogeno e quindi non assorbe il colore tipico della soluzione. In questo caso, l'epitelio squamoso normale assume una colorazione marrone mogano con lo iodio di Lugol. Allo stesso modo, l'epitelio colonnare contiene poco o nessun glicogeno e non assorbe la colorazione. Il test in questione è noto come "test di Schiller" (Fig. 7). Subito dopo la

valutazione, i risultati devono essere registrati, preferibilmente seguendo un formato standard (Shafi & Nazeer, 2012).

1.2.4 Problematiche

La capacità di eseguire esami adeguati, sicuri e accurati richiede inoltre un'elevata competenza negli aspetti tecnici, interpretativi e cognitivi, oltre alla capacità di sviluppare piani di gestione e trattamento pragmatici ed efficaci. In sintesi, le competenze necessarie sono molteplici e complesse (Prendiville et al., 2017).

Come per qualsiasi test soggettivo, dipende anche dalla competenza e dall'esperienza del colposcopista, nonché dalla prevalenza della malattia. Inoltre, la precisione diagnostica è influenzata dalla conoscenza del risultato dello screening di riferimento. L'esame colposcopico risulta più efficace quando si confrontano gli estremi dell'anormalità: normale/basso grado e alto grado definito; mentre la sua capacità di diagnosticare con precisione è più limitata nei casi di anormalità di grado intermedio o equivoco. Diversi studi hanno evidenziato in effetti che in alcune circostanze la colposcopia non ha fornito prestazioni ottimali (Prendiville et al., 2017).

Le biopsie colposcopiche dirette potrebbero essere necessarie a volte, ma non sempre. Per la maggior parte delle donne con un'indicazione colposcopica di CIN3 e un prelievo di alto grado, è consigliabile l'escissione della zona di transizione anziché una biopsia diretta. Quando un laboratorio di qualità riporta un prelievo positivo come CIN3, la colposcopia solitamente conferma una lesione di alto grado. In caso contrario, il colposcopista dovrebbe valutare una biopsia diretta, consultando anche il laboratorio di riferimento prima di decidere il trattamento. È preferibile però eseguire una colposcopia accurata piuttosto che affidarsi a biopsie casuali, poiché queste ultime raramente individuano lesioni di alto grado in un esame colposcopico normale (Prendiville et al., 2017).

Una delle soluzioni per rendere la colposcopia più efficiente potrebbe essere l'utilizzo di sistemi di supporto decisionale.

Fin dagli anni '90, i sistemi di Diagnosi Assistita da Computer (CAD) hanno migliorato notevolmente l'efficienza e la qualità dell'elaborazione delle immagini mediche.

Con il tempo, l'Intelligenza Artificiale (AI), e in particolar modo il Deep Learning (DL), hanno raggiunto risultati sempre più significativi in questo settore (Dong et al., 2022).

1.3 Intelligenza Artificiale (AI)

Da una veloce ricerca sulla letteratura, si nota subito che nel corso degli ultimi anni è aumentato esponenzialmente il numero di articoli presenti sull'analisi del cancro, soprattutto il numero di quelli che coinvolgono strumenti di intelligenza artificiale e grandi collezioni di dati riguardanti casi clinici storici per l'addestramento di modelli di AI. Gli studi mostrano inoltre che l'accuratezza diagnostica, prognostica e predittiva possono significativamente migliorare quando l'AI è utilizzata simultaneamente a complessi strumenti bioinformatici (Huang et al., 2020).

Machine Learning (ML) e Deep Learning (DL) si distinguono come parti principali nel vasto campo dell'intelligenza artificiale (Ahmadzadeh Sarhangi et al., 2024).

Diversamente dai tradizionali programmi software, il ML usa grandi quantità di dati per addestrare e poi applicare algoritmi così da imparare come alcuni compiti possono essere svolti. Il DL invece non è un metodo di apprendimento indipendente, per addestrare una rete di deep learning ci sono due modi: uno supervisionato e uno non supervisionato. In parole povere, il ML viene utilizzato per realizzare l'AI, mentre il DL, è utilizzato per implementare il ML (Huang et al., 2020).

1.3.1 Deep Learning in medicina

Uno degli ostacoli più grandi nel machine learning è la ricerca e l'apprendimento di caratteristiche e informazioni che descrivono in maniera adatta le regolarità o i modelli inerenti ai dati. Il deep learning supera questo ostacolo eliminando la necessità di estrarre manualmente le caratteristiche: utilizzando semplicemente insiemi di dati con minima preelaborazione, è in grado di individuare autonomamente le peculiarità delle immagini. L'onere dell'ingegnerizzazione delle caratteristiche si è spostato quindi dagli esseri umani ai computer, consentendo anche a persone non esperte di ML di poter usufruire efficacemente del DL per le proprie ricerche e/o applicazioni, soprattutto nell'analisi delle immagini mediche (Fig. 8) (Shen et al., 2017).

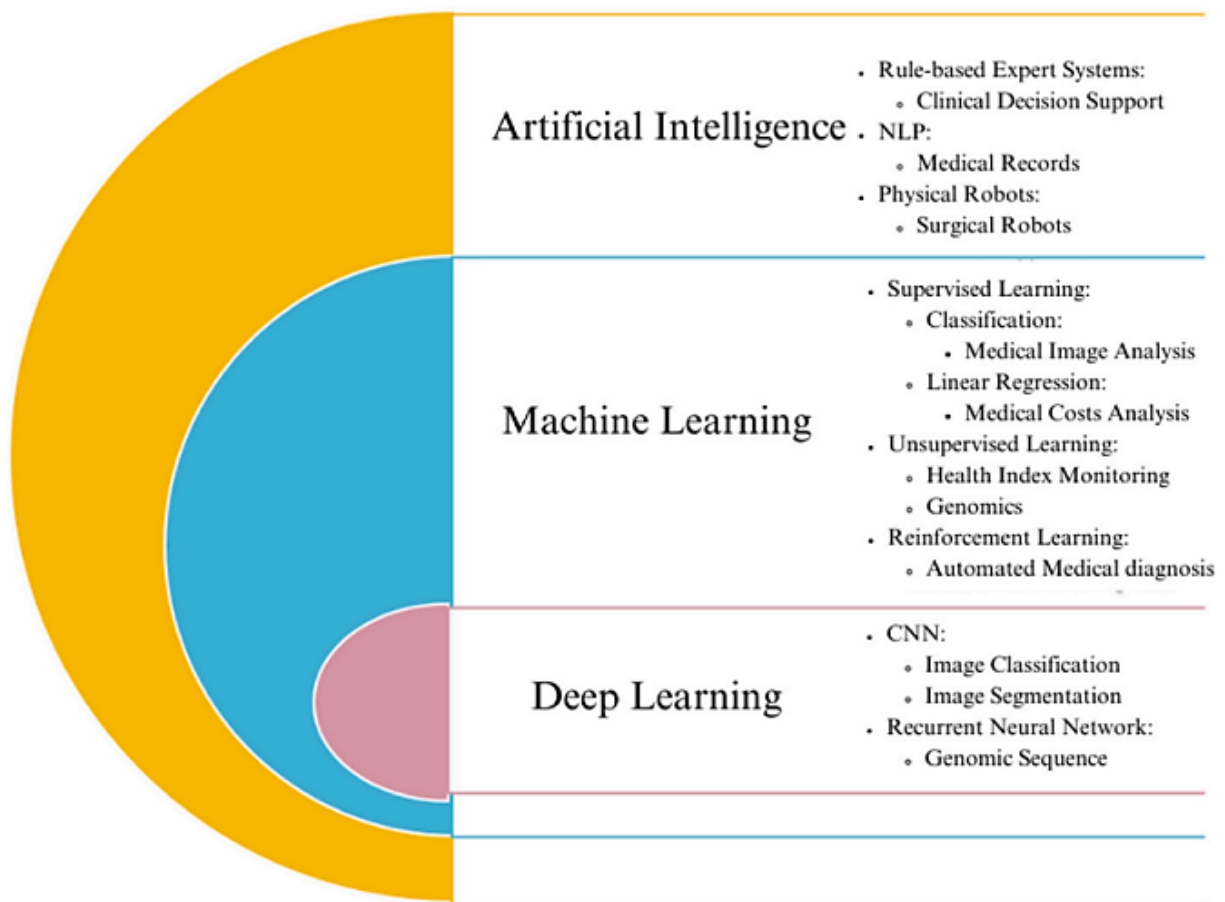


Figura 8. Rispettivi ruoli di AI, ML e DL nella medicina (Ahmadzadeh Sarhangi et al., 2024).

Gli algoritmi di deep learning sono stati applicati a pratiche in numerose specialità mediche e in diversi casi hanno raggiunto prestazioni paragonabili a quelle di esperti umani. È infatti possibile che il DL possa essere sfruttato per estrarre dalle immagini mediche informazioni che non sarebbero immediatamente evidenti con la sola analisi umana e che potrebbero essere decisive per informare sullo stato molecolare, sulla prognosi o sulla sensibilità del trattamento. Alcuni studi riguardanti la rilevazione di melanomi, coinvolgendo dermoscopia e immagini, mostrano infatti un confronto delle prestazioni di prognosi e diagnosi tra metodi di AI e patologi. Viene evidenziato come le prestazioni diagnostiche delle Convolutional Neural Network (CNN) sono risultate superiori a quelle della maggior parte dei dermatologi, anche se non di tutti (Huang et al., 2020).

Grazie alla capacità di comprendere le rappresentazioni gerarchiche delle caratteristiche direttamente dai dati, il DL ha ottenuto e continua ad ottenere risultati eccezionali in molte applicazioni di intelligenza artificiale. Un caso particolarmente importante riguarda i progressi nel campo della visione artificiale, i quali promuovono il suo impiego nell'analisi

di immagini mediche, che include la segmentazione, la registrazione, la fusione di immagini, così come l'annotazione, la diagnosi e la prognosi assistite da computer, per citarne alcune (Shen et al., 2017).

1.3.2 Importanza del Dataset

Il successo del deep learning deriva principalmente dai fattori seguenti:

- I progressi fatti per quanto riguarda le unità di elaborazione centrale (CPU) e quelle di elaborazione grafica (GPU) dei computer ad alti livelli tecnologici
- La disponibilità di lavorare con grandi set di dati
- Lo sviluppo di algoritmi di apprendimento (Shen et al., 2017).

Un classico approccio di deep learning comprende quattro fasi di elaborazione. La fase iniziale consiste nella raccolta dei dati (in questo caso di immagini), che è fondamentale nel processo di apprendimento (Ahmadzadeh Sarhangi et al., 2024).

I metodi di deep learning sono infatti efficaci quando il numero di campioni disponibili è alto durante la fase di addestramento della rete. Come esempio, lo studio di Shen et al. (2017) riporta la ImageNet Large Scale Visual Recognition Challenge (ILSVRC) dove sono state fornite più di un milione di immagini annotate per addestrare una CNN.

I modelli di DL traggono in effetti vantaggio da set di dati più ampi possibile, e la carenza di dati sufficienti può quindi compromettere le prestazioni degli algoritmi. Purtroppo, la scarsità di grandi dataset è particolarmente evidente nell'imaging medico, specialmente per le immagini di citologia e colposcopia (Ahmadzadeh Sarhangi et al., 2024).

Una delle principali sfide nell'utilizzo del deep learning allo studio di immagini mediche deriva di conseguenza dal limitato numero di campioni utilizzato per costruire modelli complessi e funzionali senza incorrere in problemi come l'overfitting (Shen et al., 2017).

Un ulteriore studio che conferma l'importanza del Dataset quando si parla di DL è quello di Dong et al. (2022) dove gli autori riconoscono che una delle limitazioni dello studio è espressamente "Il Dataset è ancora relativamente piccolo per il deep learning".

2. SCOPO DELLA TESI

L'obiettivo di questo studio è la creazione di un database a livello nazionale, mirato a raccogliere un vasto campione di immagini colposcopiche provenienti da diverse strutture ospedaliere distribuite su tutto il territorio italiano.

L'intento del progetto è inoltre quello di popolare il database con immagini mediche di alta qualità, eterogenee e dettagliatamente annotate con varie caratteristiche, così che possano essere utilizzate in futuro per addestrare delle reti neurali artificiali, migliorando l'accuratezza e l'affidabilità dei sistemi diagnostici.

L'approccio prevede la raccolta di un ampio e rappresentativo campione di dati, elemento cruciale per ottimizzare le performance dell'intelligenza artificiale nella diagnosi del tumore alla cervice uterina. Un database ricco e diversificato di immagini consente ai modelli di apprendere meglio e di generalizzare in modo efficace, riducendo i tassi di errore e incrementando la sua precisione.

Al tempo stesso, è previsto lo sviluppo di un interfaccia web dedicata, progettata per facilitare l'inserimento delle immagini in maniera chiara e controllata. Questa interfaccia utilizzerà dei sistemi di autenticazione tramite login per garantire che solamente utenti autorizzati possano caricare e annotare le immagini. Ogni fotografia sarà etichettata con caratteristiche specifiche, permettendo una catalogazione precisa e una facile reperibilità delle informazioni necessarie per il sistema di addestramento.

Questa struttura integrata non solo migliorerà la qualità e la coerenza dei dati raccolti, ma anche l'efficienza del processo di inserimento e gestione delle immagini. L'obiettivo finale è di creare uno strumento potente e affidabile con il compito di supportare la comunità medica nella diagnosi precoce e accurata del tumore alla cervice uterina, contribuendo in modo significativo alla lotta contro questa malattia.

3. MATERIALI E METODI

Gli obiettivi esaminati nel capitolo precedente possono essere perseguiti attraverso l'aiuto di specifici software per lo sviluppo di interfacce web e la creazione di un database relazionale, destinato all'archiviazione dei dati in modo strutturato ed efficiente. L'adozione di tali strumenti consente non solo di aumentare la facilità di scrittura del codice ma anche di garantire una gestione ottimale delle informazioni.

Oltre a questo, non è da dimenticare l'indispensabile utilizzo di piattaforme che forniscono servizi di data center e web hosting. Queste piattaforme sono di fondamentale importanza per assicurare la sicurezza e l'affidabilità dei sistemi sviluppati, offrendo strutture robuste e flessibili in grado di adattarsi rapidamente alle esigenze variabili degli utenti.

Per la realizzazione del progetto è stato utilizzato un PC con sistema operativo Microsoft Windows.

Riguardo all'interfaccia web, sono stati impiegati i linguaggi HTML e CSS.

L'HTML (HyperText Markup Language) viene utilizzato per strutturare il contenuto della pagina, definendo elementi quali testi, immagini, links e forms. Il CSS (Cascading Style Sheet) è invece impiegato per la definizione dell'aspetto grafico ed estetico della pagina.

Il database è stato progettato utilizzando MySQL, un sistema di gestione di database relazionali. Sono state create diverse tabelle interrelate per contenere dati riguardanti utenti, pazienti e immagini. Questo approccio consente una gestione organizzata ed efficiente delle informazioni, facilitando l'accesso, la manipolazione e l'analisi dei dati.

Per garantire la comunicazione tra l'interfaccia web e il database, è stato fondamentale l'utilizzo del linguaggio PHP.

PHP (Hypertext Preprocessor) è un linguaggio di scripting lato server (back-end) che permette di generare contenuti dinamici e interagire con i database, facilitando così l'elaborazione dei dati provenienti dall'interfaccia web e il loro inserimento, aggiornamento e recupero nel database.

Per aiutare tutte le attività di scrittura del codice, è stato sfruttato l'editor di testo Sublime Text.

Infine, per quanto riguarda la connessione al server e al database, è stata utilizzata la piattaforma software XAMPP.

3.1 Linguaggi di front-end

I linguaggi di front-end sono strumenti essenziali nello sviluppo web, utilizzati per creare e gestire l'interfaccia utente di un sito web. L'interfaccia utente è il punto di interazione tra un utente e un sistema informatico. Include tutti gli elementi visivi, interattivi e funzionali attraverso i quali l'utente può comunicare con il sistema per eseguire operazioni, visualizzare informazioni e gestire funzionalità. Tramite i principali linguaggi di front-end HTML e CSS, l'obiettivo è di rendere l'interazione dell'utente più intuitiva, efficace e piacevole possibile.

3.1.1 HTML

HTML è il linguaggio di markup più utilizzato per creare le pagine web. Definisce la struttura e il contenuto della pagina, utilizzando elementi come titoli, paragrafi, immagini, link e form. Gli elementi HTML sono rappresentati da dei marcatori (tag), che delineano l'inizio e la fine di ogni parte di contenuto. Ogni tag ha uno scopo specifico e viene interpretato dai browser web per visualizzarne il contenuto in maniera corretta. Questi marcatori si dividono in diverse categorie:

1) Tag di Base

- a) `<html>` Definisce l'inizio e la fine del documento.
- b) `<head>` Contiene informazioni di intestazione come il titolo della pagina e i collegamenti a fogli di stile CSS.
- c) `<body>` Contiene il contenuto principale visibile della pagina web.

2) Tag di Struttura

- a) `<div>` o `` Definisce una sezione della pagina, viene utilizzato per raggruppare e stilizzare i contenuti.

3) Tag di Testo

- a) Da `<h1>` a `<h6>` Definiscono i titoli con diversi livelli di importanza (da più grande a più piccolo).
- b) `<p>` Definisce un semplice paragrafo di testo.
- c) `<a>` crea un link ipertestuale a un'altra pagina web o a una parte diversa della stessa pagina.

4) Tag Multimediali

- a) `` `<video>` `<audio>` Incorporano rispettivamente un'immagine, un video o un file audio nella pagina web.

5) Tag di lista

- a) `` Crea una lista non ordinata.
- b) `` Crea una lista ordinata.
- c) `` Definisce un elemento della lista, sia in liste ordinate che non ordinate.

6) Tag di Form

- a) `<form>` Crea un modulo per la raccolta di dati dall'utente.
- b) `<input>` Definisce un campo di input per il modulo, come caselle di testo, pulsanti, ecc.
- c) `<textarea>` Crea un campo di testo multilinea all'interno di un modulo.

7) Tag di Stile e Script

- a) `<style>` Definisce stili CSS direttamente nella pagina HTML.
- b) `<script>` Incorpora script JavaScript nella pagina HTML.

3.1.2 CSS

CSS è il linguaggio utilizzato per descrivere l'aspetto e la formattazione di un documento HTML. Con CSS è possibile controllare il layout, i colori, i caratteri e altri aspetti estetici della pagina web, separando così la presentazione dal contenuto per avere una struttura meglio organizzata.

Questo tipo di linguaggio consente di applicare stili in maniera efficiente sia a singoli elementi che a gruppi di elementi, utilizzando selettori e dichiarazioni di stile. Un esempio potrebbe essere impostare il colore di tutte le intestazioni '`<h1>`' su una pagina o definire un layout per una form che si adatta a diverse dimensioni dello schermo se ci si aspetta che la pagina possa essere visitata da dispositivi differenti come monitor, smartphone o tablet. Nelle figure 9, 10 e 11 possiamo vedere esempi di come è scritta una pagina HTML con relativo codice CSS.

```
esempio.html x
1 <!DOCTYPE html>
2 <html lang="it">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>Pagina di Esempio</title>
7     <style type="text/css"></style>
8 </style>
9     <link rel="stylesheet" href="style.css">
10 </head>
11 <body>
12
13 <header>
14     <h1>Benvenuto nella mia Pagina di Esempio</h1>
15 </header>
16
17 <main>
18     <h2>Introduzione</h2>
19     <p>Questa è un semplice paragrafo HTML.</p>
20
21     <a href="#"> Questo è un link <a>
22
23     <h2>Sezione Contenuti</h2>
24     <p> Ecco un elenco puntato di esempio:</p>
25     <ul>
26         <li>Elemento 1</li>
27         <li>Elemento 2</li>
28         <li>Elemento 3</li>
29     </ul>
30
31     <h2>Immagine di Esempio</h2>
32     <p>Qui sotto è mostrata un'immagine di esempio.</p>
33     
35 </main>
36
37 <footer>
38     <p>&copy; 2024 Pagina di Esempio. Tutti i diritti riservati.</p>
39 </footer>
40
41 </body>
42 </html>
43
```

Figura 9. Esempio di pagina scritta in linguaggio HTML.

```
esempio.html x style.css
1 body {
2   font-family: Arial, sans-serif;
3   margin: 0;
4   padding: 0;
5   background-color: #f0f0f0;
6 }
7
8 header {
9   background-color: #4CAF50;
10  color: white;
11  padding: 1em;
12  text-align: center;
13 }
14
15 main {
16   padding: 2em;
17 }
18
19 img{
20   width:300px;
21 }
22
23 footer {
24   background-color: #333;
25   color: white;
26   text-align: center;
27   padding: 1em;
28   position: fixed;
29   width: 100%;
30   bottom: 0;
31 }
```

Figura 10. Esempio di linguaggio CSS da utilizzare come foglio di stile per la pagina di esempio a fig.8.

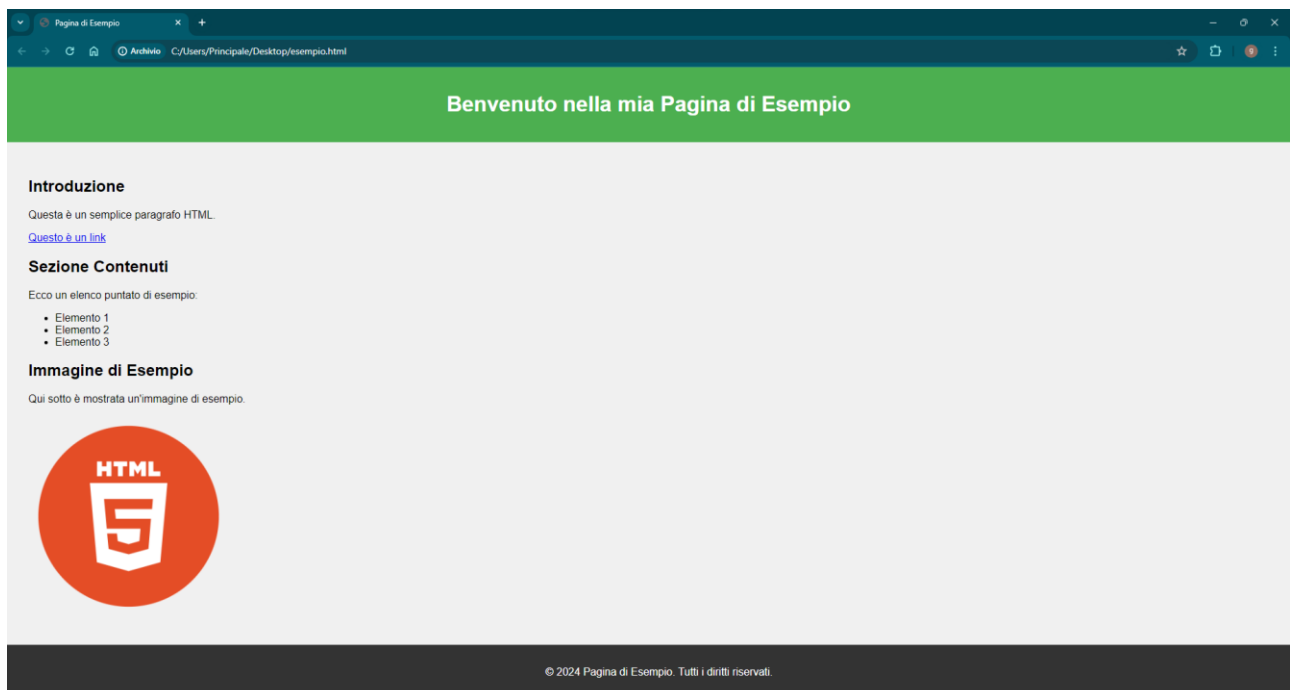


Figura 11. Presentazione web del linguaggio HTML e CSS mostrato nelle figure 8 e 9.

3.2 Linguaggi di back-end: PHP

Un linguaggio di back-end è un linguaggio di programmazione utilizzato per sviluppare la parte logica e funzionale di un sistema software o un'applicazione web che risiede sul lato server, ovvero che è eseguita su un server remoto e non sul dispositivo dell'utente. Questi linguaggi sono responsabili di gestire le operazioni di accesso ai dati, autenticazione degli utenti, la logica di presentazione dei dati e altre funzionalità fondamentali per l'utilizzo dell'applicazione. Una delle caratteristiche principali dei linguaggi di back-end è la gestione dei dati nel database.

Uno dei più popolari linguaggi lato server è PHP: linguaggio di scripting open source noto per la sua semplicità e flessibilità. Consente agli sviluppatori di creare pagine web interattive e gestire dati dinamici in modo efficiente. Spesso è utilizzato insieme a database di tipo relazionale come MySQL.

3.3 SQL e MySQL

Per la creazione del database è stato utilizzato il linguaggio SQL (Structured Query Language): un linguaggio standard per gestire e manipolare i dati nei sistemi di gestione dei database relazionali (RDBMS).

I RDBMS sono sistemi progettati per gestire database relazionali, organizzando i dati in tabelle composte da righe e colonne. Sono utilizzati anche in applicazioni importanti come la gestione dei sistemi bancari, delle risorse aziendali, l'e-commerce e molto altro. Ogni tabella rappresenta un'entità e le relazioni tra tabelle sono gestite tramite chiavi primarie e chiavi esterne.

Il linguaggio SQL è utilizzato in questo ambito per eseguire diverse operazioni come la creazione, l'aggiornamento, la cancellazione e l'interrogazione dei dati di un database.

Per il lavoro svolto in questo studio è stato utilizzato un database MySQL: sistema di gestione di RDBMS open source, sviluppato e supportato da Oracle Corporation. È popolare per le sue elevate prestazioni, affidabilità e facilità d'utilizzo. Può essere eseguito su diversi sistemi operativi come Microsoft Windows, GNU/Linux e iOS; supporta la scalabilità, la replica dei dati e robusti meccanismi di sicurezza tramite permessi e controllo degli accessi.

3.4 XAMPP

XAMPP è un pacchetto software gratuito e open source che fornisce un ambiente di sviluppo web locale completo, includendo Apache per il server web, MySQL per la gestione dei database e interpreti PHP e Pearl. Il nome è infatti un acronimo dei software citati dove X sta per 'Cross-platform'. Questo strumento è stato di grandissimo aiuto per creare un ambiente di test integrato e funzionale, facilitando lo sviluppo, la gestione e il debug delle connessioni stabilite tra l'interfaccia web e il database.

La configurazione di XAMPP è molto semplice e diretta: per prima cosa bisogna effettuare il download gratuito della piattaforma dal sito ufficiale di Apache Friends (facendo attenzione a scaricare la versione adatta al sistema operativo che si intende utilizzare). Dopo aver eseguito il file, seguire le istruzioni dell'installatore scegliendo i vari componenti predefiniti (Apache, MySQL, PHP, ecc.) e se richiesto specificare la cartella di destinazione del download (di solito la directory predefinita va bene). Una volta completata l'installazione, si può avviare il Pannello di Controllo di XAMPP (si trova nella directory dove è stata installata la piattaforma ma è consigliato aggiungere un'estensione nella barra delle applicazioni o nel Desktop). A questo punto manca solo di fare clic sui pulsanti 'Start' accanto ad Apache e MySQL per avviare i servizi. Lo stato cambierà colore diventando verde per indicare che i servizi sono attivi (Fig. 12).

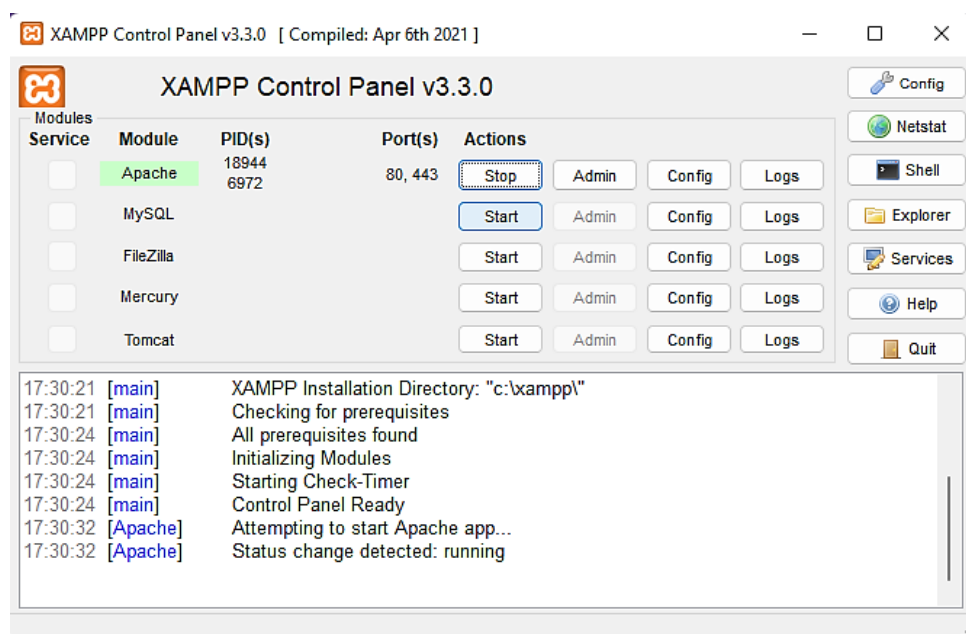


Figura 12. Pannello di Controllo della piattaforma XAMPP dove sono evidenziati i tasti 'Start' per attivare i servizi Apache e MySQL. La scritta Apache in verde indica lo stato attivo del servizio.

Per verificare che tutto sia andato a buon fine, basta aprire il browser web e digitare 'http://localhost' nella barra degli indirizzi: si dovrebbe entrare così nella pagina di benvenuto della piattaforma indicando che il server Apache è correttamente in esecuzione. Digitando sempre nella stessa barra 'http://localhost/phpmyadmin' si entra invece nell'applicazione web che consente di amministrare un database MySQL.

XAMPP è particolarmente utile per testare applicazioni web dinamiche senza la necessità di configurare e mantenere un ambiente di sviluppo separato, fornendo infatti un ambiente locale che consente anche di visualizzare in tempo reale gli aggiornamenti apportati al codice digitando 'localhost' nella barra degli indirizzi.

4. RISULTATI

4.1 Database

Nella fase iniziale, utilizzando la piattaforma XAMPP e, in particolare, di phpMyAdmin, è stato creato un database relazionale per raccogliere informazioni significative su utenti, pazienti e immagini colposcopiche. Il database è stato strutturato in quattro tabelle distinte con diversi campi ognuna.

- Tabella 'utenti'
 - id (int): chiave primaria, identificatore univoco dell'utente, con auto incremento.
 - nome (varchar 25): nome dell'utente.
 - cognome (varchar 25): cognome dell'utente.
 - sede (smallint): identificatore della sede lavorativa dell'utente.
 - pwd (varchar 255): password dell'utente.
 - e-mail (varchar 255): indirizzo di posta elettronica dell'utente.
 - autorizzazione_attiva (tinyint 1): flag che indica se l'autorizzazione dell'utente per inserire le immagini è attiva (1) o non attiva (0), modificabile solamente dall'amministratore del database.
 - data_revoca_autorizzazione (date): data in cui è stata revocata l'autorizzazione a inserire le immagini.
 - data_iscrizione (datetime): data in cui l'utente si è iscritto.
- Tabella 'anagrafica'
 - id (int): chiave primaria, identificatore univoco della paziente, con auto incremento.
 - id_utenti (int): identificatore dell'utente che ha inserito la paziente. Chiave esterna che fa riferimento alla colonna id della tabella utenti.
 - id_paziente (varchar 10): identificatore della paziente. Inserito dall'utente.
 - data_di_nascita (date).
 - etnia (varchar25).
 - bmi (decimal(3,1)): indice di massa corporea con una cifra decimale.
 - eta_menopausa (smallint).
 - tamoxifene(year): si riferisce all'ultimo anno in cui si è intrapreso un trattamento a base di tamoxifene. (0000 nel caso non presente).
 - mesi_ultimo_trattamento_tamoxifene (smallint): si riferisce ai mesi di durata dell'ultimo trattamento a base di tamoxifene.

- anno_inizio_fumo (year): anno in cui la paziente ha iniziato a fumare, se fumatrice. (0000 nel caso non fumatrice).
 - anni_fumato (smallint): numero di anni che la paziente ha fumato.
 - eta_prima_menstruazione (smallint).
 - numero_parti (smallint): numero di gestazioni avute dalla paziente.
 - terapia_ormonale (varchar 15): indica se la paziente sta facendo o ha già intrapreso una terapia ormonale.
 - data_inserimento (datetime).
- Tabella ‘patologie’
 - id (int): chiave primaria, identificatore univoco del record, con auto incremento.
 - id_inseritore (int): identificatore dell’utente che ha inserito la paziente. Chiave esterna che fa riferimento alla colonna id della tabella utenti.
 - id_paziente (varchar 10): identificatore della paziente. Inserito dall’utente.
 - diabete (varchar 2): possibili valori sì o ‘no’.
 - lynch (varchar 2): possibili valori sì o ‘no’ per sindrome di Lynch.
 - tumore_seno (varchar 2): possibili valori sì o ‘no’.
 - ipertensione (varchar 2): possibili valori sì o ‘no’.
 - data_inserimento (datetime).
- Tabella ‘immagini’
 - id (int): chiave primaria, identificatore univoco dell’immagine, con auto incremento.
 - id_inseritore (int): identificatore dell’utente che ha inserito l’immagine. Chiave esterna che fa riferimento alla colonna id della tabella utenti.
 - percorso_file (varchar 255): indica il percorso dell’immagine.
 - colposcopista (varchar 50): nome del colposcopista che ha eseguito l’esame.
 - sata_colposcopia (date).
 - strumentazione (varchar 255): nome della strumentazione utilizzata per fare l’esame.
 - fase_ciclo (varchar 20): fase del ciclo mestruale della paziente al momento della colposcopia.
 - colorazione (varchar 20): soluzioni utilizzate durante l’esame colposcopico come Lugol, acido acetico o nessuna.
 - endometrial_sampling_methods (smallint): indicatore che fa riferimento al metodo di campionamento utilizzato, se presente.

- sanguinamento_anormale (tinyint): Indicazione di sanguinamento uterino anormale al momento dell'esame. (1 = sì, 0 = no).
- note_diagnostiche (text): note aggiuntive riguardo l'esame).
- data_inserimento (datetime).

Queste tabelle sono interconnesse tramite delle chiavi esterne, per facilitare l'esecuzione di operazioni join tra esse. Sono state create utilizzando linguaggio SQL seguendo le query riportate di seguito.

- Tabella 'utenti'

```
CREATE TABLE utenti (
id INT (11) PRIMARY KEY NOT NULL AUTO_INCREMENT,
nome VARCHAR (25) NOT NULL,
cognome VARCHAR (25) NOT NULL,
sede SMALLINT (6) NOT NULL,
pwd VARCHAR (255) NOT NULL,
email VARCHAR (255) NOT NULL,
autorizzazione_attiva TINYINT (1) DEFAULT 0,
data_revoca_autorizzazione DATE DEFAULT NULL,
data_iscrizione DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP);
```

- Tabella 'anagrafica'

```
CREATE TABLE anagrafica (
id INT (11) PRIMARY KEY NOT NULL AUTO_INCREMENT,
id_utenti INT (11) NOT NULL ,
id_paziente VARCHAR (10) NOT NULL,
data_di_nascita DATE NOT NULL,
etnia VARCHAR (25) NOT NULL,
bmi DECIMAL (3,1) NOT NULL,
eta_menopausa SMALLINT (6) NOT NULL,
tamoxifene YEAR NOT NULL,
mesi_ultimo_trattamento_tamoxifene SMALLINT (6) NOT NULL,
anno_inizio_fumo YEAR NOT NULL,
anni_fumato SMALLINT (6) NOT NULL,
eta_prima_mestruazione SMALLINT (6) NOT NULL,
numero_parti SMALLINT (6) NOT NULL,
terapia_ormonale VARCHAR (15) NOT NULL,
data_inserimento DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
FOREIGN KEY (id_utenti) REFERENCES utenti(id));
```

- Tabella ‘patologie’

```
CREATE TABLE patologie (  
id INT (11) PRIMARY KEY NOT NULL AUTO_INCREMENT,  
id_inseritore INT (11) NOT NULL,  
id_paziente VARCHAR (10) NOT NULL,  
diabete VARCHAR (2) NOT NULL DEFAULT ‘no’,  
lynch VARCHAR (2) NOT NULL DEFAULT ‘no’,  
tumore_seno VARCHAR (2) NOT NULL DEFAULT ‘no’,  
ipertensione VARCHAR (2) NOT NULL DEFAULT ‘no’,  
data_inserimento DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP),  
FOREIGN KEY (id_inseritore) REFERENCES utenti(id));
```

- Tabella ‘immagini’

```
CREATE TABLE immagini (  
id INT (11) PRIMARY KEY NOT NULL AUTO_INCREMENT,  
id_inseritore INT (11) NOT NULL,  
id_tabella_anagrafica INT (11) NOT NULL,  
percorso_file VARCHAR (255) NOT NULL,  
colposcopista VARCHAR (50) NOT NULL,  
data_colposcopia DATE NOT NULL,  
strumentazione VARCHAR (50) NOT NULL,  
fase_ciclo VARCHAR (20) NOT NULL,  
colorazione VARCHAR (20) NOT NULL,  
endometrial_sampling_methods SMALLINT (6) NOT NULL,  
sanguinamento_anormale TINYINT (1) NOT NULL,  
note_diagnostiche TEXT,  
data_inserimento DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP),  
FOREIGN KEY (id_inseritore) REFERENCES utenti(id),  
FOREIGN KEY (id_tabella_anagrafica) REFERENCES anagrafica(id));
```

4.2 Codice

Tutto il codice del sito è stato salvato nella cartella ‘htdocs’ presente nella directory principale dell’installazione della piattaforma XAMPP. Per una migliore organizzazione, è stata creata una nuova cartella all’interno di ‘htdocs’, in cui i vari componenti del programma sono stati suddivisi e ordinati in modo appropriato.

Come è possibile vedere in figura 13 sottostante, all'interno di questa cartella sono presenti diversi elementi del sito web: il foglio di stile CSS, che determina l'estetica del sito, la pagina iniziale 'index.php', e altre cartelle.

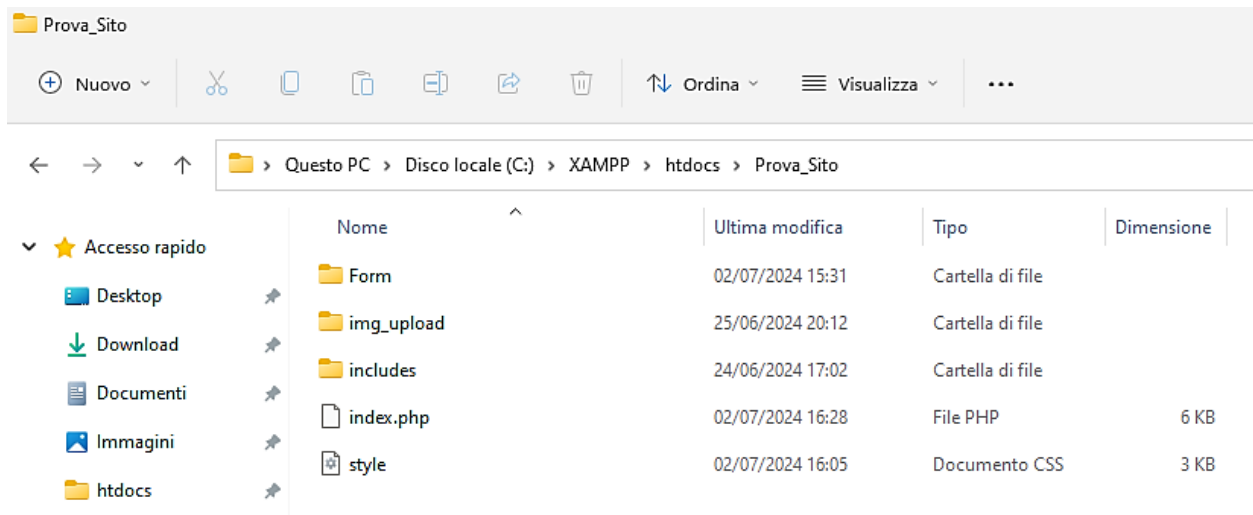


Figura 13. Cartella all'interno della quale sono stati organizzati e salvati i file contenenti il codice e le immagini di test.

Nello specifico, procedendo dall'alto verso il basso, si trovano:

- la cartella contenente tutte le form presenti nel sito.
- la cartella in cui sono state salvate le immagini.
- la cartella che includes tutto il codice di back-end, dalla connessione al database al controllo degli errori.

All'interno della cartella 'includes', è stata inserita nei nomi dei file l'estensione .inc: non è fondamentale ma serve ad indicare allo sviluppatore i file appartenenti alla cartella stessa. La maggior parte dei file qui presenti è suddivisa in quattro parti, ciascuna con funzioni specifiche:

1. nomefile.inc.php: esegue il codice quando viene inviata una richiesta tramite form.
2. nomefile_contr.inc.php: gestisce il controllo degli errori.
3. nomefile_model.inc.php: contiene le query per interrogare il database o inserire dati in esso.
4. nomefile_view.inc.php: include il codice che sarà effettivamente visualizzato nella pagina del sito.

4.2.1 Style.css

Il linguaggio CSS all'interno del file 'style.css' per rendere il sito più user friendly e intuitivo possibile è il seguente:

```
body {
  font-family: Arial, sans-serif;
  background-color: #f4f4f4;
  margin: 0;
  padding: 20px;
  display: flex;
  justify-content: center;
  align-items: center;
  min-height: 100vh; }

.container {
  background-color: #fff;
  padding: 30px;
  border-radius: 8px;
  box-shadow: 0 0 10px rgba(0, 0, 0,
0.1);
  width: 65%;

  max-width: 650px;
  text-align: center; }

.error {
  text-align: center;
  color: red;
  font-size: 20px;
  font-weight: bold;
  margin-bottom: 15px; }

h2 {
  text-align: center;
  margin-bottom: 30px;
  font-size: 26px; }

.form-group {
  margin-bottom: 20px; }

label {
  display: block;
  margin-bottom: 8px;
  font-weight: normal;
  font-size: 18px; }

input[type="text"],
input[type="date"],
input[type="number"],
input[type="email"],
input[type="password"],
select {
  width: calc(100% - 20px);
  padding: 10px;

  border: 1px solid #ccc;
  border-radius: 4px;
  box-sizing: border-box;
  font-size: 18px; }

.form-group.radio-group {
  display: flex;
  justify-content: space-between;
  align-items: center;
  margin-top: 10px; }

.form-group.radio-group label {
  margin-bottom: 0;
  font-size: 18px; }

.form-group.radio-group input {
  margin-right: 5px; }

button {
  width: 100%;
  padding: 12px;
  background-color: #007bff;
  border: none;
  border-radius: 4px;
  color: white;
  font-size: 18px;
  cursor: pointer;
  margin-top: 20px;
  transition: background-color 0.3s
ease, transform 0.3s ease;
  text-decoration: none;
  outline: none; }

button:hover {
  background-color: #0056b3; }

.success-message {
  color: #28a745;
  font-weight: bold;
  text-align: center;
  margin-top: 10px;
  font-size: 18px; }

.indietro {
  position: fixed;
  bottom: 20px;
  left: 20px;
  z-index: 100; }

.indietro a {
  display: inline-block;
```

```

padding: 10px 20px;
background-color: #007bff;
color: white;
text-decoration: none;
border-radius: 5px;
transition: background-color 0.3s
ease, transform 0.3s ease;
box-shadow: 0 4px 6px rgba(0, 0, 0,
0.1);
font-size: 18px; }

.indietro a:hover {
background-color: #0056b3; }

.btn-container {
display: flex;
justify-content: center;
gap: 10px; }

```

```

.btn {
background-color: #007bff;
color: #ffffff;
border: none;
padding: 12px 24px;
font-size: 16px;
cursor: pointer;
border-radius: 5px;
transition: background-color 0.3s
ease, transform 0.3s ease;
text-decoration: none;
outline: none; }

.btn:hover {
background-color: #0056b3; }

h1 {
color: #333333;
margin-bottom: 20px; }

```

Per motivazioni pratiche e organizzative, il codice CSS sopra indicato è stato utilizzato per stilizzare la maggior parte delle pagine del sito web. Fanno eccezione solo alcune pagine, come quella iniziale, per le quali si è deciso di incorporare il codice CSS all'interno dell'HTML.

4.2.2 Connessione al database e configurazione sessione

Una delle parti di codice fondamentali riguarda la connessione al database descritta nel capitolo 4.1. Questa connessione è stata implementata utilizzando il linguaggio PHP. Sarà richiamata in numerose pagine del codice ed è stata salvata con nome 'dbh.inc.php' ('dbh' è un'abbreviazione di 'databasehandler').

Seguono le righe di codice che permettono tale connessione.

```

<?php
$host="localhost";
$dbname="db_prova";
$dbusername = "root";
$dbpassword = "";
try {
    $pdo = new PDO("mysql:host=$host; dbname=$dbname", $dbusername, $dbpassword);
    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
} catch (PDOException $e) {
    echo "Connection Failed: " . $e->getMessage();
}

```

In molti dei file php si è scelto di utilizzare una funzione ‘try...catch’ al fine di visualizzare nell’interfaccia eventuali errori, che possono essere presenti nel codice o nella connessione al database, tramite ‘\$e->getMessage()’.

Una delle altre pagine più richiamate sarà quella contenente la configurazione della sessione: ‘config_session.inc.php’.

La configurazione della sessione è un processo essenziale che consente di gestire e mantenere lo stato dell’utente tra diverse pagine durante una sessione di navigazione. Durante la configurazione, è possibile definire delle variabili di sessione che mantengono informazioni specifiche per l’utente, come ad esempio le credenziali di accesso o altri dati dell’utente per personalizzare la sua navigazione all’interno del sito.

L’uso delle sessioni è cruciale per mantenere lo stato dell’utente senza la necessità di inviare le informazioni tramite URL o cookie ogni volta che si accede a una nuova pagina. Questo approccio è molto importante poiché migliora la sicurezza mantenendo i dati sensibili lato server e facilita la gestione dell’applicazione web.

Per questo progetto si è deciso di creare la sessione e rigenerarla ogni trenta minuti nel caso in cui l’utente rimanga connesso per più tempo.

Il codice presente nel file ‘config_session.inc.php’ è riportato a seguire.

```
<?php
ini_set('session.use_only_cookies', 1);
ini_set('session.use_strict_mode', 1); // assicura che l'id sia valido
session_set_cookie_params([ //possiamo settare diversi parametri con un array
    'lifetime' => 1800, //secondi di vita del cookie
    'domain' => 'localhost', //dominio del sito a cui punta
    'path' => '/', //tutti i path del sito
    'secure' => true,
    'httponly' => true //solo URL sicuri
]);
session_start();
function regenerate_session_id () {
    session_regenerate_id(true);
    $_SESSION['last_regeneration'] = time();
}
function regenerate_session_id_loggedin () {
    session_regenerate_id(true);
    $userId = $_SESSION["user_id"];
    $newSessionId = session_create_id();
    $sessionId = $newSessionId . "_" . $userId;
    session_id($sessionId);
}
```

```

        $_SESSION['last_regeneration'] = time();
    }
    if(isset($_SESSION["user_id"])) {
        if (!isset($_SESSION['last_regeneration'])){
            regenerate_session_id_loggedin();
        }
        else {
            $interval = 30*60; //30 minuti
            if (time() - $_SESSION['last_regeneration'] >= $interval){
                regenerate_session_id_loggedin();
            }
        }
    } else{
        if (!isset($_SESSION['last_regeneration'])){
            regenerate_session_id();
        }
        else {
            $interval = 30*60;
            if (time() - $_SESSION['last_regeneration'] >= $interval){
                regenerate_session_id();
            }
        }
    }
}

```

4.2.3 Homepage

La prima pagina visualizzata all’accesso del sito web è l’homepage, salvata come ‘index.php’ poiché il server Apache, così come la maggior parte di server web, è configurato per cercare automaticamente file con nomi predefiniti come questo.

L’homepage è stata progettata per avere una sezione <header> che contiene il nome del sito e include dei pulsanti in alto a destra. Sotto l’header si trova un menù che elenca le diverse attività disponibili per l’utente una volta effettuato l’accesso. Nel tag HTML <body> è presente un messaggio di benvenuto e una breve descrizione del progetto.

Nelle prime righe di codice sono inclusi anche dei riferimenti ad altre pagine, come la configurazione della sessione descritta nel sottocapitolo precedente e ‘login_view.inc.php’, che sarà trattata prossimamente.

Alcune parti del codice sono state rese dinamiche tramite dichiarazioni ‘if...’ come i pulsanti in alto a destra che mostrano le opzioni ‘iscriviti’ o ‘accedi’ se nessun utente ha effettuato il login, ovvero nel caso in cui la variabile ‘\$_SESSION[“user_id”]’ non è impostata. In caso contrario, viene visualizzato un pulsante di logout e verrà mostrato il menù con le diverse azioni.

Per ragioni organizzative, il codice CSS relativo all'homepage è stato incluso all'interno dell'HTML nei tag <style> come è possibile vedere nelle righe che seguono.

```
<?php
require_once 'includes/config_session.inc.php';
require_once 'includes/login_view.inc.php';
?>
<!DOCTYPE html>
<html lang="it">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Homepage</title>
  <style>
body {
  font-family: 'Helvetica Neue',
Arial, sans-serif;
  margin: 0;
  padding: 0;
  background-color: #eef2f3;
}
.header {
  background-color: #0056b3;
  color: white;
  padding: 20px 20px;
  text-align: center;
  position: relative;
}
.header h1 {
  margin: 0;
  font-size: 2em;
}
.auth-buttons {
  position: absolute;
  top: 20px;
  right: 20px;
}
.auth-buttons a {
  display: inline-block;
  margin-left: 10px;
  padding: 10px 20px;
  font-size: 18px;
  color: white;
  background-color: #007bff;
  text-decoration: none;
  border-radius: 5px;
  transition: background-color 0.3s
ease;
}
.auth-buttons a:hover {
  background-color: #0056b3;
}
.menu {
  list-style: none;
  padding: 0;
  margin: 0;
  display: flex;
  justify-content: center;
  background-color: #007bff;
  font-size: 20px;
}
.menu li {
  margin: 0;
}
.menu a {
  display: block;
  padding: 15px 20px;
  text-align: center;
  text-decoration: none;
  color: white;
  transition: background-color 0.3s
ease;
}
.menu a:hover {
  background-color: #0056b3;
}
.container {
  text-align: center;
  max-width: 900px;
  margin: 30px auto;
  padding: 20px;
  background-color: white;
  border-radius: 8px;

```

```

        box-shadow: 0 4px 8px rgba(0, 0, 0,
0.1);
    }
    .success-message {
        color: green;
        font-weight: bold;
        margin: 20px 0;
        text-align: center;
        padding: 10px;
        border: 1px solid green;
        border-radius: 5px;
        background-color: #e8f5e9;
    }
    .intro {

        margin: 20px 0;
        padding: 20px;
        background-color: #f8f9fa;
        border-left: 5px solid #007bff;
        border-radius: 5px;
        font-size: 20px;
    }
    .intro h2 {
        margin-top: 0;
        color: #0056b3;
    }
    .intro p {
        margin-bottom: 0;
    }

</style>
</head>
<body>
    <div class="header">
        <h1>Homepage</h1>
        <?php if(!isset($_SESSION["user_id"])){ ?>
        <div class="auth-buttons">
            <a href="../../../Prova_Sito/Form/signupForm.php">Signup</a>
            <a href="../../../Prova_Sito/Form/loginForm.php">Login</a>
        </div>
        <?php }
        if(isset($_SESSION["user_id"])){ ?>
        <div class="auth-buttons">
            <a href="../../../Prova_Sito/includes/logout.inc.php">Logout</a>
        </div>
        <?php } ?>
    </div>
    <?php if(isset($_SESSION["user_id"])){ ?>
        <ul class="menu">
            <li><a href="../../../Prova_Sito/Form/anagraficaForm.php">Inserisci una nuova
paziente</a></li>
            <li><a href="../../../Prova_Sito/Form/upload.php">Carica immagine</a></li>
            <li><a href="../../../Prova_Sito/Form/galleria.php">Galleria</a></li>
            <li><a href="../../../Prova_Sito/Form/gestioneAnagrafica.php">Rivedi o modifica
schede di pazienti già inserite</a></li>
            <li><a href="../../../Prova_Sito/Form/gestioneUtenti.php">Account</a></li>
        </ul>
        <?php } ?>
    <div class="container">
        <?php output_username(); ?>
        <?php
        if(isset($_GET['signup']) && $_GET['signup'] === "success"){
            echo "<p class='success-message'>CONGRATULAZIONI, TI SEI ISCRITTO CON
SUCCESSO!</p>";
        }
        if(isset($_GET['login']) && $_GET['login'] === "success"){

```

```

        echo "<p class='success-message'>CONGRATULAZIONI, HAI ESEGUITO L'ACCESSO
CON SUCCESSO!</p>";
    }

if(isset($_GET["upload"]) && $_GET["upload"] === "success"){
    echo "<p class='success-message'>IL FILE &Egrave STATO INSERITO CON
SUCCESSO!</p>";
}
if(isset($_GET["anagrafica"]) && $_GET["anagrafica"] === "success"){
    echo "<p class='success-message'>LA NUOVA PAZIENTE &Egrave STATA INSERITA
CON SUCCESSO!</p>";
}
?>
<div class="intro">
    <h2>Benvenuti nella nostra piattaforma</h2>
    <p> DESCRIZIONE PROGETTO </p>
</div>
</div>
</body>
</html>

```

4.2.4 Signup, Login e Logout

L'iscrizione e l'accesso sono gestiti da due moduli distinti.

Per l'iscrizione, il modulo richiede al nuovo utente di fornire nome, cognome, sede lavorativa, email e password. La struttura del codice è organizzata in cinque file. Il file 'signupForm.php' (codice sottostante) contiene il modulo di iscrizione vero e proprio; nelle prime righe vengono richiamate la configurazione della sessione e il file 'signup_view.inc.php', che contiene tutto ciò che verrà visualizzato nel sito web. È stato aggiunto, come in quasi tutte le pagine, un tasto in basso a sinistra che riporta alla pagina precedente. Per quanto riguarda lo stile, è stato incluso il foglio di stile visto a 4.2.1.

```

<?php
require_once '../includes/config_session.inc.php';
require_once '../includes/signup_view.inc.php';
?>
<!DOCTYPE html>
<html lang="it">
<head>
    <meta name = "author" content = "Lapponi Gabriele" />
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Form di Iscrizione</title>
<link rel="stylesheet" href="../style.css">

```

```

    <link rel="stylesheet" href="../style.css">
</head>
<body>
    <div class="container">
        <h2>Iscriviti</h2>
        <form action="../includes/signup.inc.php" method="post">
            <?php
                signup_inputs();
            ?>
            <button type="submit">Iscriviti</button>
        </form>
        <br>
        <?php
            check_signup_errors();
        ?>
    </div>
    <div class="indietro">
        <a href="../index.php">Indietro</a>
    </div>
</body>
</html>

```

All'interno di 'signup_view.inc.php' (codice che segue), sono contenute le funzioni `signup_inputs()` e `check_signup_errors()`. La prima si occupa di mostrare etichette e campi di input (riempiti con le informazioni scritte in precedenza nel caso risulti qualche errore) mentre la seconda riguarda il display, nell'interfaccia, dei vari errori.

```

function signup_inputs(){
    if (isset($_SESSION["signup_data"]["nome"])){
        echo '<div class="form-group">
            <label for="nome">Nome:</label>
            <input type="text" id="nome" name="nome" value="' .
$_SESSION["signup_data"]["nome"] .'" />
            </div>' ;
    } else{
        echo '<div class="form-group">
            <label for="nome">Nome:</label>
            <input type="text" id="nome" name="nome">
            </div>';
    }
    if (isset($_SESSION["signup_data"]["cognome"])){
        echo '<div class="form-group">
            <label for="cognome">Cognome:</label>
            <input type="text" id="cognome" name="cognome" value="' .
$_SESSION["signup_data"]["cognome"] .'" />
            </div>' ;
    } else{
        echo '<div class="form-group">
            <label for="cognome">Cognome:</label>
            <input type="text" id="cognome" name="cognome">
            </div>';
    }
}

```



```

echo '<div class="form-group">
    <label for="sede">Sede lavorativa:*/label>
    <select id="sede" name="sede">';
$sedi = ["1", "2", "3"];
foreach ($sedi as $sede) {
    $selected = (isset($_SESSION["signup_data"]["sede"]) &&
$_SESSION["signup_data"]["sede"] == $sede) ? 'selected' : '';
    echo "<option value=\"\$sede\" $selected>$sede</option>";
}
echo '</select>
</div>';
if(isset($_SESSION["signup_data"]["email"]) &&
!isset($_SESSION["errors_signup"]["email_used"]) &&
!isset($_SESSION["errors_signup"]["invalid_email"])){
    echo '<div class="form-group">
        <label for="email">Indirizzo Email:</label>
        <input type="email" id="email" name="email" value="" .
$_SESSION["signup_data"]["email"] . "" />
        </div>';
    } else{
        echo '<div class="form-group">
            <label for="email">Indirizzo Email:</label>
            <input type="email" id="email" name="email">
            </div>';
        }
echo '<div class="form-group">
    <label for="pwd">Password:</label>
    <input type="password" id="pwd" name="pwd">
    </div>';
echo '<br>';
}

function check_signup_errors(){
    if (isset($_SESSION['errors_signup'])){
        $errors = $_SESSION['errors_signup'];
        echo "<br>";
        foreach ($errors as $error) {
            echo "<p class='error'>" . $error . "</p>";
        }
        unset($_SESSION['errors_signup']);
    }
}
}

```

Una volta inviata la form con il metodo POST, viene eseguito il codice di ‘signup.inc.php’, che prende le variabili di input e le rinomina. Se sono presenti errori, il codice crea due array: il primo contenente le variabili inviate e il secondo gli errori commessi. In questo caso l’esecuzione si interrompe. Se invece non vengono rilevati errori, viene richiamata la funzione ‘create_user’ come è possibile vedere a pagina successiva.

```

<?php
if ($_SERVER["REQUEST_METHOD"] == "POST"){
    $nome = $_POST["nome"];
    $cognome = $_POST["cognome"];
    $pwd = $_POST["pwd"];
    $email = $_POST["email"];
    $sede = $_POST["sede"];
    try {
        require_once "dbh.inc.php";
        require_once "signup_model.inc.php";
        require_once "signup_contr.inc.php";
        // ERROR HANDLERS
        $errors = [];
        if (is_input_empty($nome, $cognome, $pwd, $email)){
            $errors["empty_input"] = "Riempi tutti i campi richiesti!";
        }
        if (is_email_invalid($email)){
            $errors["invalid_email"] = "L'e-mail scritta non è valida!";
        }
        if (is_email_registered($pdo, $email)){
            $errors["email_used"] = "L'e-mail scritta è già utilizzata!";
        }
        require_once "config_session.inc.php";
        if($errors){
            $_SESSION["errors_signup"] = $errors;

            $signupData = [
                "nome" => $nome,
                "cognome" => $cognome,
                "email" => $email,
                "sede" => $sede
            ];
            $_SESSION["signup_data"] = $signupData;
            header("Location: ../Form/signupForm.php");
            die();
        }
        create_user($pdo, $nome, $cognome, $sede, $pwd, $email);
        header("Location: ../index.php?signup=success");
        $pdo = null;
        $stmt = null;
        die();
    }
    catch (PDOException $e) {
        die("Query failed: " . $e->getMessage());
    }
}
else {
    header("Location: ../Form/signupForm.php");
    die();
}
}

```

Le funzioni presenti nel codice sono richiamate da 'signup_contr.inc.php':

```
<?php
declare (strict_types=1);
function is_input_empty(string $nome, string $cognome, string $pwd, string $email){
    if(empty($nome) or empty($cognome) or empty($pwd) or empty($email)){
        return true;
    }
    else {
        return false;
    }
}
function is_email_invalid(string $email){
    if(!filter_var($email, FILTER_VALIDATE_EMAIL)){
        return true;
    }
    else {
        return false;
    }
}
function is_email_registered(object $pdo, string $email){
    if(get_email($pdo, $email)){
        return true;
    }
    else {
        return false;
    }
}
function create_user(object $pdo, string $nome, string $cognome, string $sede, string
$pwd, string $email){
    set_user($pdo, $nome, $cognome, $sede, $pwd, $email);
}
}
```

Una volta superati i controlli per gli errori, viene eseguita la funzione set_user(...), nella quale è presente una query che si occupa di inserire i dati richiesti nel database.

```
<?php
declare (strict_types=1);
function get_email (object $pdo, string $email){
    $query = "SELECT nome FROM utenti WHERE email = :email;";
    $stmt = $pdo -> prepare($query);
    $stmt->bindParam(":email", $email);
    $stmt->execute();
    $result = $stmt -> fetch(PDO::FETCH_ASSOC);
    return $result;
}
function set_user(object $pdo, string $nome, string $cognome, string $sede, string
$pwd, string $email) {
    $query = "INSERT INTO utenti (nome, cognome, sede, pwd, email) VALUES (:nome,
:cognome, :sede, :pwd, :email)";
    $stmt = $pdo -> prepare($query);
    $options = [
        'cost' => 12
    ];
    $hashedPwd = password_hash($pwd, PASSWORD_BCRYPT, $options);
```

```

$stmt->bindParam(':nome', $nome);
$stmt->bindParam(':cognome', $cognome);
$stmt->bindParam(':sede', $sede);
$stmt->bindParam(':pwd', $hashedPwd);
$stmt->bindParam(':email', $email);
$stmt->execute();
}

```

In ogni query viene utilizzata la funzione `bindParam(':variabile', $variabile)` come controllo di sicurezza per evitare possibili attacchi di SQL injection da malintenzionati. Un ulteriore controllo di sicurezza consiste nell'utilizzo della funzione `password_hash($pwd, PASSWORD_BCRYPT, $options)` in maniera da rendere criptati dati sensibili come le password, prima di inserirli nel database.

Per quanto riguarda il Login l'approccio è simile: all'interno del file 'loginForm' è inserita la form nella quale vengono richiamate le funzioni presenti in 'login_view.inc.php', e un tasto indietro che riporta all'homepage.

```

<?php
require_once '../includes/config_session.inc.php';
require_once '../includes/login_view.inc.php';
?>
<!DOCTYPE html>
<html lang="it">
<head>
    <meta name = "author" content = "Lapponi Gabriele" />
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Form di Login</title>
    <link rel="stylesheet" href="../style.css">
</head>
<body>
    <div class="container">
        <h2>Login</h2>
        <form action="../includes/login.inc.php" method="post">
            <?php
            login_inputs();
            ?>
            <button type="submit">Login</button>
        </form>
        <br>
        <?php
        check_login_errors();
        ?>
    </div>
    <div class="indietro">
        <a href="../index.php">Indietro</a>
    </div>
</body>
</html>

```

La prima delle due funzioni mostra i campi di inserimento per accedere con e-mail e password mentre la seconda si occupa del display dei vari errori, se commessi.

```
<?php
function login_inputs(){
    if(isset($_SESSION["login_data"]["email"]) &&
!isset($_SESSION["errors_login"]["email_wrong"])){
        echo '<div class="form-group">
            <label for="email">Indirizzo Email:</label>
            <input type="email" id="email" name="email" value="' .
$_SESSION["login_data"]["email"] .'"/>
        </div>';
    } else{
        echo '<div class="form-group">
            <label for="email">Indirizzo Email:</label>
            <input type="email" id="email" name="email">
        </div>';
    }
    echo '<div class="form-group">
        <label for="pwd">Password:</label>
        <input type="password" id="pwd" name="pwd">
    </div>';
    echo '<br>';
}
function check_login_errors(){
    if(isset($_SESSION['errors_login'])){
        $errors = ($_SESSION['errors_login']);
        echo "<br>";
        foreach ($errors as $error) {
            echo "<p class='error'>" . $error . "</p>";
        }
        unset($_SESSION['errors_login']);
    }
}
}
```

Anche qui, una volta premuto il tasto per il login, viene eseguito il codice `action="../../../includes/login.inc.php"`.

```
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST"){
    $email = $_POST["email"];
    $pwd = $_POST["pwd"];
    try {
        require_once "dbh.inc.php";
        require_once "login_model.inc.php";
        require_once "login_contr.inc.php";
        // ERROR HANDLERS
        $errors = [];
        if (is_input_empty($email, $pwd)){
            $errors["empty_input"] = "Riempì tutti i campi richiesti!";
        }
        $result = get_user($pdo, $email);
        if (is_email_wrong($result)){
            $errors["email_wrong"] = "Dati inseriti sono errati!";
        }
    }
}
```

```

        if (!is_email_wrong($result) && is_password_wrong($pwd,
$result["pwd"])){
            $errors["password_wrong"] = "Dati inseriti sono errati!";
        }
        require_once "config_session.inc.php";
        if ($errors) {
            $_SESSION["errors_login"] = $errors;
        }
        $loginData = [
            "email" => $email
        ];
        $_SESSION["login_data"] = $loginData;
        header("Location: ../Form/loginForm.php");
        die();
    }
    $newSessionId = session_create_id();
    $sessionId = $newSessionId . "_" . $result["id"];
    session_id($sessionId);
    $_SESSION["user_id"] = $result["id"];
    $_SESSION["user_pwd"] = $result["pwd"];
    $_SESSION["user_email"] = htmlspecialchars($result["email"]);
    $_SESSION["user_name"] = $result["nome"];
    $_SESSION['last_regeneration'] = time();
    header("Location:../index.php?login=success");
    $pdo = null;
    $stmt = null;
    die();
} catch (PDOException $e) {
    die("Query failed: " . $e->getMessage());
}
}
else {
    header("Location:../Form/loginForm.php");
    die();
}
}

```

In questo caso, se non sono stati commessi errori, vengono create delle variabili globali `$_SESSION` per poi inviare l'utente connesso alla prima pagina del sito. Le funzioni che svolgono il compito di controllo sono quelle presenti in 'login_contr.inc.php':

```

<?php
declare(strict_types=1);
function is_input_empty(string $email, string $pwd){
    if(empty($email) or empty($pwd)){
        return true;
    }
    else {
        return false;
    }
}
function is_email_wrong(bool|array $result){
    if(!$result){
        return true;
    } else{
        return false;
    }
}
}

```

```
function is_password_wrong(string $pwd, string $hashedPwd){
    if(!password_verify($pwd, $hashedPwd)){
        return true;
    } else{
        return false;
    }
}
```

Come prima cosa si è scelto di controllare se uno dei due campi richiesti è vuoto, se l'email inserita non è tra le presenti nel database e se la password non corrisponde. La query utilizzata per interrogare il database è scritta nel 'login_model.inc.php', mostrato di seguito.

```
<?php
declare(strict_types=1);
function get_user(object $pdo, string $email){
    $query = "SELECT * FROM utenti WHERE email = :email;";
    $stmt = $pdo -> prepare($query);
    $stmt->bindParam(":email", $email);
    $stmt->execute();
    $result = $stmt -> fetch(PDO::FETCH_ASSOC);
    return $result;
}
```

Una volta connesso, l'utente può decidere di disconnettersi semplicemente cliccando sul pulsante di logout, ora visibile in alto a destra nella pagina principale. Questo tasto esegue un codice breve contenuto nel file 'logout.inc.php', il quale distrugge la sessione attiva creata.

```
<?php

session_start();
session_unset();
session_destroy();

header("Location:../index.php");
die();
```

4.2.5 Inserimento pazienti

La prima attività disponibile nel menu dell'homepage, se l'utente ha effettuato correttamente l'accesso, è l'inserimento di una nuova scheda anagrafica relativa ad una paziente.

Come discusso in precedenza, per una migliore organizzazione, il codice relativo all'inserimento è diviso in vari file. Il primo di questi è 'anagraficaForm.php', situato nella cartella 'Form', dove sono richiamate le funzioni da 'anagrafica_view.inc.php' nonché la pagina di configurazione della sessione. Il codice relativo è il seguente:

```
<?php
require_once '../includes/config_session.inc.php';
require_once '../includes/anagrafica_view.inc.php';
require_once '../includes/login_view.inc.php';
?>
<!DOCTYPE html>
<html lang="it">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Inserimento Nuovo Paziente</title>
    <!--LINK ALLO STILE CSS-->
    <link rel="stylesheet" href="../style.css">
</head>
<body>
    <div class="container">
        <h2 style="color:#0056b3;">Anagrafica Nuova Paziente</h2>
        <form action="../includes/anagrafica.inc.php" method="post">
            <?php
                anagrafica_inputs();
            ?>
            <button type="submit">Inserisci Paziente</button>
        </form>
        <br>
        <?php
            check_anagrafica_errors();
        ?>
    </div>
    <div class="indietro">
        <a href="../index.php">Indietro</a>
    </div>
</body>
</html>
```

Nella funzione `anagrafica_inputs()`, vengono richiesti tutti i campi considerati significativi, riguardanti una paziente, già discussi nel capitolo 4.1 per le tabelle 'anagrafica' e 'patologie' del database. Anche in questa parte di codice, analogamente a quanto fatto per il modulo di iscrizione, è stata prestata particolare attenzione al

mantenimento dei campi di input compilati con i valori inseriti dall'utente nel caso egli commetta degli errori nell'inserimento, in modo tale da evitare la necessità di effettuare nuovamente la compilazione completa del modulo.

La seconda funzione, `check_anagrafica_errors()`, si occupa come sempre del display degli errori.

```
<?php
declare (strict_types=1);
// Funzione per ottenere la data odierna in formato YYYY-MM-DD
function getTodayDate() {
    return date('Y-m-d');
}

function anagrafica_inputs() {
    //Check for paziente
    if (isset($_SESSION["anagrafica_data"]["id_paziente"]) &&
!isset($_SESSION["errors_anagrafica"]["paziente_esistente"])) {
        echo '<div class="form-group">
            <label for="id_paziente">ID del Paziente:*</label>
            <input type="text" id="id_paziente" name="id_paziente" value="" .
htmlspecialchars($_SESSION["anagrafica_data"]["id_paziente"], ENT_QUOTES) . "' />
            </div>';
    } else {
        echo '<div class="form-group">
            <label for="id_paziente">ID del Paziente:*</label>
            <input type="text" id="id_paziente" name="id_paziente">
            </div>';
    }
    $todayDate = getTodayDate();
    if (isset($_SESSION["anagrafica_data"]["data_nascita"])) {
        echo '<div class="form-group">
            <label for="data_nascita">Data di Nascita:*</label>
            <input type="date" id="data_nascita" name="data_nascita" value="" .
htmlspecialchars($_SESSION["anagrafica_data"]["data_nascita"], ENT_QUOTES) . "'
max="" . $todayDate . "' />
            </div>';
    } else {
        echo '<div class="form-group">
            <label for="data_nascita">Data di Nascita:*</label>
            <input type="date" id="data_nascita" name="data_nascita" max="" .
$todayDate . "' />
            </div>';
    }
    //Check for Etnia
    echo '<div class="form-group">
        <label for="etnia">Etnia:*</label>
        <select id="etnia" name="etnia">';
    $etnie = ["caucasica", "afroamericana", "asiatica", "latina", "altro"];
    foreach ($etnie as $etnia) {
        $selected = (isset($_SESSION["anagrafica_data"]["etnia"]) &&
$_SESSION["anagrafica_data"]["etnia"] == $etnia) ? 'selected' : '';
        echo "<option value=\"\$etnia\" $selected>$etnia</option>";
    }
    echo '</select>
        </div>';
}
```

```

// Check for BMI
if (isset($_SESSION["anagrafica_data"]["bmi"])) {
    echo '<div class="form-group">
        <label for="bmi">BMI:*</label>
        <input type="number" id="bmi" name="bmi" min="1" max="50" step="0.1"
value="" . htmlspecialchars($_SESSION["anagrafica_data"]["bmi"], ENT_QUOTES) . "' />
        </div>';
} else {
    echo '<div class="form-group">
        <label for="bmi">BMI:*</label>
        <input type="number" id="bmi" name="bmi" min="1" max="50" step="0.1">
        </div>';
}
// Check for Età della Menopausa
echo '<div class="form-group">
    <label for="eta_menopausa">Età della Menopausa (inserire 0 se non ancora
in menopausa):*</label>
    <select id="eta_menopausa" name="eta_menopausa">';
for ($i = 0; $i < 100; $i++) {
    $selected = (isset($_SESSION["anagrafica_data"]["eta_menopausa"]) &&
$_SESSION["anagrafica_data"]["eta_menopausa"] == $i) ? 'selected' : '';
    echo "<option value=\"\$i\" $selected>\$i</option>";
}
echo '</select>
    </div>';
// Check for Anno di Inizio ultima Terapia a Base di Tamoxifene
if (isset($_SESSION["anagrafica_data"]["anno_tamoxifene"])) {
    echo '<div class="form-group">
        <label for="anno_tamoxifene">Anno di Inizio ultima Terapia a Base di
Tamoxifene:</label>
        <input type="number" id="anno_tamoxifene" name="anno_tamoxifene"
min="1930" max="2099" step="1" value="" .
htmlspecialchars($_SESSION["anagrafica_data"]["anno_tamoxifene"], ENT_QUOTES) . "' />
        </div>';
} else {
    echo '<div class="form-group">
        <label for="anno_tamoxifene">Anno di Inizio ultima Terapia a Base di
Tamoxifene:</label>
        <input type="number" id="anno_tamoxifene" name="anno_tamoxifene"
min="1930" max="2099" step="1">
        </div>';
}
// Check for Mesi di Uso di Tamoxifene
if (isset($_SESSION["anagrafica_data"]["mesi_tamoxifene"])) {
    echo '<div class="form-group">
        <label for="mesi_tamoxifene">Mesi di Uso di Tamoxifene:</label>
        <input type="number" id="mesi_tamoxifene" name="mesi_tamoxifene"
min="0" value="" . htmlspecialchars($_SESSION["anagrafica_data"]["mesi_tamoxifene"],
ENT_QUOTES) . "' />
        </div>';
} else {
    echo '<div class="form-group">
        <label for="mesi_tamoxifene">Mesi di Uso di Tamoxifene:</label>
        <input type="number" id="mesi_tamoxifene" name="mesi_tamoxifene"
min="0">
        </div>';
}
}

```

```

// Check for Fumatore
echo '<div class="form-group">
    <label class="radio-label">Fumatrice:</label>
    <div class="radio-buttons">
        <input type="radio" id="fumatore_si" name="fumatore" value="si"
onclick="toggleFumo(true)" ' . (isset($_SESSION["anagrafica_data"]["fumatore"]) &&
$_SESSION["anagrafica_data"]["fumatore"] == "si" ? 'checked' : '') . '> Sì
        <input type="radio" id="fumatore_no" name="fumatore" value="no"
onclick="toggleFumo(false)" ' . (isset($_SESSION["anagrafica_data"]["fumatore"]) &&
$_SESSION["anagrafica_data"]["fumatore"] == "no" ? 'checked' : '') . '> No
    </div>
</div>';

// Check for Anno di Inizio del Fumo
echo '<div class="form-group" id="anno_fumo_group" style="display: ' .
(isset($_SESSION["anagrafica_data"]["fumatore"]) &&
$_SESSION["anagrafica_data"]["fumatore"] == "si" ? 'block' : 'none') . ';">
    <label for="anno_fumo">Anno di Inizio del Fumo:</label>
    <input type="number" id="anno_fumo" name="anno_fumo" min="1930"
max="2099" step="1" value="" . (isset($_SESSION["anagrafica_data"]["anno_fumo"]) ?
htmlspecialchars($_SESSION["anagrafica_data"]["anno_fumo"], ENT_QUOTES) : '') . "' />
</div>';

// Check for Anni di Fumo
echo '<div class="form-group" id="anni_fumo_group" style="display: ' .
(isset($_SESSION["anagrafica_data"]["fumatore"]) &&
$_SESSION["anagrafica_data"]["fumatore"] == "si" ? 'block' : 'none') . ';">
    <label for="anni_fumo">Anni di Fumo:</label>
    <input type="number" id="anni_fumo" name="anni_fumo" min="0" value="" .
(isset($_SESSION["anagrafica_data"]["anni_fumo"]) ?
htmlspecialchars($_SESSION["anagrafica_data"]["anni_fumo"], ENT_QUOTES) : '') . "' />
</div>';

// Check for Età della Prima Mestruazione
echo '<div class="form-group">
    <label for="eta_mestruazione">Età della Prima Mestruazione (inserire 0 se
non ancora avvenuta):*</label>
    <select id="eta_mestruazione" name="eta_mestruazione">';
    for ($i = 0; $i < 21; $i++) {
        $selected = (isset($_SESSION["anagrafica_data"]["eta_mestruazione"]) &&
$_SESSION["anagrafica_data"]["eta_mestruazione"] == $i) ? 'selected' : '';
        echo "<option value=\"$i\" $selected>$i</option>";
    }
    echo '</select>
</div>';

// Check for Numero di gestazioni
echo '<div class="form-group">
    <label for="numero_parti">Numero di Gestazioni (inserire 0 se non
avvenute):*</label>
    <select id="numero_parti" name="numero_parti">';
    for ($i = 0; $i < 15; $i++) {
        $selected = (isset($_SESSION["anagrafica_data"]["numero_parti"]) &&
$_SESSION["anagrafica_data"]["numero_parti"] == $i) ? 'selected' : '';
        echo "<option value=\"$i\" $selected>$i</option>";
    }
    echo '</select>
</div>';

// Check for Terapia ormonale
echo '<div class="form-group">
    <label for="terapia_ormonale">Terapia Ormonale:</label>
    <select id="terapia_ormonale" name="terapia_ormonale">';

```

```

$terapie = ["assente", "attuale", "pregressa"];
foreach ($terapie as $terapia) {
    $selected = (isset($_SESSION["anagrafica_data"][$terapia_ormonale]) &&
$_SESSION["anagrafica_data"][$terapia_ormonale] == $terapia) ? 'selected' : '';
    echo "<option value=\"\$terapia\" $selected>$terapia</option>";
}
echo '</select>
</div>';
// Check for Diabete
echo '<div class="form-group">
<label for="diabete">La paziente presenta diabete?*</label>
<select id="diabete" name="diabete">';
$diabete = ["no", "si"];
foreach ($diabete as $valore) {
    $selected = (isset($_SESSION["anagrafica_data"]["diabete"]) &&
$_SESSION["anagrafica_data"]["diabete"] == $valore) ? 'selected' : '';
    echo "<option value=\"\$valore\" $selected>$valore</option>";
}
echo '</select>
</div>';
// Check for Lynch
echo '<div class="form-group">
<label for="lynch">La paziente presenta sindrome di Lynch?*</label>
<select id="lynch" name="lynch">';
$lynchs = ["no", "si"];
foreach ($lynchs as $lynch) {
    $selected = (isset($_SESSION["anagrafica_data"]["lynch"]) &&
$_SESSION["anagrafica_data"]["lynch"] == $lynch) ? 'selected' : '';
    echo "<option value=\"\$lynch\" $selected>$lynch</option>";
}
echo '</select>
</div>';
// Check for Tumore Seno
echo '<div class="form-group">
<label for="tumore_seno">La paziente soffre di tumore al seno?*</label>
<select id="tumore_seno" name="tumore_seno">';
$tumore = ["no", "si"];
foreach ($tumore as $tumore_seno) {
    $selected = (isset($_SESSION["anagrafica_data"]["tumore_seno"]) &&
$_SESSION["anagrafica_data"]["tumore_seno"] == $tumore_seno) ? 'selected' : '';
    echo "<option value=\"\$tumore_seno\" $selected>$tumore_seno</option>";
}
echo '</select>
</div>';
// Check for Ipertensione
echo '<div class="form-group">
<label for="ipertensione">La paziente soffre di ipertensione?*</label>
<select id="ipertensione" name="ipertensione">';
$pressione = ["no", "si"];
foreach ($pressione as $ipertensione) {
    $selected = (isset($_SESSION["anagrafica_data"]["ipertensione"]) &&
$_SESSION["anagrafica_data"]["ipertensione"] == $ipertensione) ? 'selected' : '';
    echo "<option value=\"\$ipertensione\" $selected>$ipertensione</option>";
}
echo '</select>
</div>';
}

```

```

echo '<script>
function toggleFumo(isSmoker) {
    var annoFumoGroup = document.getElementById("anno_fumo_group");
    var anniFumoGroup = document.getElementById("anni_fumo_group");
    if (isSmoker) {
        annoFumoGroup.style.display = "block";
        anniFumoGroup.style.display = "block";
    } else {
        annoFumoGroup.style.display = "none";
        anniFumoGroup.style.display = "none";
    }
}
</script>';

function check_anagrafica_errors(){
    if (isset($_SESSION['errors_anagrafica'])){
        $errors = $_SESSION['errors_anagrafica'];
        echo "<br>";
        foreach ($error as $error) {
            echo "<p class='error'>" . $error . "</p>";
        }
        unset($_SESSION['errors_anagrafica']);
    }
}

```

Anche per questo modulo, una volta cliccato il tasto di inserimento, è eseguita la parte del codice che si trova in 'anagrafica.inc.php': si rinominano le variabili ottenute in maniera POST, si effettuano i controlli degli errori e nel caso in cui questi fossero passati, i dati ottenuti vengono inseriti nel database tramite apposite funzioni come si può vedere di seguito.

```

<?php
if ($_SERVER["REQUEST_METHOD"] == "POST"){
    $id_paziente = $_POST["id_paziente"];
    $data_nascita = $_POST["data_nascita"];
    $etnia = $_POST["etnia"];
    $bmi = $_POST["bmi"];
    $eta_menopausa = $_POST["eta_menopausa"];
    $anno_tamoxifene = $_POST["anno_tamoxifene"];
    $mesi_tamoxifene = $_POST["mesi_tamoxifene"];
    $anno_fumo = $_POST["anno_fumo"];
    $anni_fumo = $_POST["anni_fumo"];
    $eta_mestruazione = $_POST["eta_mestruazione"];
    $numero_parti = $_POST["numero_parti"];
    $terapia_ormonale = $_POST["terapia_ormonale"];
    $diabete = $_POST["diabete"];
    $lynch = $_POST["lynch"];
    $tumore_seno = $_POST["tumore_seno"];
    $ipertensione = $_POST["ipertensione"];
    try {
        require_once "dbh.inc.php";
        require_once "anagrafica_model.inc.php";
        require_once "anagrafica_contr.inc.php";
        require_once "config_session.inc.php";
        // ERROR HANDLERS
    }
}

```

```

$errors = [];
if (is_input_empty($id_paziente, $data_nascita, $bmi)){
    $errors["empty_input"] = "Riempi tutti i campi richiesti!";
}
if (is_paziente_registered($pdo, $id_paziente)){
    $errors["paziente_esistente"] = "Il paziente inserito è già
esistente";
}
if($errors){
    $_SESSION["errors_anagrafica"] = $errors;
    $anagraficaData = [
        "id_paziente" => $id_paziente,
        "data_nascita" => $data_nascita,
        "etnia" => $etnia,
        "bmi" => $bmi,
        "eta_menopausa" => $eta_menopausa,
        "anno_tamoxifene" => $anno_tamoxifene,
        "mesi_tamoxifene" => $mesi_tamoxifene,
        "anno_fumo" => $anno_fumo,
        "anni_fumo" => $anni_fumo,
        "eta_mestruazione" => $eta_mestruazione,
        "numero_parti" => $numero_parti,
        "terapia_ormonale" => $terapia_ormonale,
        "diabete" =>$diabete,
        "lynch" => $lynch,
        "tumore_seno" => $tumore_seno,
        "ipertensione" => $ipertensione
    ];
    $_SESSION["anagrafica_data"] = $anagraficaData;
    header("Location: ../Form/anagraficaForm.php");
    die();
}
create_anagrafica($pdo, $id_paziente, $data_nascita, $etnia, $bmi,
$eta_menopausa, $anno_tamoxifene, $mesi_tamoxifene, $anno_fumo, $anni_fumo,
$eta_mestruazione, $numero_parti, $terapia_ormonale);

create_patologie($pdo, $id_paziente, $diabete, $lynch, $tumore_seno,
$ipertensione);

header("Location: ../index.php?anagrafica=success");
$pdo = null;
$stmt = null;
die();
}
catch (PDOException $e) {
    die("Query failed: " . $e->getMessage());
}
}
else {
    header("Location: ../Form/anagraficaForm.php");
    die();
}
}

```

I controlli per gli errori non sono molti dato che nella maggior parte dei campi sarà inserito '0' o '0000' di default se lasciati vuoti.

Particolare attenzione è stata invece riservata all'id delle pazienti: un utente non può inserire lo stesso identificativo per più pazienti, questo viene controllato tramite risultati di una query. Le funzioni richiamate nella parte di codice appena vista sono quelle del file 'anagrafica_contr.inc.php' che segue.

```
<?php
declare (strict_types=1);
function is_input_empty(string $id_paziente, string $data_nascita, string $bmi){
    if(empty($id_paziente) or empty($data_nascita) or empty($bmi)){
        return true;
    }
    else {
        return false;
    }
}
```

```
function is_paziente_registered(object $pdo, string $id_paziente){
    if(get_id_paziente($pdo, $id_paziente)){
        return true;
    }
    else {
        return false;
    }
}
```

```
function create_anagrafica (object $pdo, string $id_paziente, string $data_nascita,
string $etnia, string $bmi, string $eta_menopausa, string $anno_tamoxifene, string
$mesi_tamoxifene, string $anno_fumo, string $anni_fumo, string $eta_mestruazione,
string $numero_parti, string $terapia_ormonale){
    set_anagrafica($pdo, $id_paziente, $data_nascita, $etnia, $bmi,
$eta_menopausa, $anno_tamoxifene, $mesi_tamoxifene, $anno_fumo, $anni_fumo,
$eta_mestruazione, $numero_parti, $terapia_ormonale);
}
```

```
function create_patologie (object $pdo, string $id_paziente, string $diabete, string
$lynch, string $tumore_seno, string $ipertensione){
    set_patologie($pdo, $id_paziente, $diabete, $lynch, $tumore_seno,
$ipertensione);
}
```

La query che interroga il database nel caso dell'errore 'is_paziente_registered' relativo all'id della paziente e quelle che inseriscono i dati nelle tabelle 'anagrafica' e 'patologie' sono parte del file 'anagrafica_model.inc.php' e possono essere consultate di seguito.

```
<?php
declare (strict_types=1);
function get_id_paziente (object $pdo, string $id_paziente){
    $query = "SELECT * FROM anagrafica WHERE (id_utenti = :id_utenti AND id_paziente
= :id_paziente);";
    $stmt = $pdo -> prepare($query);
    $stmt->bindParam(':id_paziente', $id_paziente);
    $stmt->bindParam(':id_utenti', $_SESSION["user_id"]);
```

```

$stmt->execute();
$result = $stmt -> fetch(PDO::FETCH_ASSOC);
return $result;
}

function set_anagrafica(object $pdo, string $id_paziente, string $data_nascita, string
$etnia, string $bmi, string $eta_menopausa, string $anno_tamoxifene, string
$mesi_tamoxifene, string $anno_fumo, string $anni_fumo, string $eta_mestruazione,
string $numero_parti, string $terapia_ormonale) {

    $query = "INSERT INTO anagrafica (id_utenti, id_paziente, data_di_nascita,
etnia, bmi, eta_menopausa, tamoxifene, mesi_ultimo_trattamento_tamoxifene,
anno_inizio_fumo, anni_fumato, eta_prima_mestruazione, numero_parti, terapia_ormonale)
VALUES (:id_utenti, :id_paziente, :data_nascita, :etnia, :bmi, :eta_menopausa,
:anno_tamoxifene, :mesi_tamoxifene, :anno_fumo, :anni_fumo, :eta_mestruazione,
:numero_parti, :terapia_ormonale)";

    $stmt = $pdo -> prepare($query);
    $stmt->bindParam(':id_paziente', $id_paziente);
    $stmt->bindParam(':id_utenti', $_SESSION["user_id"]);
    $stmt->bindParam(':data_nascita', $data_nascita);
    $stmt->bindParam(':etnia', $etnia);
    $stmt->bindParam(':bmi', $bmi);
    $stmt->bindParam(':eta_menopausa', $eta_menopausa);
    $stmt->bindParam(':anno_tamoxifene', $anno_tamoxifene);
    $stmt->bindParam(':mesi_tamoxifene', $mesi_tamoxifene);
    $stmt->bindParam(':anno_fumo', $anno_fumo);
    $stmt->bindParam(':anni_fumo', $anni_fumo);
    $stmt->bindParam(':eta_mestruazione', $eta_mestruazione);
    $stmt->bindParam(':numero_parti', $numero_parti);
    $stmt->bindParam(':terapia_ormonale', $terapia_ormonale);
    $stmt->execute();
}

function set_patologie(object $pdo, string $id_paziente, string $diabete, string
$lynch, string $tumore_seno, string $ipertensione) {

    $query = "INSERT INTO patologie (id_inseritore, id_paziente, diabete, lynch,
tumore_seno, ipertensione) VALUES (:id_utenti, :id_paziente, :diabete, :lynch,
:tumore_seno, :ipertensione)";

    $stmt = $pdo -> prepare($query);
    $stmt->bindParam(':id_paziente', $id_paziente);
    $stmt->bindParam(':id_utenti', $_SESSION["user_id"]);
    $stmt->bindParam(':diabete', $diabete);
    $stmt->bindParam(':lynch', $lynch);
    $stmt->bindParam(':tumore_seno', $tumore_seno);
    $stmt->bindParam(':ipertensione', $ipertensione);
    $stmt->execute();
}

```


4.2.6 Inserimento immagini

La seconda azione disponibile nel menu del sito web consiste nell'inserimento di immagini colposcopiche nel database. Anche per questa azione, viene richiesto all'utente di fornire diversi dati attraverso un modulo. In primo luogo, l'utente deve selezionare la paziente, tra quelle precedentemente da lui inserite, a cui corrisponde l'immagine. Successivamente, deve compilare i vari campi richiesti (descritti nella tabella 'immagini' menzionata nella sezione 4.1).

Come tutti le altre parti, il file contenente il modulo da compilare (riportato di seguito) si trova nella cartella 'Form' e richiama alcune funzioni sviluppate nel file 'upload_view.inc.php'.

```
<?php
require_once '../includes/config_session.inc.php';
require_once '../includes/upload_view.inc.php';
require_once '../includes/upload_model.inc.php';
?>
<!DOCTYPE html>
<html lang="it">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Caricamento Immagine</title>
    <link rel="stylesheet" href="../style.css">
</head>
<body>
    <div class="container">
        <h2 style="color:#0056b3;">Carica immagine colposcopicca</h2>
        <form action="../includes/upload.inc.php" method="post"
enctype="multipart/form-data">
            <?php
                upload_inputs();
            ?>
            <button type="submit">Inserisci Immagine</button>
        </form>
        <br>
        <?php
            check_upload_errors();
        ?>
    </div>
    <div class="indietro">
        <a href="../index.php">Indietro</a>
    </div>
</body>
</html>
```

Il codice corrispondente alle due funzioni 'upload_inputs()' e 'check_upload_errors()' richiamate invece è il seguente:

```
<?php
function getTodayDate() {
    return date('Y-m-d');
}

function upload_inputs(){
    require_once '../includes/dbh.inc.php';
    // Check for ID del Paziente
    if (isset($_SESSION["upload_data"]["id_paziente"]) ||
(isset($_SESSION["upload_data"]["nuova_immagine"]) &&
$_SESSION["upload_data"]["nuova_immagine"] == 'si')) {
        echo '<div class="form-group">
            <label for="id_paziente">Seleziona Id della paziente</label>
            <select name="id_paziente" id="id_paziente">';
        $query = "SELECT * FROM anagrafica WHERE id_utenti = :id_utenti";
        $stmt = $pdo->prepare($query);
        $stmt->bindParam(":id_utenti", $_SESSION["user_id"]);
        $stmt->execute();
        $result = $stmt->fetchAll(PDO::FETCH_ASSOC);
        if (count($result) > 0) {
            echo '<option value="0">--Seleziona--</option>';
            foreach ($result as $row) {
                $selected = ($_SESSION["upload_data"]["id_paziente"] ==
$row["id_paziente"]) ? 'selected' : '';
                echo '<option value="' . $row["id_paziente"] . "' ' . $selected .
'>' . htmlspecialchars($row["id_paziente"]) . '</option>';
            }
        } else {
            echo '<option value="0">Nessun risultato</option>';
        }
        echo '</select>
</div>';
    } else {
        echo '<div class="form-group">
            <label for="id_paziente">Seleziona Id della paziente di cui caricare
le foto</label>
            <select name="id_paziente" id="id_paziente">';
        $query = "SELECT * FROM anagrafica WHERE id_utenti = :id_utenti";
        $stmt = $pdo->prepare($query);
        $stmt->bindParam(":id_utenti", $_SESSION["user_id"]);
        $stmt->execute();
        $result = $stmt->fetchAll(PDO::FETCH_ASSOC);
        if (count($result) > 0) {
            echo '<option value="0">--Seleziona--</option>';
            foreach ($result as $row) {
                echo '<option value="' . $row["id_paziente"] . "'>' .
htmlspecialchars($row["id_paziente"]) . '</option>';
            }
        } else {
            echo '<option value="0">Nessun risultato</option>';
        }
        echo '</select>
</div>';
    }
}
```

```

// Check for Colposcopista
if (isset($_SESSION["upload_data"]["colposcopista"]) ||
(isset($_SESSION["upload_data"]["nuova_immagine"]) &&
$_SESSION["upload_data"]["nuova_immagine"] == 'si')) {
    echo '<div class="form-group">
        <label for="colposcopista">Nome del Colposcopista:*</label>
        <input type="text" id="colposcopista" name="colposcopista" value=""
. htmlspecialchars($_SESSION["upload_data"]["colposcopista"], ENT_QUOTES) . "'>
    </div>';
} else {
    echo '<div class="form-group">
        <label for="colposcopista">Nome del Colposcopista:*</label>
        <input type="text" id="colposcopista" name="colposcopista">
    </div>';
}
$todayDate = getTodayDate();
// Check for Data Colposcopia
if (isset($_SESSION["upload_data"]["data_colposcopia"]) ||
(isset($_SESSION["upload_data"]["nuova_immagine"]) &&
$_SESSION["upload_data"]["nuova_immagine"] == 'si')) {
    echo '<div class="form-group">
        <label for="data_colposcopia">Data in cui è stata eseguita la
colposcopia:*</label>
        <input type="date" id="data_colposcopia" name="data_colposcopia"
value="" . htmlspecialchars($_SESSION["upload_data"]["data_colposcopia"], ENT_QUOTES)
. "' max="" . $todayDate . "' />
    </div>';
} else {
    echo '<div class="form-group">
        <label for="data_colposcopia">Data in cui è stata eseguita la
colposcopia:*</label>
        <input type="date" id="data_colposcopia" name="data_colposcopia"
max="" . $todayDate . "' />
    </div>';
}
// Check for Nome Strumentazione
if (isset($_SESSION["upload_data"]["strumentazione"]) ||
(isset($_SESSION["upload_data"]["nuova_immagine"]) &&
$_SESSION["upload_data"]["nuova_immagine"] == 'si')) {
    echo '<div class="form-group">
        <label for="strumentazione">Nome della strumentazione
utilizzata:*</label>
        <input type="text" id="strumentazione" name="strumentazione" value=""
. htmlspecialchars($_SESSION["upload_data"]["strumentazione"], ENT_QUOTES) . "'>
    </div>';
} else {
    echo '<div class="form-group">
        <label for="strumentazione">Nome della strumentazione
utilizzata:*</label>
        <input type="text" id="strumentazione" name="strumentazione">
    </div>';
}
// Check for Fase Ciclo
if ((isset($_SESSION["upload_data"]["ciclo"]) ||
(isset($_SESSION["upload_data"]["nuova_immagine"]) &&
$_SESSION["upload_data"]["nuova_immagine"] == 'si')) &&
!isset($_SESSION["errors_upload"]["ciclo_not_selected"])) {
    $selected_ciclo = $_SESSION["upload_data"]["ciclo"];
}

```

```

        echo '<div class="form-group">
            <label for="ciclo">Fase del ciclo mestruale:*/label>
            <select id="ciclo" name="ciclo">
                <option value="0" ' . ($selected_ciclo == '0' ? 'selected' :
'' ) . '>--Seleziona--</option>
                <option value="mestruazioni" ' . ($selected_ciclo ==
'mestruazioni' ? 'selected' : '') . '>Mestruazioni</option>
                <option value="follicolare" ' . ($selected_ciclo ==
'follicolare' ? 'selected' : '') . '>Follicolare</option>
                <option value="ovulazione" ' . ($selected_ciclo ==
'ovulazione' ? 'selected' : '') . '>Ovulazione</option>
                <option value="lutenica" ' . ($selected_ciclo == 'lutenica' ?
'selected' : '') . '>Lutenica</option>
            </select>
        </div>';
    } else {
        echo '<div class="form-group">
            <label for="ciclo">Fase del ciclo mestruale:*/label>
            <select id="ciclo" name="ciclo">
                <option value="0">--Seleziona--</option>
                <option value="mestruazioni">Mestruazioni</option>
                <option value="follicolare">Follicolare</option>
                <option value="ovulazione">Ovulazione</option>
                <option value="lutenica">Lutenica</option>
            </select>
        </div>';
    }
    // Check for colorazione
    if ((isset($_SESSION["upload_data"]["colorazione"]) ||
(isset($_SESSION["upload_data"]["nuova_immagine"]) &&
$_SESSION["upload_data"]["nuova_immagine"] == 'si')) &&
!isset($_SESSION["errors_upload"]["colorazione_not_selected"])) {
        $selected_colorazione = $_SESSION["upload_data"]["colorazione"];
        echo '<div class="form-group">
            <label for="colorazione">Colorazione:*/label>
            <select id="colorazione" name="colorazione">
                <option value="0" ' . ($selected_colorazione == '0' ?
'selected' : '') . '>--Seleziona--</option>
                <option value="acido_acetico" ' . ($selected_colorazione ==
'acido_acetico' ? 'selected' : '') . '>Acido Acetico</option>
                <option value="lugol" ' . ($selected_colorazione == 'lugol' ?
'selected' : '') . '>Lugol</option>
                <option value="bianco" ' . ($selected_colorazione == 'bianco'
? 'selected' : '') . '>Bianco</option>
            </select>
        </div>';
    } else {
        echo '<div class="form-group">
            <label for="colorazione">Colorazione:*/label>
            <select id="colorazione" name="colorazione">
                <option value="0">--Seleziona--</option>
                <option value="acido_acetico">Acido Acetico</option>
                <option value="lugol">Lugol</option>
                <option value="bianco">Bianco</option>
            </select>
        </div>';
    }
}

```

```

// Check for Endometrial Sampling Methods
if ((isset($_SESSION["upload_data"]["sampling"]) ||
(isset($_SESSION["upload_data"]["nuova_immagine"]) &&
$_SESSION["upload_data"]["nuova_immagine"] == 'si')) &&
!isset($_SESSION["errors_upload"]["sampling_not_selected"]) ) {
    $selected_sampling = $_SESSION["upload_data"]["sampling"];
    echo '<div class="form-group">
        <label for="sampling">Metodi di campionamento endometriale:*</label>
        <select id="sampling" name="sampling">
            <option value="0" ' . ($selected_sampling == '0' ? 'selected' :
'' ) . '>--Seleziona--</option>
            <option value="nessuno" ' . ($selected_sampling == 'nessuno' ?
'selected' : '' ) . '>Nessuno</option>
            <option value="rettoscopia" ' . ($selected_sampling ==
'rettoscopia' ? 'selected' : '' ) . '>Rettoscopia</option>
            <option value="isteroscopia" ' . ($selected_sampling ==
'isteroscopia' ? 'selected' : '' ) . '>Isteroscopia</option>
            <option value="curettaggio" ' . ($selected_sampling == 'curettaggio'
? 'selected' : '' ) . '>Curettaggio</option>
        </select>
    </div>';
} else {
    echo '<div class="form-group">
        <label for="sampling">Metodi di campionamento endometriale:*</label>
        <select id="sampling" name="sampling">
            <option value="0">--Seleziona--</option>
            <option value="nessuno">Nessuno</option>
            <option value="rettoscopia">Rettoscopia</option>
            <option value="isteroscopia">Isteroscopia</option>
            <option value="curettaggio">Curettaggio</option>
        </select>
    </div>';
}
// Check for Sanguinamento
if (isset($_SESSION["upload_data"]["sanguinamento"]) ||
(isset($_SESSION["upload_data"]["nuova_immagine"]) &&
$_SESSION["upload_data"]["nuova_immagine"] == 'si')) {
    $selected_sanguinamento = $_SESSION["upload_data"]["sanguinamento"];
    echo '<div class="form-group">
        <label for="sanguinamento">&Egrave presente sanguinamento uterino
anormale?:*</label>
        <select id="sanguinamento" name="sanguinamento">
            <option value="0" ' . ($selected_sanguinamento == '0' ?
'selected' : '' ) . '>No</option>
            <option value="1" ' . ($selected_sanguinamento == '1' ?
'selected' : '' ) . '>Si</option>
        </select>
    </div>';
} else {
    echo '<div class="form-group">
        <label for="sanguinamento">&Egrave presente sanguinamento uterino
anormale?:*</label>
        <select id="sanguinamento" name="sanguinamento">
            <option value="0">no</option>
            <option value="1">si</option>
        </select>
    </div>';
}
}

```

```

    // Check for Note
    if (isset($_SESSION["upload_data"]["note"]) ||
(isset($_SESSION["upload_data"]["nuova_immagine"]) &&
$_SESSION["upload_data"]["nuova_immagine"] == 'si')) {
        echo '<div class="form-group">
            <label for="note">Note diagnostiche:</label>
            <textarea id="note" name="note" rows="10" cols="82">' .
htmlspecialchars($_SESSION["upload_data"]["note"], ENT_QUOTES) . '</textarea>
        </div>';
    } else {
        echo '<div class="form-group">
            <label for="note">Note diagnostiche:</label>
            <textarea id="note" name="note" rows="10" cols="82"></textarea>
        </div>';
    }
    // Check for File
    echo '<div class="form-group">
        <label for="file">Seleziona File da caricare:</label>
        <input type="file" name="file">
    </div>';
    echo ' <div class="form-group radio-group">
        <label>Vuoi inserire un\'altra immagine relativa alla stessa
paziente?</label>
        <label for="no">
        <input type="radio" id="no" name="nuova_immagine" value="no" checked> No
        </label>
        <label for="si">
        <input type="radio" id="si" name="nuova_immagine" value="si"> Sì
        </label>
    </div>';

}

function check_upload_errors(){
    if (isset($_SESSION['errors_upload'])){
        $errors = $_SESSION['errors_upload'];

        echo "<br>";

        foreach ($errors as $error) {
            echo "<p class='error'>" . $error . "</p>";
        }

        unset($_SESSION['errors_upload']);
    }
}

```

È stata posta particolare attenzione a semplificare l’inserimento di più immagini relative alla stessa paziente tramite un bottone radio a fine modulo che permette di indicare se si desidera aggiungere un’altra immagine relativa alla stessa paziente. In caso affermativo, dopo l’invio della form, l’utente viene reindirizzato alla stessa pagina con i capi precompilati.

Una volta fatto clic sul bottone ‘inserisci immagine’, viene eseguito il codice presente in ‘upload.inc.php’ che gestisce il controllo di errori e, se superato, inserisce i record nel database.

```
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST"){
    $id_paziente = $_POST["id_paziente"];
    $colposcopista = $_POST["colposcopista"];
    $data_colposcopia = $_POST["data_colposcopia"];
    $strumentazione = $_POST["strumentazione"];
    $ciclo = $_POST["ciclo"];
    $colorazione = $_POST["colorazione"];
    $sampling = $_POST["sampling"];
    $sanguinamento = $_POST["sanguinamento"];
    $note = $_POST["note"];
    $nuova_immagine = $_POST["nuova_immagine"];

    // DIVIDO IL FILE NELLE SUE COMPONENTI DATO CHE $file é UN ARRAY
    $file = $_FILES["file"];
    $fileName = $_FILES['file']['name'];
    $fileTmpName = $_FILES['file']['tmp_name'];
    $fileSize = $_FILES['file']['size'];
    $fileError = $_FILES['file']['error'];
    $fileType = $_FILES['file']['type'];

    $fileExt = explode('.', $fileName);
    $fileActualExt = strtolower(end($fileExt));
    $allowed = array('jpg', 'jpeg', 'png', 'gif', 'tif', 'bmp', 'svg', 'pdf');
    try {
        require_once "upload_model.inc.php";
        require_once "upload_contr.inc.php";
        require_once "config_session.inc.php";
        require_once 'dbh.inc.php';
        // ERROR HANDLERS
        $errors = [];
    } catch (Exception $e) {
        $errors["empty_input"] = "Riempi tutti i campi richiesti!";
    }
    if(ciclo_not_selected($ciclo)){
        $errors["ciclo_not_selected"] = "Seleziona una fase del ciclo mestruale!";
    }
    if(colorazione_not_selected($colorazione)){
        $errors["colorazione_not_selected"] = "Seleziona una colorazione!";
    }
    if(sampling_not_selected($sampling)){
        $errors["sampling_not_selected"] = "Seleziona un metodo di campionamento!";
    }
    if(id_not_selected($id_paziente)){
        $errors["id_not_selected"] = "Scegli un id di cui inserire l'immagine!";
    }
    if(file_not_uploaded($fileName)){
        $errors["file_not_uploaded"] = "Seleziona un file da caricare!";
    }
    if (wrong_format($fileActualExt, $allowed)) {
        $errors["wrong_format"] = "Il formato del file non è supportato!";
    }
}
```

```

if (file_error($fileError)) {
    $errors["file_error"] = "C'è stato un errore nell'upload del file!";
}
if (too_big_size($fileSize)) {
    $errors["too_big_size"] = "Il file è troppo grande!";
}

if($nuova_immagine == 'si'){
    $_SESSION["errors_upload"] = $errors;
    $uploadData = [
        "id_paziente" => $id_paziente,
        "colposcopista" => $colposcopista,
        "data_colposcopia" => $data_colposcopia,
        "strumentazione" => $strumentazione,
        "ciclo" => $ciclo,
        "colorazione" => $colorazione,
        "sampling" => $sampling,
        "sanguinamento" => $sanguinamento,
        "note" => $note,
        "file" => $file,
        "nuova_immagine" => $nuova_immagine
    ];
    $_SESSION["upload_data"] = $uploadData;
}
if ($errors) {
    $_SESSION["errors_upload"] = $errors;
    $uploadData = [
        "id_paziente" => $id_paziente,
        "colposcopista" => $colposcopista,
        "data_colposcopia" => $data_colposcopia,
        "strumentazione" => $strumentazione,
        "ciclo" => $ciclo,
        "colorazione" => $colorazione,
        "sampling" => $sampling,
        "sanguinamento" => $sanguinamento,
        "note" => $note,
        "file" => $file,
        "nuova_immagine" => $nuova_immagine
    ];
    $_SESSION["upload_data"] = $uploadData;
    header("Location: ../Form/upload.php");
    die();
}
$userId = $_SESSION["user_id"];
$result_anagrafica = get_anagrafica_id($pdo, $id_paziente);
if ($result_anagrafica) {
    $id_tabella_anagrafica = $result_anagrafica["id"];
}
$result_immagini = get_status($pdo);
if ($result_immagini) {
    $imgId = $result_immagini["Auto_increment"];
    //CREA IL NUOVO NOME DEL FILE DEL TIPO S123_101.jpg
    $fileNewName = $id_paziente . "_" . $imgId . "." . $fileActualExt;
    $fileDestination = '../img_upload/' . $userId . '/' . $fileNewName;
    move_uploaded_file($fileTmpName, $fileDestination);
} else {
    echo "Non è stato possibile ottenere lo status della tabella.";
}

```



```

        header("Location: ../Form/upload.php");
        die();
    }
    create_immagini($pdo, $id_tabella_anagrafica, $userId, $fileDestination, $note,
    $colposcopista, $data_colposcopia, $strumentazione, $ciclo, $colorazione, $sampling,
    $sanguinamento);
        if (isset($_SESSION["upload_data"]["nuova_immagine"]) &&
    $_SESSION["upload_data"]["nuova_immagine"] == 'no') {
            header("Location: ../index.php?upload=success");
        }
        $pdo = null;
        $stmt = null;
        die();
    }
    catch (PDOException $e) {
        die("Query failed: " . $e->getMessage());
    }
}
else {
    header("Location: ../Form/upload.php");
    die();
}
}

```

Per questo modulo, la parte relativa agli errori è più sostanziosa poiché il file deve essere ben controllato. Si pone ad esempio attenzione sul formato del file e sulla grandezza massima raggiungibile in byte. Queste funzioni sono state scritte nel ‘_contr.inc.php’ come segue.

```

<?php
declare (strict_types=1);
// FUNZIONI DI CONTROLLO ERRORI PER INPUT VUOTI, NON SELEZIONATI O ERRORI NEL FILE

function is_input_empty(string $colposcopista, string $data_colposcopia, string
    $strumentazione, string $ciclo){
    if(empty($colposcopista) or empty($data_colposcopia) or empty($strumentazione)
    or empty($ciclo)){
        return true;
    }
    else {
        return false;}
    }
function file_not_uploaded(string $fileName){
    if (empty ($fileName)) {
        return true;
    }
    else {
        return false;}
}
function id_not_selected(string $id_paziente){
    if ($id_paziente == "0"){
        return true;
    }else{
        return false;
    }
}
}

```

```

function colorazione_not_selected(string $colorazione){
    if ($colorazione == "0"){
        return true;
    }
    else {
        return false;
    }
} //PER TUTTI I CAMPI SELEZIONABILI AGGIUNGERE FUNZIONE COME QUESTA

//CONTROLLO SE L'ESTENSIONE DEL FILE È NELL'ARRAY DA ME CREATO
function wrong_format(string $fileActualExt, array $allowed){
    if (!in_array($fileActualExt, $allowed)){
        return true;
    }
    else {
        return false;
    }
}
function file_error(int $fileError){
    if ($fileError != 0){
        return true;
    }
    else {
        return false;
    }
}
// CONTROLLO SE LA GRANDEZZA DEL FILE SUPERA 100.000.000 DI BYTE (500MB)
function too_big_size(int $fileSize){
    if ($fileSize > 500000000){
        return true;
    }
    else {
        return false;
    }
}

function create_immagini(object $pdo, string $id_tabella_anagrafica, string $userId,
string $fileDestination, string $colposcopista, string $data_colposcopia, string
$strumentazione, string $ciclo, string $colorazione, string $sampling, string
$sanguinamento, string $note){
    set_immagini($pdo, $id_tabella_anagrafica, $userId, $fileDestination,
$colposcopista, $data_colposcopia, $strumentazione, $ciclo, $colorazione, $sampling,
$sanguinamento, $note);
}

```

Le query che interrogano il database e inseriscono in esso le immagini se tutto è andato a buon fine seguono le stesse modalità di quelle viste finora ma per chiunque voglia, si possono controllare nella pagina seguente.

```

<?php
declare (strict_types=1);
function get_anagrafica_id (object $pdo, string $id_paziente){
    $query = "SELECT * FROM anagrafica WHERE (id_paziente = :id_paziente AND
id_utenti = :id_utenti)";
    $stmt = $pdo -> prepare($query);
    $stmt->bindParam(":id_paziente", $id_paziente);
    $stmt->bindParam(':id_utenti', $_SESSION["user_id"]);
    $stmt->execute();

    $result_anagrafica = $stmt -> fetch(PDO::FETCH_ASSOC);
    return $result_anagrafica;
}

//OTTIENI LO STATUS DELLA TABELLA IMMAGINI PER IL NUMERO DEL PROSSIMO AUTO_INCREMENT
function get_status(object $pdo){
    $query = "SHOW TABLE STATUS LIKE 'immagini'";
    $stmt = $pdo -> prepare($query);
    $stmt->execute();
    $result_immagini = $stmt -> fetch(PDO::FETCH_ASSOC);
    return $result_immagini;
}

function set_immagini(object $pdo, string $id_tabella_anagrafica, string $userId,
string $fileDestination, string $colposcopista, string $data_colposcopia, string
$strumentazione, string $ciclo, string $colorazione, string $sampling, string
$sanguinamento, string $note) {

    $query = "INSERT INTO immagini (id_tabella_anagrafica, id_inseritore,
percorso_file, colposcopista, data_colposcopia, strumentazione, fase_ciclo,
colorazione, endometrial_sampling_methods, sanguinamento_anormale, note_diagnostiche)
VALUES (:id_tabella_anagrafica, :userId, :fileDestination, :colposcopista,
:data_colposcopia, :strumentazione, :ciclo, :colorazione, :sampling, :sanguinamento,
:note)";

    $stmt = $pdo -> prepare($query);
    $stmt->bindParam(':id_tabella_anagrafica', $id_tabella_anagrafica);
    $stmt->bindParam(':userId', $userId);
    $stmt->bindParam(':fileDestination', $fileDestination);
    $stmt->bindParam(':colposcopista', $colposcopista);
    $stmt->bindParam(':data_colposcopia', $data_colposcopia);
    $stmt->bindParam(':strumentazione', $strumentazione);
    $stmt->bindParam(':ciclo', $ciclo);
    $stmt->bindParam(':colorazione', $colorazione);
    $stmt->bindParam(':sampling', $sampling);
    $stmt->bindParam(':sanguinamento', $sanguinamento);
    $stmt->bindParam(':note', $note);
    $stmt->execute();
}

```

4.2.7 Galleria

Nel caso in cui un utente voglia vedere le immagini da lui inserite, si è deciso di aggiungere al menù la possibilità di entrare nella pagina ‘Galleria’. Qui l’utente può selezionare l’identificativo della paziente (tra quelle da lui inserite) per visualizzare le immagini a lei riferite.

In questa parte di codice è stata inclusa una piccola parte di linguaggio di programmazione JavaScript che consente di aprire le immagini a schermo intero e navigare tra esse con le freccette da tastiera o con delle frecce cliccabili all’interno dell’interfaccia grafica.

Successivamente è riportato il file ‘galleria.php’ di cui si è parlato.

```
<?php
require_once '../includes/dbh.inc.php';
require_once '../includes/config_session.inc.php';
$user_id = $_SESSION["user_id"];
    // Ottenere gli ID dei pazienti inseriti dall'utente
$query = $pdo->prepare("SELECT id_paziente FROM anagrafica WHERE id_utenti =
:user_id");
$query->execute(['user_id' => $user_id]);
$pazienti = $query->fetchAll(PDO::FETCH_ASSOC);
?>

<!DOCTYPE html>
<html lang="it">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Galleria di Immagini</title>
    <style>
body {
    font-family: Arial, sans-
serif;
    display: flex;
    flex-direction: column;
    align-items: center;
    background-color: #f4f4f4;
    padding: 20px;
    margin: 0;
}
h1 {
    color: #0056b3;
    margin-bottom: 20px;
    font-size: 40px;
}
.form-group {
    margin-bottom: 15px;
}
label {
    display: block;
    margin-bottom: 5px;
    font-size: 25px;
}
select {
    width: 100%;
    padding: 10px;
    border: 1px solid #ccc;
    border-radius: 4px;
    box-sizing: border-box;
    cursor: pointer;
    font-size: 18px;
}
.gallery {
    display: flex;
    flex-wrap: wrap;
    gap: 15px;
    justify-content: center;
    position: relative; /*
Aggiunto per posizionare le frecce */
}
.gallery img {
    width: 200px;
```

```

        height: 200px;
        object-fit: cover;
        border-radius: 10px;
        box-shadow: 0 4px 8px
rgba(0, 0, 0, 0.2);
        cursor: pointer;
        transition: transform 0.3s,
box-shadow 0.3s;
    }
    .gallery img:hover {
        transform: scale(1.05);
        box-shadow: 0 8px 16px
rgba(0, 0, 0, 0.3);
    }
    .fullscreen {
        display: none;
        position: fixed;
        top: 0;
        left: 0;
        width: 100%;
        height: 100%;
        background: rgba(0, 0, 0,
0.8);
        justify-content: center;
        align-items: center;
        z-index: 1000;
    }
    .fullscreen img {
        max-width: 90%;
        max-height: 90%;
        box-shadow: 0 4px 8px
rgba(0, 0, 0, 0.5);
    }
    .fullscreen .close-btn {
        position: absolute;
        top: 20px;
        right: 30px;
        font-size: 30px;
        color: white;
        cursor: pointer;
        transition: color 0.3s;
    }
    .fullscreen .close-btn:hover {
        color: #f0f0f0;
    }

```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1>Galleria</h1>
```

```
<div class="form-group">
```

```
<form method="POST" id="pazienteForm">
```

```
<label for="paziente">Seleziona Paziente:</label>
```

```
<select name="paziente" id="paziente"
```

```
onchange="document.getElementById('pazienteForm').submit();">
```

```
<option value="0">Seleziona...</option>
```

```
<?php foreach ($pazienti as $paziente): ?>
```

```
<option value="<?=" htmlspecialchars($paziente['id_paziente'])
```

```
?>" <?=" (isset($_POST['paziente']) && $_POST['paziente'] == $paziente['id_paziente'])
```

```
? 'selected' : '' ?>>
```

```

    }
    .arrow {
        position: absolute;
        top: 50%;
        transform: translateY(-
50%);
        color: #fff;
        font-size: 36px;
        cursor: pointer;
        z-index: 1000;
    }
    .arrow-left {
        left: 20px;
    }
    .arrow-right {
        right: 20px;
    }
    .arrow:hover {
        color: #f0f0f0;
    }
    .indietro {
        width: 100%;
        display: flex;
        justify-content: flex-
start;
        align-items: flex-end;
        position: absolute;
        bottom: 20px;
        left: 20px;
    }
    .indietro a {
        padding: 10px 20px;
        background-color: #007bff;
        color: white;
        text-decoration: none;
        border-radius: 5px;
        transition: background-
color 0.3s ease, transform 0.3s ease;
        box-shadow: 0 4px 6px
rgba(0, 0, 0, 0.1);
    }
    .indietro a:hover {
        background-color: #0056b3;
    }

```

```

                <?= htmlspecialchars($paziente['id_paziente']) ?>
            </option>
        <?php endforeach; ?>
    </select>
</form>
</div>
<br>
<br>
<div class="gallery">
    <?php
        if (isset($_POST['paziente']) && $_POST['paziente'] != "") {
            $paziente_id = $_POST['paziente'];
            require '../includes/dbh.inc.php'; // File di configurazione con la
            connessione al database
            if ($stmt = $pdo->prepare("SELECT i.percorso_file FROM anagrafica a JOIN
            immagini i ON a.id = i.id_tabella_anagrafica WHERE a.id_paziente = :id_paziente AND
            a.id_utenti = :id_utenti")) {
                $stmt->bindParam(':id_paziente', $paziente_id);
                $stmt->bindParam(':id_utenti', $user_id);
                $stmt->execute();
                $result = $stmt->fetchAll(PDO::FETCH_ASSOC);
                foreach ($result as $row) {
                    echo '';
                }
                $stmt->closeCursor();
            }
            $pdo = null;
        }
    ?>
</div>
<div class="fullscreen" id="fullscreen">
    <span class="close-btn">&times;</span>
    <img class="fullscreen-img" src="" alt="Fullscreen Image">
    <div class="arrow arrow-left">&lt;</div>
    <div class="arrow arrow-right">&gt;</div>
</div>
<script>
    document.addEventListener('DOMContentLoaded', () => {
        const galleryImgs = document.querySelectorAll('.gallery-img');
        const fullscreenView = document.querySelector('.fullscreen');
        const fullscreenImg = document.querySelector('.fullscreen-img');
        const closeBtn = document.querySelector('.close-btn');
        let arrowLeft, arrowRight; // Definiamo le variabili globali per le frecce
        let currentImgIndex = 0;
        // Aggiungi event listener per cliccare sulle immagini
        galleryImgs.forEach((img, index) => {
            img.addEventListener('click', () => {
                currentImgIndex = index;
                showFullscreenImage(index);
            });
        });
        // Funzione per mostrare immagine a schermo intero
        function showFullscreenImage(index) {
            fullscreenImg.src = galleryImgs[index].src;
            fullscreenView.style.display = 'flex';
            // Mostra le frecce di navigazione
            if (!arrowLeft && !arrowRight) {

```

```

        showNavigationArrows();
    }
    // Aggiungi event listener per chiudere a schermo intero
    closeBtn.addEventListener('click', closeFullscreen);
    // Aggiungi event listener per navigare con le frecce
    document.addEventListener('keydown', navigateWithArrows);
}
// Funzione per mostrare le frecce di navigazione
function showNavigationArrows() {
    arrowLeft = document.createElement('div');
    arrowLeft.classList.add('arrow', 'arrow-left');
    arrowLeft.innerHTML = '<';
    arrowLeft.addEventListener('click', navigateLeft);
    fullscreenView.appendChild(arrowLeft);
    arrowRight = document.createElement('div');
    arrowRight.classList.add('arrow', 'arrow-right');
    arrowRight.innerHTML = '>';
    arrowRight.addEventListener('click', navigateRight);
    fullscreenView.appendChild(arrowRight);
}
// Funzione per chiudere l'immagine a schermo intero
function closeFullscreen() {
    fullscreenView.style.display = 'none';
    document.removeEventListener('keydown', navigateWithArrows);
    closeBtn.removeEventListener('click', closeFullscreen);
    // Nascondi le frecce di navigazione
    if (arrowLeft && arrowRight) {
        fullscreenView.removeChild(arrowLeft);
        fullscreenView.removeChild(arrowRight);
        arrowLeft = null;
        arrowRight = null;
    }
}
// Funzione per navigare a sinistra
function navigateLeft() {
currentImgIndex = (currentImgIndex - 1 + galleryImgs.length) % galleryImgs.length;
    fullscreenImg.src = galleryImgs[currentImgIndex].src;
}
// Funzione per navigare a destra
function navigateRight() {
    currentImgIndex = (currentImgIndex + 1) % galleryImgs.length;
    fullscreenImg.src = galleryImgs[currentImgIndex].src;
}
// Funzione per navigare con le frecce sulla tastiera
function navigateWithArrows(e) {
    if (e.key === 'ArrowLeft') {
        navigateLeft();
    } else if (e.key === 'ArrowRight') {
        navigateRight();
    }
}
});
</script>
<div class="indietro">
    <a href="../index.php">Indietro</a>
</div>
</body>
</html>

```

4.2.8 Visualizza schede pazienti

Una funzionalità aggiunta all'interno del sito web è la possibilità di visualizzare le schede delle pazienti inserite in una tabella nella pagina 'Rivedi o modifica schede di pazienti già inserite'. Il file 'gestioneAnagrafica.php' si occupa di questa sezione.

La pagina HTML visualizza una tabella con i dati delle pazienti e include funzioni di ordinamento sui vari campi. Ogni colonna della tabella ha un link che permette di ordinare i dati in essa contenuti in modo ascendente o discendente. Per rendere la pagina più leggibile e interattiva è stato inserito dello stile CSS. Nell'ultima casella di ogni riga, è presente un tasto 'vedi immagini'; se premuto reinvia alla galleria mostrando le immagini relative alla paziente il cui identificativo è posto nella prima casella della stessa riga.

Nel caso l'utente voglia modificare i dati di qualche paziente, sono stati aggiunti dei bottoni che consentono di modificare vari attributi relativi ai dati anagrafici e patologici di cui si discuterà in seguito.

È stato inoltre aggiunto un messaggio di conferma se la modifica ai dati è stata effettuata con successo.

Il codice che genera la pagina è scritto di seguito e prosegue nelle prossime pagine.

```
<?php
require_once '../includes/config_session.inc.php';
require_once '../includes/anagrafica_view.inc.php';
require_once '../includes/dbh.inc.php';
// Leggi i parametri di ordinamento dalla query string
$orderby = isset($_GET['orderby']) ? $_GET['orderby'] : 'a.id_paziente';
$orderdir = isset($_GET['orderdir']) && $_GET['orderdir'] === 'desc' ? 'desc' :
'asc';
// Mappa delle colonne valide per l'ordinamento
$valid_columns = [
    'a.id_paziente', 'a.data_di_nascita', 'a.etnia', 'a.bmi', 'a.eta_menopausa',
    'a.tamoxifene',
    'a.mesi_ultimo_trattamento_tamoxifene', 'a.anno_inizio_fumo', 'a.anni_fumato',
    'a.eta_prima_mestruazione',
    'a.numero_parti', 'a.terapia_ormonale', 'p.diabete', 'p.lynch', 'p.tumore_seno',
    'p.ipertensione'
];
// Verifica che la colonna di ordinamento sia valida
if (!in_array($orderby, $valid_columns)) {
    $orderby = 'a.id_paziente';
}
// Costruisci la query con l'ordinamento
$query = "SELECT * FROM anagrafica a INNER JOIN patologie p ON a.id_utenti =
p.id_inseritore AND a.id_paziente = p.id_paziente WHERE a.id_utenti = :id_utenti
ORDER BY $orderby $orderdir";
$stmt = $pdo->prepare($query);
```



```

$stmt->bindParam(":id_utenti", $_SESSION["user_id"]);
$stmt->execute();
$result = $stmt->fetchAll(PDO::FETCH_ASSOC);
?>
<!DOCTYPE html>
<html lang="it">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Tabella Anagrafica</title>
  <style>
    body {
      font-family: 'Arial', sans-
serif;
      margin: 0;
      padding: 20px;
      justify-content: center;
      align-items: center;
      min-height: 100vh;
      background-color: #f4f4f4;
    }
    h1 {
      text-align: center;
      color: #0056b3;
    }
    table {
      width: 100%;
      border-collapse: collapse;
      margin: 20px 0;
      background-color: #fff;
      box-shadow: 0 2px 5px
rgba(0, 0, 0, 0.1);
    }
    table th, table td {
      padding: 12px 15px;
      border: 1px solid #ddd;
      text-align: left;
    }
    table th {
      background-color: #007bff;
      color: white;
      text-transform: uppercase;
      cursor: pointer;
    }
    table tr:nth-child(even) {
      background-color: #f2f2f2;
    }
    table tr:hover {
      background-color: #f1f1f1;
    }
    .button-container {
      display: flex;
      flex-wrap: wrap;
      justify-content: center;
      gap: 10px;
      max-width: 100%;
      overflow-x: auto;
      padding: 10px;
      margin-top: 20px;
    }
    .button-container a {
      display: inline-block;
      padding: 10px 20px;
      background-color: #007bff;
      color: white;
      text-decoration: none;
      border-radius: 5px;
      transition: background-
color 0.3s ease, transform 0.3s ease;
      box-shadow: 0 4px 6px
rgba(0, 0, 0, 0.1);
      white-space: nowrap;
      font-size: 18px;
    }
    .button-container a:hover {
      background-color: #0056b3;
    }
    .success-message {
      color: green;
      font-weight: bold;
      margin: 20px 0;
      text-align: center;
      padding: 10px;
      border: 1px solid green;
      border-radius: 5px;
      background-color: #e8f5e9;
    }
    .indietro {
      width: 100%;
      display: flex;
      justify-content: flex-
start;
      align-items: flex-end;
      position: absolute;
      bottom: 20px;
      left: 20px;
    }
    .indietro a {
      padding: 10px 20px;
      background-color: #007bff;
      color: white;
      text-decoration: none;
      border-radius: 5px;
      transition: background-
color 0.3s ease, transform 0.3s ease;

```

```

        box-shadow: 0 4px 6px
        rgba(0, 0, 0, 0.1);
    }
    .indietro a:hover {
        background-color: #0056b3;
    }

    .vedi-immagini-button {
        display: inline-block;
        padding: 8px 16px;
        background-color: #007bff;
        color: white;
        text-decoration: none;
        border: none;
        border-radius: 5px;

```

```

        transition: background-
        color 0.3s ease, transform 0.3s ease;
        font-size: 14px;
        cursor: pointer;
        box-shadow: 0 4px 6px
        rgba(0, 0, 0, 0.1);
    }
    .vedi-immagini-button:hover {
        background-color: #0056b3;
        transform: translateY(-
        2px);
    }
    .vedi-immagini-button:active {
        background-color: #1e7e34;
        transform: translateY(0);
    }

```

```

</style>
</head>
<body>
<h1>Tabella Anagrafica Pazienti</h1>
<table><thead><tr>
    <th><p>ID Paziente</p><a href="?orderby=a.id_paziente&orderdir=<?php echo
    $orderdir === 'asc' ? 'desc' : 'asc'; ?>">Ordina</a></th>
    <th><p>Data di nascita</p><a
    href="?orderby=a.data_di_nascita&orderdir=<?php echo $orderdir === 'asc' ? 'desc' :
    'asc'; ?>">Ordina</a></th>
    <th><p>Etnia</p><a href="?orderby=a.etnia&orderdir=<?php echo $orderdir
    === 'asc' ? 'desc' : 'asc'; ?>">Ordina</a></th>
    <th><p>BMI</p><a href="?orderby=a.bmi&orderdir=<?php echo $orderdir ===
    'asc' ? 'desc' : 'asc'; ?>">Ordina</a></th>
    <th><p>Età della menopausa</p><a
    href="?orderby=a.eta_menopausa&orderdir=<?php echo $orderdir === 'asc' ? 'desc' :
    'asc'; ?>">Ordina</a></th>
    <th><p>Anno ultima terapia tamoxifene</p><a
    href="?orderby=a.tamoxifene&orderdir=<?php echo $orderdir === 'asc' ? 'desc' : 'asc';
    ?>">Ordina</a></th>
    <th><p>Mesi ultimo trattamento tamoxifene</p><a
    href="?orderby=a.mesi_ultimo_trattamento_tamoxifene&orderdir=<?php echo $orderdir ===
    'asc' ? 'desc' : 'asc'; ?>">Ordina</a></th>
    <th><p>Anno inizio fumo</p><a
    href="?orderby=a.anno_inizio_fumo&orderdir=<?php echo $orderdir === 'asc' ? 'desc' :
    'asc'; ?>">Ordina</a></th>
    <th><p>Numero anni che la paziente ha fumato</p><a
    href="?orderby=a.anni_fumato&orderdir=<?php echo $orderdir === 'asc' ? 'desc' :
    'asc'; ?>">Ordina</a></th>
    <th><p>Età del menarca</p><a
    href="?orderby=a.eta_prima_mestruazione&orderdir=<?php echo $orderdir === 'asc' ?
    'desc' : 'asc'; ?>">Ordina</a></th>
    <th><p>Numero di gestazioni avute</p><a
    href="?orderby=a.numero_parti&orderdir=<?php echo $orderdir === 'asc' ? 'desc' :
    'asc'; ?>">Ordina</a></th>
    <th><p>Terapie ormonale</p><a
    href="?orderby=a.terapia_ormonale&orderdir=<?php echo $orderdir === 'asc' ? 'desc' :
    'asc'; ?>">Ordina</a></th>
    <th><p>Diabete</p><a href="?orderby=p.diabete&orderdir=<?php echo
    $orderdir === 'asc' ? 'desc' : 'asc'; ?>">Ordina</a></th>

```

```

        <th><p>Sindrome di Lynch</p><a href="?orderby=p.lynch&orderdir=<?php echo
$orderdir === 'asc' ? 'desc' : 'asc'; ?>">Ordina</a></th>
        <th><p>Tumore al seno</p><a href="?orderby=p.tumore_seno&orderdir=<?php
echo $orderdir === 'asc' ? 'desc' : 'asc'; ?>">Ordina</a></th>
        <th><p>Ipertensione</p><a href="?orderby=p.ipertensione&orderdir=<?php
echo $orderdir === 'asc' ? 'desc' : 'asc'; ?>">Ordina</a></th>
        <th>Vedi Immagini</th>
    </tr></thead><tbody>
<?php
if (count($result) > 0) { foreach ($result as $row) { ?>
<tr>
<td><?php echo htmlspecialchars($row['id_paziente']); ?></td>
<td><?php echo htmlspecialchars($row['data_di_nascita']); ?></td>
<td><?php echo htmlspecialchars($row['etnia']); ?></td>
<td><?php echo htmlspecialchars($row['bmi']); ?></td>
<td><?php echo htmlspecialchars($row['eta_menopausa']); ?></td>
<td><?php echo htmlspecialchars($row['tamoxifene']); ?></td>
<td><?php echo htmlspecialchars($row['mesi_ultimo_trattamento_tamoxifene']);?></td>
<td><?php echo htmlspecialchars($row['anno_inizio_fumo']); ?></td>
<td><?php echo htmlspecialchars($row['anni_fumato']); ?></td>
<td><?php echo htmlspecialchars($row['eta_prima_mestruazione']); ?></td>
<td><?php echo htmlspecialchars($row['numero_parti']); ?></td>
<td><?php echo htmlspecialchars($row['terapia_ormonale']); ?></td>
<td><?php echo htmlspecialchars($row['diabete']); ?></td>
<td><?php echo htmlspecialchars($row['lynch']); ?></td>
<td><?php echo htmlspecialchars($row['tumore_seno']); ?></td>
<td><?php echo htmlspecialchars($row['ipertensione']); ?></td>
<td> <form action="galleria.php" method="post">
        <input type="hidden" name="paziente" value="<?php echo
htmlspecialchars($row['id_paziente']); ?>">
<button type="submit" class="vedi-immagini-button">Vedi Immagini</button></form></td>
</tr>
        <?php } }?></tbody></table>
<div class="button-container">
    <a href="modificaId.php">Modifica Id Paziente</a>
    <a href="modificaNascita.php">Modifica Data di nascita</a>
    <a href="modificaEtnia.php">Modifica Etnia</a>
    <a href="modificaBmi.php">Modifica Bmi</a>
    <a href="modificaMenopausa.php">Modifica Età della menopausa</a>
    <a href="modificaTamoxifene.php">Modifica Anno di inizio dell'ultima terapia a
base di tamoxifene</a>
    <a href="modificaMesiTamox.php">Modifica Mesi di utilizzo di tamoxifene
nell'ultima terapia</a>
    <a href="modificaFumo.php">Modifica Anno di inizio del fumo</a>
    <a href="modificaAnniFumo.php">Modifica Anni di fumo</a>
    <a href="modificaMestruazione.php">Modifica Età della prima mestruazione</a>
    <a href="modificaParti.php">Modifica Numero di gestazioni che ha avuto la
paziente</a>
    <a href="modificaOrm.php">Modifica Valore della terapia ormonale</a>
    <a href="modificaPatologie.php">Modifica Patologie</a>
</div>
<?php if (isset($_GET["modifica"]) && $_GET["modifica"] === "success") {
echo"<p class='success-message'>LA MODIFICA FATTA è AVVENUTA CON SUCCESSO! </p>";}?>
<div class="indietro">
    <a href=" ../index.php">Indietro</a>
</div>
</body>
</html>

```

4.2.9 Gestione account

Nel caso in cui l'utente volesse vedere i suoi stessi dati (nome, cognome e sede lavorativa) per verificare che siano corretti, può tranquillamente entrare nella sezione 'Account'. Il codice relativo a questa pagina mostra una tabella con i campi sopra detti relativi all'utente connesso, presi dal database tramite una query. Se l'utente si accorge che uno o più dati sono errati, è a lui data la possibilità di modificarli tramite dei tasti sottostanti.

```
<?php
require_once '../includes/config_session.inc.php';
require_once '../includes/anagrafica_view.inc.php';
require_once '../includes/login_view.inc.php';
require_once '../includes/dbh.inc.php';
$query = "SELECT * FROM utenti WHERE id = :id";
$stmt = $pdo -> prepare($query);
$stmt->bindParam(":id", $_SESSION["user_id"]);
$stmt->execute();
$result = $stmt -> fetchAll(PDO::FETCH_ASSOC);?>
<!DOCTYPE html>
<html lang="it">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Gestione Utenti</title>
</head>
<body>
<div class="content">
    <h1>Utente</h1>
    <table><thead><tr>
        <th>Nome</th>
        <th>Cognome</th>
        <th>Sede lavorativa</th>
    </tr></thead>
    <tbody><?php if (count($result) > 0) {
        foreach ($result as $row) {
            echo "<tr>";
            echo "<td>" . htmlspecialchars($row['nome']) . "</td>";
            echo "<td>" . htmlspecialchars($row['cognome']) . "</td>";
            echo "<td>" . htmlspecialchars($row['sede']) . "</td>";
            echo "</tr>";}}?>
    </tbody></table>
    <div class="button-container">
        <a href="modificaNome.php">Modifica Nome</a>
        <a href="modificaCognome.php">Modifica Cognome</a>
        <a href="modificaSede.php">Modifica Sede lavorativa</a>
        <a href="modificaPwd.php">Modifica Password</a>
    </div>
<?php if(isset($_GET["modifica"]) && $_GET["modifica"] === "success"){
    echo "<p class='success-message'>La modifica è avvenuta con successo!</p>";}?>
<div class="indietro"><a href="../index.php">Indietro</a></div>
</div>
</body>
</html>
```

4.2.10 Modifica

Come accennato nei sottocapitoli 4.2.8 e 4.2.9, si è deciso di concedere all'utente la libertà di poter modificare alcuni dati da lui inseriti nel database. Per garantire il corretto funzionamento del processo, ciascun modulo di modifica è stato suddiviso in quattro file distinti.

Ogni voce modificabile dispone quindi di quattro file propri, ma poiché le parti di codice sono molto simili tra le diverse voci, per evitare ripetizioni verrà analizzato il codice corrispondente a una sola voce: la data di nascita della paziente.

Il primo file riportato corrisponde al modulo HTML stesso nel quale tramite una query si interroga il database per far scegliere all'utente l'identificativo della paziente (tra quelle da lui inserite) di cui desidera modificare i dati. Come secondo input, in questo caso è presente la scelta di una data (giorno, mese e anno).

```
<?php
require_once '../includes/config_session.inc.php';
require_once '../includes/Modifica/modifica_view.inc.php';
require_once '../includes/dbh.inc.php';
?>
<!DOCTYPE html>
<html lang="it">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Modifica Data di nascita paziente</title>
    <link rel="stylesheet" href="../style.css">
</head>
<body>

<div class="container">
    <h2>Modifica Data di nascita della paziente</h2>
    <form action="../includes/modificaAnagrafica/modNascita.inc.php" method="post">
        <div class="form-group">
            <?php echo '<label for="id">Id:' . $_SESSION["user_id"] . '</label>'?>
        </div>
        <div class="form-group">
            <label for="vecchio_id">Seleziona Id della paziente da modificare:</label>
            <select name="vecchio_id" id="vecchio_id">
                <?php
                $query = "SELECT * FROM anagrafica WHERE id_utenti = :id_utenti";
                $stmt = $pdo -> prepare($query);
                $stmt->bindParam(":id_utenti", $_SESSION["user_id"]);
                $stmt->execute();
                $result = $stmt -> fetchAll(PDO::FETCH_ASSOC);
                if (count($result)>0) {
                    echo '<option value="0"> --Seleziona-- </option>';
```

```

        foreach ($result as $row) {
            echo '<option value="' . $row["id_paziente"] . '">' .
htmlspecialchars($row["id_paziente"]) . '</option>';
        }
    } else {
        echo '<option value="0">Nessun risultato</option>';
    }
?>
</select>
</div>
<div class="form-group">
    <label for="nuova_nascita">Nuova data di Nascita:*</label>
    <?php
        // Funzione per ottenere la data odierna in formato YYYY-MM-DD
        function getTodayDate() {
            return date('Y-m-d');
        }
        // Ottieni la data odierna
        $todayDate = getTodayDate();
    ?>
    <input type="date" id="nuova_nascita" name="nuova_nascita" max="<?php
echo $todayDate; ?>">
    </div>
    <button type="submit">Modifica Data di nascita paziente</button>
</form>
<br>
<?php check_errors(); ?>
</div>
<div class="indietro">
    <a href=" ../Form/gestioneAnagrafica.php">Indietro</a>
</div>
</body>
</html>

```

Una volta cliccato il tasto di modifica, viene eseguito il codice ‘.inc.php’ sottostante, richiamato nella sezione ‘action’ della form.

```

<?php
if ($_SERVER["REQUEST_METHOD"] == "POST"){
    $nuova_nascita = $_POST["nuova_nascita"];
    $vecchio_id = $_POST["vecchio_id"];
    try {
        require_once "../dbh.inc.php";
        require_once "modNascita_model.inc.php";
        require_once "modNascita_contr.inc.php";
        require_once "../config_session.inc.php";
        // ERROR HANDLERS
        $errors = [];
        if (is_input_empty($nuova_nascita)){
            $errors["empty_input"] = "Riempi i campi richiesti!";
        }
        if(id_not_selected($vecchio_id)){
            $errors["id_not_selected"] = "Scegli un id da modificare!";
        }
    }
}

```

```

        if($errors){
            $_SESSION["errors_modifica"] = $errors;
            $modificaData = [
                "nuova_nascita" => $nuova_nascita,
                "vecchio_id" => $vecchio_id
            ];
            $_SESSION["modifica_data"] = $modificaData;
            header("Location:http://localhost/Prova_Sito/Form/modificaNascita.php");
            die();
        }
        set_data($pdo, $nuova_nascita, $vecchio_id);
        header("Location: ../../Form/gestioneAnagrafica.php?modifica=success");
        $pdo = null;
        $stmt = null;
        die();
    }
    catch (PDOException $e) {
        die("Query failed: " . $e->getMessage());
    }
}
else {
    header("Location: http://localhost/Prova_Sito/Form/modificaNascita.php");
    die();
}
}

```

Le righe soprastanti controllano se siano stati passati i dati richiesti in forma POST, se la consegna è andata a buon fine li rinomina ed esegue il controllo relativo agli errori. Nel caso si superi anche questo tipo di controllo, i dati vengono inseriti nel database tramite la funzione ‘set_data’.

Le funzioni richiamate si trovano nel file ‘_contr.inc.php’ che segue.

```

<?php
declare (strict_types=1);
function is_input_empty(string $nuova_nascita){
    if(empty($nuova_nascita)){
        return true;
    }
    else {
        return false;
    }
}
function id_not_selected(string $vecchio_id){
    if ($vecchio_id == "0"){
        return true;
    }else{
        return false;
    }
}
function update_data(object $pdo, string $nuova_nascita, string $vecchio_id){
    set_data($pdo, $nuova_nascita, $vecchio_id);
}
}

```

Le query che si occupano di interrogare il database e inserire i dati al suo interno in assenza di errori sono, come di consueto, scritte nel file ‘_model.inc.php’.

```
<?php
declare (strict_types=1);
function get_utenti(object $pdo){
    $query = "SELECT * FROM anagrafica WHERE id_utenti = :id_utenti";
    $stmt = $pdo -> prepare($query);
    $stmt->bindParam(":id_utenti", $_SESSION["user_id"]);
    $stmt->execute();
    $result = $stmt -> fetchAll(PDO::FETCH_ASSOC);
    return $result;
}

function set_data(object $pdo, string $nuova_nascita, string $vecchio_id) {
    $query = "UPDATE anagrafica SET data_di_nascita = :nuova_nascita WHERE
(id_utenti = :id_utenti AND id_paziente = :vecchio_id)" ;
    $stmt = $pdo -> prepare($query);
    $stmt->bindParam(':nuova_nascita', $nuova_nascita);
    $stmt->bindParam(':id_utenti', $_SESSION["user_id"]);
    $stmt->bindParam(':vecchio_id', $vecchio_id);
    $stmt->execute();
}
```

Le righe di codice viste sono quasi identiche per tutti i campi modificabili. Cambiano solamente i nomi delle variabili e alcuni campi nelle query di inserimento.

Per quanto riguarda i dati modificabili relativi all’utente stesso, il codice che permette la modifica varia da quello sopra scritto solamente nella form HTML. Ovviamente in quel caso non è presente un input che richiede quale identificativo si vuole modificare, tantomeno la query annessa.

È possibile per l’utente modificare anche la password con cui si esegue l’accesso. Per questo campo viene però richiesta e controllata la vecchia password per motivi di sicurezza.

5. CONCLUSIONI

In questo progetto di tesi si è discusso del tumore alla cervice uterina e delle diverse problematiche che emergono al momento della diagnosi, con particolare riferimento alla soggettività nell'interpretazione delle immagini colposcopiche. Questa soggettività è influenzata da vari fattori, tra cui l'esperienza del medico, la sua manualità e il livello di allarmismo che presenta nell'analisi delle immagini.

Per ovviare a queste difficoltà, si potrebbe ricorrere all'utilizzo di sistemi di supporto decisionale avanzati, come reti neurali artificiali appositamente addestrate a riconoscere specifiche caratteristiche all'interno delle immagini colposcopiche. Tuttavia, un passo cruciale in questo ambito è la raccolta e la classificazione accurata dei dati di addestramento, che spesso risultano carenti in ambito medico, in particolare per quanto riguarda la colposcopia.

Diverse ricerche hanno evidenziato come la raccolta di immagini in campo medico, al fine di sviluppare algoritmi di intelligenza artificiale, sia di grande interesse anche per la comunità internazionale nella lotta contro il cancro alla pelle. In questo contesto, numerosi studiosi di diverse nazionalità hanno collaborato per creare un sito internet denominato ISIC: International Skin Imaging Collaboration. Questa piattaforma rappresenta un'importante risorsa globale, avendo raccolto finora più di un milione di immagini di tumori cutanei provenienti da tutto il mondo. La ISIC sta fornendo un contributo significativo alla ricerca scientifica, consentendo lo sviluppo di strumenti diagnostici avanzati basati sull'intelligenza artificiale.

Per le motivazioni appena descritte, in questo progetto è stato sviluppato un database per la raccolta e la classificazione del maggior numero possibile di immagini colposcopiche, provenienti da numerose strutture ospedaliere italiane. Per rendere l'inserimento delle immagini semplice e intuitivo, e ridurre la possibilità che il procedimento diventi noioso o troppo complesso, è stato creato un sito web apposito. Questo portale ha l'obiettivo di facilitare l'utente nell'upload delle immagini offrendo un'interfaccia intuitiva.

Durante lo sviluppo sono emerse diverse difficoltà, che tuttavia sono state affrontate e superate con successo. Al momento, il sito non è stato reso disponibile e pubblicato nel web, poiché è ancora in fase di perfezionamento.

Con il contributo di specialisti in ambito ginecologico e ostetrico, si sta lavorando per migliorare ulteriormente la piattaforma, valutando quali ulteriori informazioni potrebbero essere richieste agli utenti durante l'inserimento delle immagini, al fine di categorizzarle nel modo più accurato possibile.

Una volta completata questa fase di sviluppo, il sito sarà reso disponibile, nella speranza di incentivare la partecipazione e la collaborazione di un numero sempre maggiore di ospedali, contribuendo così in modo significativo alla ricerca medica in questo campo.

BIBLIOGRAFIA

- Ahmadzadeh Sarhangi, H., Beigifard, D., Farmani, E., & Bolhasani, H. (2024). Deep learning techniques for cervical cancer diagnosis based on pathology and colposcopy images. *Informatics in Medicine Unlocked*, 47, 101503.
- Bermudez, A., Bhatla, N., & Leung, E. (2015). Cancer of the cervix uteri. *International Journal of Gynecology & Obstetrics*, 131, S88–S95.
- Boon, S. S., Luk, H. Y., Xiao, C., Chen, Z., & Chan, P. K. S. (2022). Review of the Standard and Advanced Screening, Staging Systems and Treatment Modalities for Cervical Cancer. *Cancers*, 14(12).
- Dong, B., Xue, H., Li, Y., Li, P., Chen, J., Zhang, T., Chen, L., Pan, D., Liu, P., & Sun, P. (2022). Classification and diagnosis of cervical lesions based on colposcopy images using deep fully convolutional networks: A man-machine comparison cohort study. *Fundamental Research*.
- Ellis, H. (2011). Anatomy of the uterus. *Anaesthesia & Intensive Care Medicine*, 12(3), 99–101.
- Endale, H. T., Mariye, Y. F., Negash, H. K., Hassen, F. S., Asrat, W. B., Mengstie, T. A., & Tesfaye, W. (2024). MiRNA in cervical cancer: Diagnosis to therapy: Systematic review. *Heliyon*, 10(3), e24398.
- Hilal, Z., Tempfer, C. B., Burgard, L., Rehman, S., & Rezniczek, G. A. (2020). How long is too long? Application of acetic acid during colposcopy: a prospective study. *American Journal of Obstetrics and Gynecology*, 223(1), 101.e1-101.e8.
- Huang, S., Yang, J., Fong, S., & Zhao, Q. (2020). Artificial intelligence in cancer diagnosis and prognosis: Opportunities and challenges. *Cancer Letters*, 471, 61–71.
- Newaz, A., Muhtadi, S., & Haq, F. S. (2022). An intelligent decision support system for the accurate diagnosis of cervical cancer. *Knowledge-Based Systems*, 245, 108634.
- Prendiville, W., Sankaranarayanan, R., & World Health Organisation. (2017). Colposcopy and Treatment of Cervical Precancer. *IARC Technical Publication No.45*, 1–172.
- Reich, O., & Regauer, S. (2023). Elimination of reserve cells for prevention of HPV-associated cervical cancer. *Virus Research*, 329, 199068.

Sambyal, D., & Sarwar, A. (2023). Recent developments in cervical cancer diagnosis using deep learning on whole slide images: An Overview of models, techniques, challenges and future directions. *Micron*, *173*, 103520.

Shafi, Mahmood., & Nazeer, Saloney. (2012). *Colposcopy : a Practical Guide*. 94.

Shen, D., Wu, G., & Suk, H. Il. (2017). Deep Learning in Medical Image Analysis. *Annual Review of Biomedical Engineering*, *19*, 221.