



**ArtReader**

Daniele Genta

Davide Massimino

INDICE

1. Introduzione
   1. Premessa
   2. Omogeneità: logo, font e colori
2. Architettura del progetto
   1. Schema architettura
   2. Risorse e piattaforme utilizzate
3. Lato server
   1. Idea e funzionamento
4. Database
   1. Idea e funzionamento
5. Android
   1. Idea e premessa
   2. Login
   3. Homepage
   4. Scan
   5. Informazioni singolo artwork
   6. Vicino a te
   7. Gestione cronologia e preferiti
   8. Gestione recensioni
   9. Gestione mappe
   10. Altre funzioni
6. Web
   1. Premessa e funzionamento
   2. Homepage
   3. Login
   4. Aiuto
   5. Index
   6. Singolo artwork
   7. Singolo autore
   8. Singolo museo
   9. Nota su upload
   10. Altre funzioni

1 - Introduzione

**PREMESSA**

L’Idea

Con l’arrivo del secondo quadrimestre abbiamo iniziato a pensare ad i possibili progetti per l’esame di maturità, essendo parte di un gruppo ci siamo confrontati ed abbiamo attentamente esaminato i pro ed i contro di ogni idea che uno di noi avesse proposto, la scelta è ricaduta su una piattaforma formata essenzialmente da un sito *web* ed una applicazione per *Smartphone*.

Alla fine abbiamo scelto di gestire il concetto di musei ed opere d’arte sia in funzione del loro grande potenziale, siamo riusciti infatti ad inserire diverse funzioni peculiari ed interessanti, sia per ragioni più pratiche: secondo noi mancava un motore ed una piattaforma che legasse in maniera semplice diverse realtà artistiche dislocate in tutto il pianeta.

Il Nome

Abbiamo posto particolare enfasi anche su aspetti come la nomenclatura del nostro progetto.

Dopo varie ipotesi ha prevalso il nome che tuttora campeggia sulla nostra piattaforma: ‘Art Reader’.

Il suddetto nome si pone come obiettivo la coniugazione del tema fondamentale affrontato nella notraapp: l’arte e la funzione madre del nostro progetto: la lettura e la presentazione di informazioni.

Abbiamo inoltre inserito in più parti del progetto uno slogan: ‘LearnFaster, Learn More’, che vuole essere un’estrema esemplificazione e riassunto dell’intero progetto, imparare in maniera più veloce ed interattiva.

Cronistoria

Come tutte le idee, anche questa denominata ‘ArtReader’ ha subito notevoli progressi e radicali cambiamenti nel corso della sua evoluzione.

Dopo aver individuato lo scopo del nostro progetto d’esame abbiamo compiuto una fase di brainstorming volto a definire le prime funzioni.

Inizialmente il sito web doveva essere soltanto un appoggio per l’app, un back-end che svolgesse funzioni primitive quali la gestione degli *artwork*e la loro semplice visualizzazione.

Con il passare del tempo ci siamo accorti che volevamo fare qualcosa di più grande, più complesso: il nostro back-end doveva essere un qualcosa di a sé stante, pienamente godibile e completo sotto ogni aspetto.

Per l’App invece abbiamo inizialmente pensato e tuttora pensiamo a qualcosa di semplice ma efficace, una piattaforma snella ed intuitiva che permetta in maniera accattivante la presentazione di informazioni e *media (foto, file audio)*.

Come abbiamo lavorato

La nostra piattaforma è frutto di un lavoro svolto ambo a casa ed a scuola nelle ore dedicate, armati di strumenti di condivisione quali *GitHub*e simil, abbiamo infatti avuto modo di lavorare comunamente al progetto mediante un continuo flusso di ricerca, sperimentazione ed arricchimento funzionale e grafico.

Fonti

Le fonti utilizzate per l’inserimento dei dati sono aggiornate ed autorevoli, ci siamo impegnati nel ricercare le informazioni a noi utili su più siti web, tra cui enciclopedie online e siti ufficiali dei musei interessati.

**Omogeneità: Logo, font e colori**

Fin da subito abbiamo voluto imprimere ad *ArtReader*un’impronta fortemente personalizzata: il logo che abbiamo disegnato e che appare nel front-end web e nell’applicazione ne è l’esempio più evidente.

Come logo abbiamo infatti scelto un qualcosa di astratto, slegato dal progetto in sé, questo perché per noi l’arte è anche questo: astrattezza.

Il logo vuole simboleggiare la leggerezza e la creatività che si avvertono quando si entra in contatto con il mondo artistico, riprende inoltre il colore primario del progetto.

Per quanto riguarda i colori ne abbiamo scelti due rappresentativi e li abbiamo utilizzati in modo da garantire un’omogeneità ed un family-feeling in qualunque parte della piattaforma a contatto con l’utente.

Stesso discorso per il font, scelto in maniera precisa al fine di risaltare ed aver sempre la massima leggibilità, in tutte le parti della piattaforma.



2 –Architettura

**Schema**

DATABASE

SERVER

EXPRESS

Back-end

APPLICAZIONE

WEB

APPLICAZIONE

ANDROID

Front-end

Come sopra riportato l’architettura del nostro progetto si compone essenzialmente di:

* Database SqlLite, per l’immagazzinazione di dati
* Server Express, espone le risorse e le API che interfacciano le applicazioni client con il database
* Applicazione Web, front-end lato client per l’interazione con le informazioni via browser
* Applicazione Android, front-end lato client per l’interazione via smartphone

**Risorse e piattaforme utilizzate**

* Lato server
  + Express/NodeJs, abbiamo scelto di usare il framework Express in quanto fornisce una gestione integrata e semplificata delle sessioni e non necessita di un dispatcher a parte.
* Database
  + SqlLite, scelto per semplicità d’uso e potenza. Esso permette la fruizione dei dati anche in maniera grafica interattiva tramite *SqlLite Browser*.
* Smartphone
  + Android, la scelta di sviluppare un’appAndroid ha una duplice connotazione: è il linguaggio che abbiamo utilizzato durante il percorso scolastico ed è fruibile per una gamma pressochè infinita di dispositivi.
* Front-end web
  + HTML5
  + jQuery – jQueryUi
  + CSS3

3 - Lato Server

Idea

Il server presenta una serie di API utilizzate in gran parte sia dalla controparte web che da quella Android.

Le API sono per la maggiorparte di tipo asincrono e restituiscono risorse ed informazioni mediante lo standard JSON, il Server attraverso le sue API è l’unica parte della piattaforma a diretto contatto con il database.

Come funziona

Il server Express viene utilizzato per la comunicazione tra il front-end ed il database, espone API per la gestione del login e la presentazione delle informazioni. Alcune API sono specificamente create per un utilizzo WEB altre vengono utilizzate dalla controparte Smartphone, tuttavia la maggioranza delle API sono sfruttate da ambedue gli ambienti al fine di garantire un’omogeneità e semplicità di gestione.

Nota sulla gestione dei dati sensibili

I dati sensibili, come la gestione e la registrazione della password, sono salvati usando la crittografia. Mediante il modulo Node ‘*Crypto’* si procede alla segretezza dei suddetti dati che vengono crittografati con una chiave a 256bit.

Nota sulla gestione delle sessioni

Quando si effettua l’accesso vengono sfruttate le sessioni offerte da *express*, salvando una traccia dell’utente avente effettuato il login.

Le sessioni vengono inoltre utilizzate per il salvataggio di dati al cambiare della pagina web (quando si visualizzano i dati di un singolo artwork, autore e museo).

4 - Database

Premessa

Abbiamo scelto SqlLite perché permette una gestione leggera ed efficace dei dati, inoltre presenta un’apertura a diversi ambienti come Javascript ma anche Java per Android che ci ha permesso di sfruttarne a pieno le potenzialità.

Come funziona

Il database è formato essenzialmente dalle seguenti tabelle:

* Artwork
* Author
* LocationArtwork
* Feedback
* Users

Il loro legamepuò essere rappresentato con il sottostante schema:

LOCATION

ARTWORKS

AUTHORS

USERS

ARTWORKS

FEEDBACK

5 -Android



Premessa

Abbiamo scelto di sviluppare un AppAndroid per una serie di ragioni tra cui la familiarità acquisita in mesi di lavoro scolastico e la versatilità della piattaforma.

Come funziona

All’avvio dell’app è richiesto di effettuare il Login che andrà a salvare mediante *SharedPreferences*l’identificativo di chi ha effettuato l’accesso.

Il login può essere saltato con conseguente perdita della possibilità di inserire recensioni per gli *artwork*.

L’homepage è divisa in tre sezioni, *fragment*, scorribili mediante uno swipe sullo schermo verso destra o sinistra.

Se si sceglie di effettuare una scansione di un codice QR, questo viene scannerizzato all’interno dell’app, attraverso la fotocamera e viene presentata una schermata con la quale l’utente può interagire.

**Login**



Funzionamento

Il login rappresenta l’entry point dell’applicazione, la prima schermata con cui l’utente interagisce.

Abbiamo pensato ad un login non vincolante, esso può essere saltato ed eseguito in seguito.

Non eseguire il login preclude la possibilità di inserire una nuova recensione ma non quella di visualizzare quelle già inserite o effettuare una qualunque altra operazione gestita dall’applicazione.

**Homepage**



Funzionamento

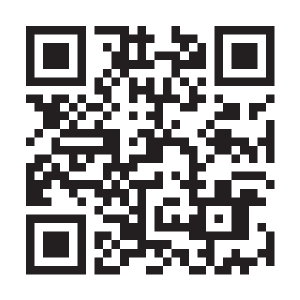
Effettuato il login si accede alla pagina principale suddivisa in due fragment, scrollabili con semplici swipe sullo schermo.

* Gestione dei preferiti, vengono visualizzati in una lista tutti gli artwork che l’utente ha selezionato come preferiti durante la sua navigazione.
* Pagina principale, abbiamo pensato ad una pagina iniziale ricca di contenuti e dinamica, si suddivide in tre parti
  + Visualizzati di recente, analizzando la cronologia vengono presentati in ordine cronologico decrescente gli ultimi artwork visualizzati dall’utente.
  + Consigli per te, mediante un algoritmo viene calcolato l’autore più visualizzato e preferito dell’utente, vengono poi estratti i suoi quadri e presentati i tre più famosi.
  + Le nostre opere, con una tabella dinamica con tutte le opere presenti nel database, offrendo una panoramica di contenuti scrollabile dall’utente.

In ambo le pagine sono presenti

* Bottone per la scansione, permette di avviare l’activity che si occupa della scansione del codice Qr.
* Top-Menù, permette di accedere ad una serie di funzionalità:
  + Vicino a te
  + Cronologia
  + Cancella cronologia
  + Sessione privata, permette una sessione senza il salvataggio della cronologia

**Scan**



Premendo il bottone adibito alla scansione nella homepage, viene aperta una activity che sfruttando la fotocamera è in grade di leggere un codice QR, ciò lo abbiamo fatto appoggiandoci alla libreria dedicata *ZXING*.

Se la lettura del codice Qr avviene correttamente il risultato ne viene analizzato, se è del tipo *IdArtworkArtreader*, mediante api dedicate vengono presentate le informazioni relative ad un certo *artwork*.

**Informazioni del singolo Artwork**



In seguito alla scansione le informazioni dell’*artwork* sono presentate in un’*activity* dedicata.

Questa si compone di un *header* che presenta la foto del museo nel quale l’opera ha luogo ed il bottone per la gestione dei preferiti.

Sotto l’header è presenta una griglia formata da tre bottoni:

* Mappa, apre l’activity con la visualizzazione di una mappa che segnala l’indirizzo del museo nel quale l’opera ha luogo.
* Recensioni, apre l’activity per la visualizzazione e gestione delle recensioni della singola opera
* Audio, mette in play la audioguida dell’opera selezionata

Nella parte centrale vengono presentate le informazioni dell’*artwork* e la relativa foto.

Infine sono disponibili le ricerche correlate le quali sfruttando le stesse api della controparte web estraggono opere aventi autore e movimento artistico simili all’opera selezionata.

**Vicino a te**



La pagina esplora si pone come obbiettivo di visualizzare utilizzando la geolocalizzazione, i musei più vicini all’utente. Questi vengono presentati con una fotografia, tre bottoni posizionati sopra di essa che gestiscono i contatti:

* Telefono
* Sito web
* Mappa

Vengono inoltre presentati per ogni museo una gallerie delle opere d’arte più famose in esso presenti.

Algoritmi per la gestione della vicinanza:

Per il calcolo della distanza utilizziamo una funzione di Google, che inserendo le coordinate geografiche di due punti A e B ne calcola la distanza in linea d’aria in metri.

Per individuare la posizione attuale sfruttiamo il GPS del dispositivo.

**Gestione cronologia e preferiti**



Abbiamo deciso di gestire in locale la cronologia ed i preferiti, in maniera autonoma dall’utente, questo ci permette di rendere opzionale il login e di non dipendere dalla connessione ad internet.

Cronologia e preferiti sono salvati attraverso file di testo che vengono letti ed utilizzati per popolare liste dedicate utilizzate per la gestione.

La cronologia viene aggiornata ogni qual volta si visualizzano le informazioni di un certo *artwork*, mentre i preferiti in seguito al premere del bottone dedicato.

**Gestione recensioni**



Le recensioni si compongono essenzialmente di due poli

* Lista per la visualizzazione, visualizza per l’artwork corrente tutte le recensioni lasciate dagli utenti, mediante il click su di una delle recensioni è possibile leggerne il testo completo.
* Pagina per l’inserimento di una nuova recensione, raggiungibile solamente in seguito al login, permette appunto l’aggiunta da parte di un utente loggato di una nuova recensione rispetto all’*artwork*correntemente visualizzato.

**Gestione mappe**



Le mappe sono rappresentate da una Activity dedicata messa a disposizione nelle ultime release di Android, esse presentato una mappa attraverso le API di *Google Maps*, con un marker impostato utilizzando latitudine e longitudine.

Nota conversione da indirizzo a latlng:

per convertire l’indirizzo del museo in coordinate geografiche accettate da googlemaps utilizziamo la seguente funzione

public LatLnggetLocationFromAddress(Contextcontext,StringstrAddress) {

Geocodercoder = new Geocoder(context);

List<Address>address;

LatLng p1 = null;

try {

address = coder.getFromLocationName(strAddress, 5);

if (address == null) {

returnnull;

}

Address location = address.get(0);

location.getLatitude();

location.getLongitude();

p1 = new LatLng(location.getLatitude(), location.getLongitude() );

} catch (Exception ex) {

ex.printStackTrace();

}

return p1;

}

**Altre funzioni**

Gestione della sessione privata

Nell’activity principale, mediante il menù, è possibile attivare la modalità ‘sessione privata’ che gestisce la navigazione anonima ovvero la possibilità di non registrare la cronologia dell’utente durante la fruizione della piattaforma.

Gestione dei file audio

Per realizzare l’audioguida dei singoli artwork abbiamo sfruttato la cartella *raw* messa a disposizione da Android, questa ci permette di immagazzinare e di riprodurre (mediante apposite classi) il file audio che viene registrato con la nomenclatura: *titoloartwork.mp3.*

Cancellazione cronologia

Nell’activity principale, mediante il menù, è possibile cancellare la cronologia corrente, questa azione è irreversibile e permette di ‘svuotare’ la lista della cronologia ed il relativo file di testo memorizzato sul dispositivo.

Gradient

Nell’activity vicino a te sovrapponiamo il testo alle immagini, per rendere il testo sempre leggibile, anche se fosse dello stesso colore dello sfondo applichiamo uno strato aggiuntivo che sfuma dal nero al trasparente e rende le scritte sempre leggibili.

6 - Web

Premessa

Il sito web è stato concepito come una piattaforma che permetta a tutti gli utenti di poter visualizzare le informazioni delle diverse opere e trovare le specifiche informazioni degli autori e dei musei.

Contemporaneamente il sito permette agli amministratori di andare a modificare, eliminare e aggiungere nuove opere d'arte.

Come funziona

Dalla pagina home oltre ad effettuare il login o registrarsi se non si ha ancora un account, sarà possibile accedere ad informazioni generali sull' utilità e sullo scopo del sito permettendo a chiunque di inviare un feedback oppure chiedere informazioni.

Effettuato il login ci verranno mostrate tramite una nuova pagina le diverse opere d'arte, i musei e gli autori.

A seconda del tipo di utente sarà possibile attuare modifiche o aggiunte, altrimenti sarà possibile solo visualizzare le informazioni dettagliate tramite un' altra pagina.

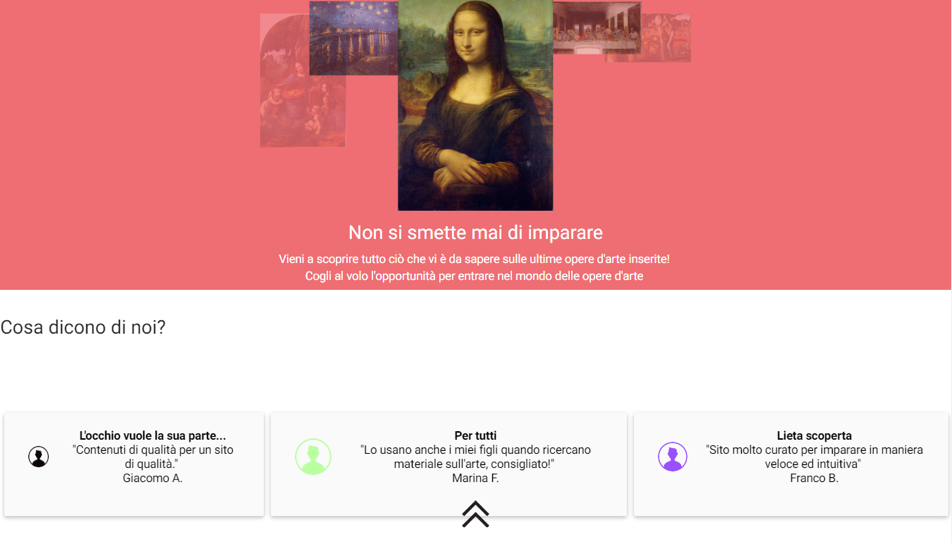
Cross-Browsing

Il sito web è stato testato sui più recenti browser tra cui: Chrome, Firefox, Safari, Opera e Microsoft Edge. Il Css e tutte le funzioni sono adattate mediante appositi accorgimenti ad i suddetti ambienti.

Responsive

Il sito web è inoltre fruibile in tutte le sue funzioni anche da Smartphone e Tablet dove avviene una ri-disposizione automatica degli oggetti sulla pagina al fine di un’esperienza pienamente godibile.

**Homepage**

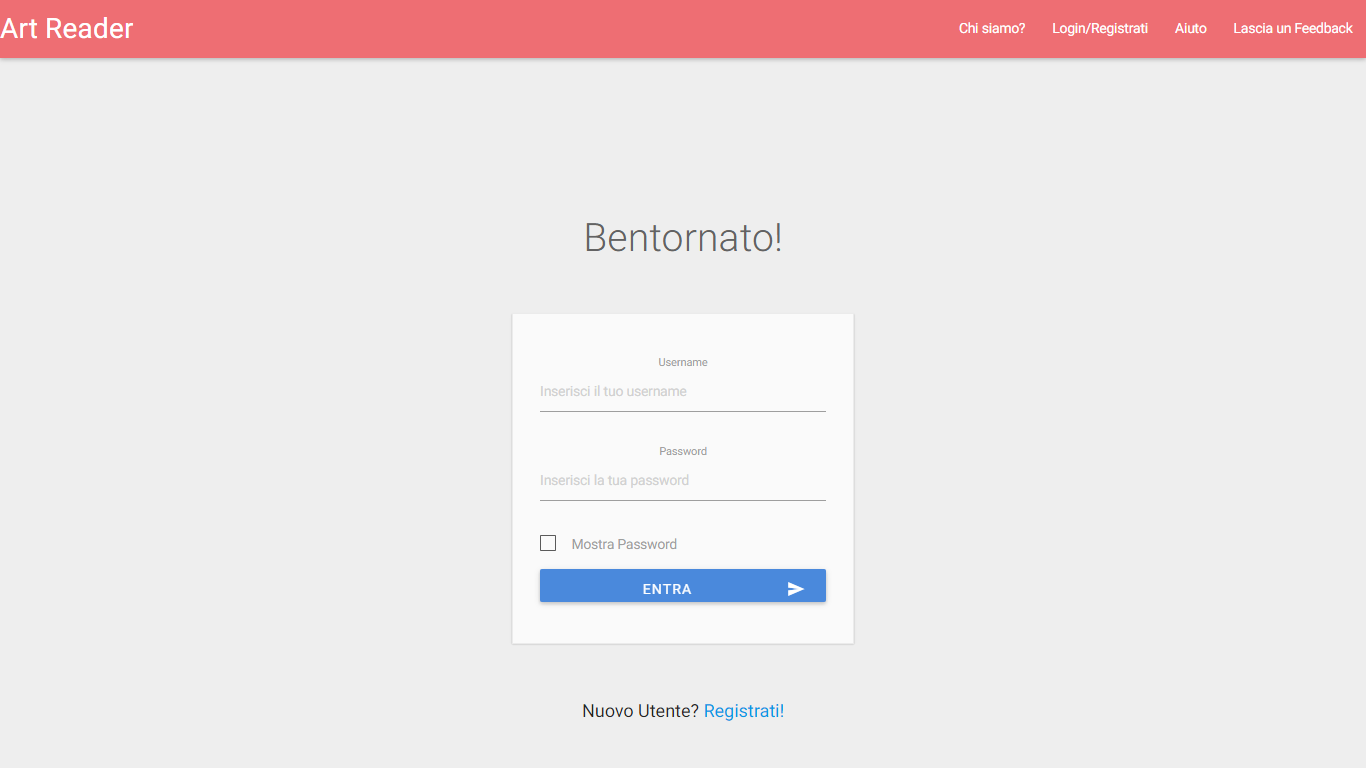


Pagina principale contenente informazioni sul sito e sulla sua creazione, informa gli utenti sulla presenza dell'applicazione per *Smartphone* e sulla possibilità di lasciare *feedback*.

premendo il bottone in fondo alla pagina vengono mostrati i musei nella quale il progetto è attivo mediante una mappa che viene caricata tramite la richiesta di una lista di tutti i musei fatta al caricamento della pagina al server, le opere più famose trattate nel sito tramite un image-slider che viene caricato tramite la richiesta delle opere che hanno avuto più visualizzazioni e gli ultimi feedback mostrati singolarmente via blocchi.

Dal menù superiore si possono raggiungere le altre pagine visualizzabili senza dover effettuare il login tra cui la pagina per l'invio di un feedback, la pagina per effettuare il login e una pagina per la richiesta di informazioni e aiuto.

**Login**

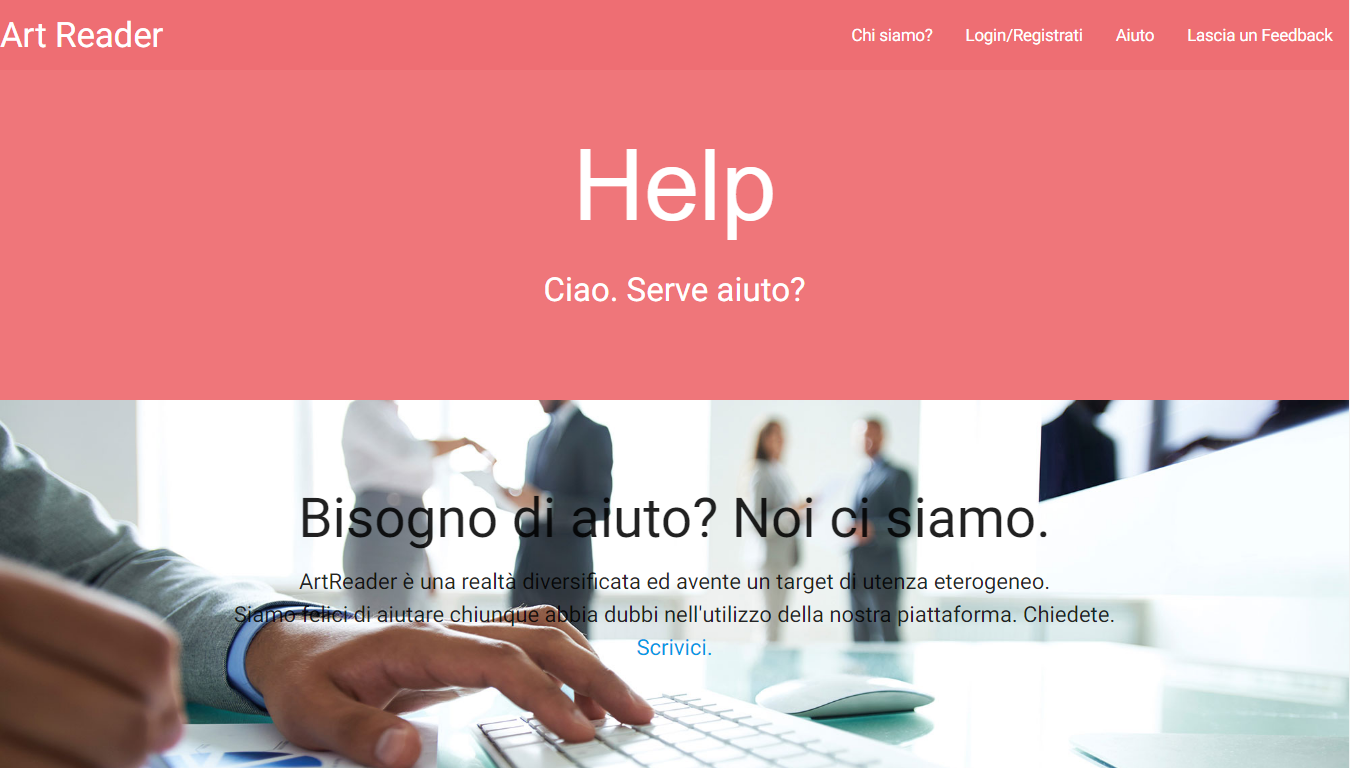


La pagina si presenta in modo molto semplice mostrando solo la form per l'inserimento delle credenziali, premuto il pulsante per il login dopo il controllo sui campi per verificare che questi non siano vuoti verrà richiamata l'API del server che gestisce la certificazione delle credenziali inserite, se all' interno del database non sarà presente una corrispondenza verrà restituita la home-page altrimenti sarà restituita la pagina contenente le liste delle opere, degli autori e dei musei.

Andando a premere la scritta sottostante la form di login si potrà effettuare una nuova registrazione, Infatti ci apparirà una nuova form che richiederà tutte le informazioni necessarie per l'iscrizione.

Prima dell' aggiunta del nuovo utente tutti i dati verranno controllati anche tramite l'utilizzo di regex per quanto riguarda il campo mail.

**Aiuto**

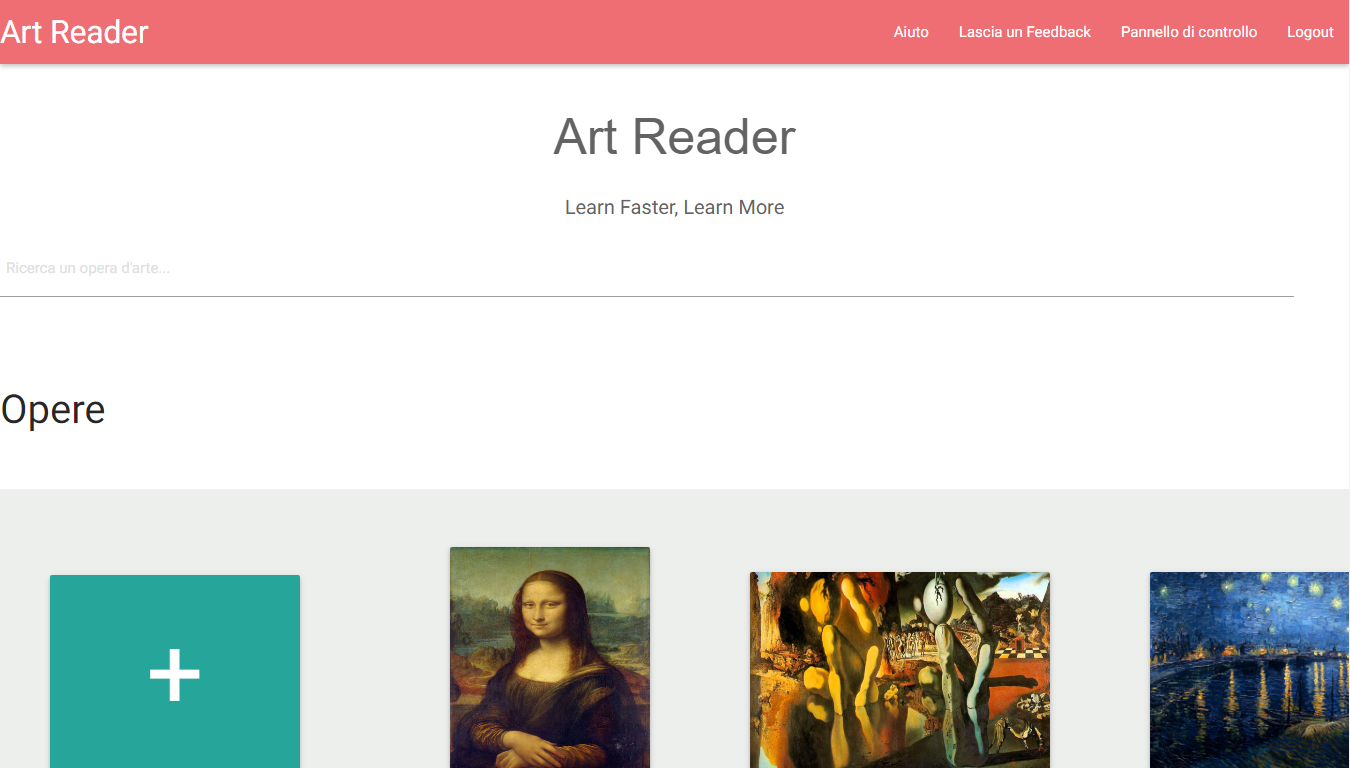


Questa pagina permette ai possibili visitatori di richiedere informazioni agi amministratori del sito.

Tramite una scritta sarà possibile richiamare un client di posta che indirizzerà la mail da inviare direttamente all' amministratore con oggetto personalizzato ed una priorità di risposta alta.

**Dopo il login**

**Index**

****

Questa pagina permette agli utenti registrati di raggiungere le informazioni specifiche di ogni opera, autore e museo trattato dal sito.

La pagina è suddivisa in tre liste che mostrano tramite blocchi una foto dell' oggetto specifico che trattano(opera, autore, museo) e vengono caricate tramite il richiamo di tre API diverse che restituiscono il JSON contenente le informazioni necessarie per ognuna delle diverse liste .

Andando a cliccare su una di queste immagini comparirà al posto della foto una lista delle possibili azioni da svolgere rappresentate da diversi bottoni che varieranno a seconda del tipo di utente, mentre per la normale utenza sarà possibile richiamare la pagina con le informazioni aggiuntive, l'amministratore potrà anche modificare o eliminare quel determinato oggetto dalla lista.

Il pulsante per l'aggiunta di una nuova opera d'arte sarà visibile solo all' amministratore e verrà visualizzato come il primo blocco della lista di opere d'arte.

Sarà possibile effettuare ricerche sulle opere d'arte inserendo il nome dell'opera nel *textbox* superiore alla lista, la ricerca verrà effettuata ogni volta che si aggiunge o si elimina un carattere dalla stringa, di conseguenza verrà mostrato come risultato l'insieme dei quadri il cui nome inizia con la stringa di ricerca inserita.

API per popolare lista opere d'arte:

functionreadUpdate()

{

if (readRequest.readyState == 4 &&readRequest.status == 200)

{

varresponse = JSON.parse(readRequest.responseText);

varartworks = response;

var i = 0;

cleanTable();

for (varcounter in artworks)

{

showArtwork(artworks, i);

i++;

}

}

}

la funzione *showArtwork* viene richiamata per ogni oggetto del *json* e sarà questo a creare la struttura a blocchi di ogni opera e ad aggiungerla alla lista.

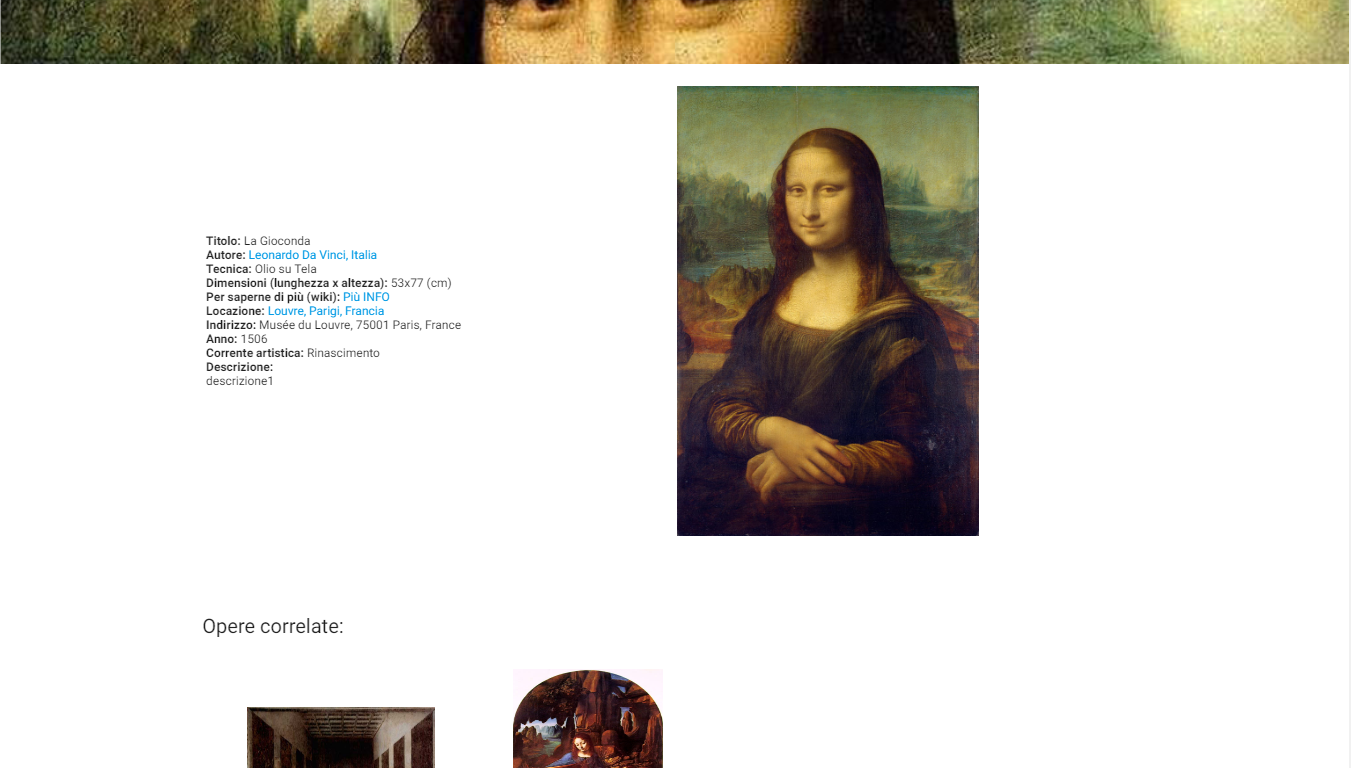
**Singolo artwork**

La pagina viene utilizzata per mostrare le informazioni dettagliate delle singole opere.

Nella parte superiore ritroviamo la foto dell'opera d'arte che scorre seguendo lo scorrimento del mouse nella pagina, in egual modo in fondo alla pagina ritroviamo la foto del museo nella quale si trova il quadro, questo effetto viene gestito tramite l'utilizzo di una libreria esterna di nome *Parallax*.

Nel centro pagina ritroviamo una versione completa del quadro con a fianco tutte le due informazioni, di cui alcune come l'autore sono collegate a pagine *Wikipedia*.

Inseguito viene mostrata una lista di opere che hanno in comune con quella ricercata l'autore e/o il movimento artistico, in questo modo cliccando su uno di questi quadri verremo riportati subito alla loro corrispondente pagina di informazioni.



**Singolo autore**

La pagina viene utilizzata per mostrare le informazioni dettagliate del singolo autore.

Nel centro pagina ritroviamo una versione completa dell'autore con a fianco tutte le due informazioni, di cui alcune come l'autore sono collegate a pagine *Wikipedia*.

Sono inoltre presenti due gallerie:

* Autori correlati, presenta gli autori della medesima nazionalità
* Opere dell’autore, presenta una panoramica delle opere più famose dell’autore scelto.

Gestione JSON contenente informazioni su singolo autore:

functionsingleAuthorUpdate()

{

if (readRequest.readyState == 4 &&readRequest.status == 200)

{

varresponse = JSON.parse(readRequest.responseText);

varauthor = response;

printAuthorDetails(author);

}

}

Visualizzazione dati del singolo autore:

functionprintAuthorDetails(author)

{

$("#lblAuthorName, #infoHeader").text(author.name);

$("#lblAuthorDateBorn").text(author.dateBorn);

$("#lblAuthorLocationBorn").text(author.locationBorn);

$("#lblAuthorNationality").text(author.nationalityAuthor);

$("#lblWikipediaPageAuthor").attr("href", author.wikipediaPageAuthor);

$("#imgAuthor").attr("src","img/immagini/autori/"+author.pictureUrlAuthor);

//relatedartworks

relatedAuthors\_SinglePageAuthor(author.nationalityAuthor);

authorArtworks(author.id);

}

**Singolo museo**

La pagina viene utilizzata per mostrare le informazioni dettagliate del singolo museo.

Nel centro pagina ritroviamo una versione completa del museo con a fianco tutte le due informazioni, di cui alcune come l'autore sono collegate a pagine *Wikipedia*.

Sono inoltre presenti una galleria degli *artwork* più famosi all’interno del museo ed una mappa che permette la localizzazione di esso.

Gestione JSON contenente informazioni su singolo museo:

functionsingleLocationUpdate()

{

if (readRequest.readyState == 4 &&readRequest.status == 200)

{

varresponse = JSON.parse(readRequest.responseText);

var location = response;

printLocationDetails(location);

}

}

Visualizzazione dati del singolo museo:

functionprintLocationDetails(location)

{

$("#lblLocationDescription, #infoHeader").text(location.description);

$("#lblLocationCity").text(location.city);

$("#lblLocationNation").text(location.nation);

$("#lblWikipediaPageLocation").attr("href", location.wikipediaPageLocation);

$("#lblLocationAddress").text(location.address);

$("#lblLocationPhone").text(location.telephone);

$("#lblLocationWebsite").text(location.website);

$("#lblLocationWebsite").attr("href", location.website);

$("#imgLocation").attr("src", "img/parallax/"+location.pictureUrlMuseum);

mostFamousArtworks(location.id)

}

**Feedback**

Questa pagina permette a chiunque abbia visitato il sito di poter lasciare un commento su cosa ne pensa e su quanto l'ha trovato utile.

La pagina è composta da una *form* che richiede l'inserimento delle informazioni minime per poter scrivere un feedback.

Tramite un pulsante il feedback verrà aggiunto al database in modo cronologico in modo da poter avere a portata di mano sempre gli ultimi commenti lasciati

API per inserimento feedback:

app.get("/insertFeedback", function(request, response,next) {

vartype=request.query["Type"];

varartwork=request.query["idArtwork"];

varapproved=request.query["Approved"];

vardescription=request.query["Description"];

var username=request.query["Username"];

varphonenumber=request.query["Phonenumber"];

var email=request.query["Email"];

vartitle=request.query["Title"];

vardateReview=request.query["DateReview"];

//sql

varheader = {'Content-Type' : 'text/plain',

'Cache-Control':'no-cache, must-revalidate'};

vardb = new sqlite.Database("Database/myDatabase.db");

db.serialize(function(){

//crea istruzione

varinsert=db.prepare("INSERT INTO Feedbacks(Type,Artwork,Approved,Description,Username,Phonenumber,Email, Title, DateReview) values(?,?,?,?,?,?,?,?,?)"); insert.run(type,artwork,approved,description,username,phonenumber,email, title, dateReview);

//esegui sql

insert.finalize();

response.writeHead(200,header);

db.close();

});

});

**Nota su upload**

Dalla pagina Index viene gestita, sia per l'aggiunta che per la modifica di opere d'arte, una *dialog form*.

questa a seconda dell'azione da svolgere andrà a caricare i campi dell' opera da modificare oppure li lascerà vuoti se sta compiendo un’aggiunta.

Dalla questa from verranno scelte le immagini del quadro e anche quelle del museo, inizialmente le foto verranno dal computer e verranno mostrate in *imageView*, ma solo in seguito tramite una specifica API inizierà il loro upload.

Il server al suo interno contiene una cartella nelle quali vengono salvate le foto dei quadri e un'altra nella quale vengono salvate le foto dei musei e degli autori.

API per l'upload delle immagini:

app.post('/api/photo',multipartMiddleware,function(request,response){

varserverPath = '/public/img/immagini/' + request.files.userPhoto.name;

require('fs').rename(

request.files.userPhoto.path,

'C:/Users/Hp Notebook/Desktop/QrReaderProject1104'+serverPath,

function(error){

if(error){

response.send({

error: 'errore upload'

});

return;

}

response.send({

path: serverPath

});

}

);

});

**Altre funzioni**

Nota sulla ricerca

La ricerca presente nella *home-screen* effettua alla pressione di ogni tasto al suo interno una serie di chiamate al server utili per ricercare tra *Artwork*, Autori e Musei.

Non è il caso di scrivere la stringa completa da ricercare bensì man mano che si scrive appaiono suggerimenti (simili a *googlenow*) per aiutare l’utente.

Vengono inoltre fornire una serie di ricerche correlate a ricerca conclusa, l’algoritmo esamina l’*artwork* scelto e ne analizza autore e movimento artistico, provvede poi con i suddetti criteri a ricercare *artwork* nel database.

Nota chi può inserire, modificare, cancellare

Con la gestione della session in Express, al momento del login viene estrapolato qualora l’utente sia amministratore o meno, in caso di utente normale non verranno presentate le possibilità di aggiunta, modifica e cancellazione degli artwork.

Algoritmo per la gestione delle immagini

In più punti del progetto viene sfruttato un algoritmo, *AsignImage*, al fine di ottenere immagini godibili e non snaturalizzate. L’algoritmo è molto semplice ed analizza larghezza e lunghezza di una tale immagine provvedendo ad allargare o allungare il *div* contenitore.