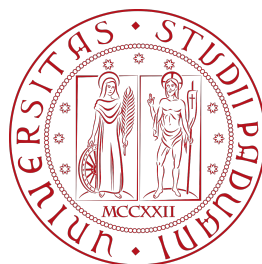


# elusive

Francesco Ceccato  
Nicola Ravagnan  
Sebastiano Sanson  
Matteo Stocco



Università degli Studi di Padova  
A.A. 2022/2023

## Informazioni sul sito

**Login consegna:** fceccato

**Indirizzo sito web:** <http://tecweb.studenti.math.unipd.it/fceccato/>

**Email referente gruppo:** francesco.ceccato.2@studenti.unipd.it

### Accessi interni al sito

Ruolo	Username	Password
Utente	user	user
Autore	writer	writer
Amministratore	admin	admin

# Indice

<b>1</b>	<b>Abstract</b>	<b>4</b>
<b>2</b>	<b>Analisi dei Requisiti</b>	<b>5</b>
2.1	Utenti . . . . .	5
2.1.1	Utente non autenticato . . . . .	5
2.1.2	Utente autenticato . . . . .	5
2.1.3	Utente autore . . . . .	5
2.1.4	Utente amministratore . . . . .	6
<b>3</b>	<b>Progettazione</b>	<b>7</b>
3.1	Struttura del progetto . . . . .	7
3.1.1	Struttura delle cartelle . . . . .	7
3.1.2	Approccio allo sviluppo . . . . .	7
3.2	Database . . . . .	7
<b>4</b>	<b>Implementazione Back-end</b>	<b>9</b>
4.1	PHP . . . . .	9
4.1.1	Pagine . . . . .	9
4.1.2	Interazione con il database . . . . .	10
4.1.3	Controllo input lato server . . . . .	10
4.2	Javascript . . . . .	11
4.2.1	Librerie utilizzate . . . . .	11
4.2.2	Immagini . . . . .	11
4.2.3	Voti . . . . .	11
4.2.4	Suggerimenti . . . . .	11
<b>5</b>	<b>Implementazione Front-end</b>	<b>12</b>
<b>6</b>	<b>Accessibilità</b>	<b>13</b>
6.1	Riferimenti normativi . . . . .	13
6.2	Strumenti utilizzati . . . . .	13
6.3	Verifica . . . . .	13
6.3.1	HTML . . . . .	13
6.3.2	CSS . . . . .	13

6.3.3	JavaScript . . . . .	14
6.3.4	Immagini . . . . .	14
6.3.5	Link . . . . .	14
6.3.6	Colori . . . . .	14
6.3.7	Form . . . . .	14
6.3.8	Tabelle . . . . .	15
6.3.9	Modalità di input . . . . .	15
6.3.10	Tastiera . . . . .	15
6.3.11	Ruoli, stati e proprietà WAI ARIA . . . . .	15
6.3.12	Aiuti alla navigazione . . . . .	15
6.3.13	Disorientamento . . . . .	16
6.3.14	Separazione tra struttura, contenuto e comportamento . . . . .	16
<b>7</b>	<b>Suddivisione del lavoro</b>	<b>17</b>
<b>8</b>	<b>Note finali</b>	<b>19</b>

## 1 Abstract

Il sito si propone come un aggregatore di notizie riguardanti il mondo della criptozoologia, pseudoscienza il cui ambito di interesse ricade su animali leggendari, la cui esistenza è controversa o non provata.

In particolare è possibile, per gli utenti appassionati, consultare articoli riguardanti le creature criptozoologiche, e i dettagli delle creature stesse, ed esprimere la propria opinione a riguardo.

Inoltre, gli utenti possono contribuire alla crescita del sito, aggiungendo nuove creature e creando nuovi articoli.

Sarà possibile, inoltre, visualizzare una classifica delle creature, ordinata secondo tre criteri differenti, in modo tale da poter avere una panoramica delle creature più quotate.

Classifica che verrà aggiornata in tempo reale, in base ai voti espressi dagli utenti.

## 2 Analisi dei Requisiti

Per il design del sito è stato necessario studiare il problema e definire le funzionalità opportune.

Di seguito verranno elencati le tipologie di utenti a cui è rivolto con le relative funzionalità, si specifica che tali utenti verranno descritti rispettando l'ordine gerarchico dei permessi a loro assegnati, quelli con permessi più elevati ereditano le funzionalità degli utenti con permessi inferiori.

### 2.1 Utenti

#### 2.1.1 Utente non autenticato

Questa tipologia di utente non possiede un account, e può usufruire di tutte le seguenti funzionalità di lettura:

- **Articoli:** possibilità consultare gli articoli presenti nel sito, il contenuto di ciascuno si può riferire ad una creatura in particolare;
- **Creature:** possibilità di visualizzare le caratteristiche delle creature presenti nel sito;
- **Lista degli articoli:** possibilità di effettuare una ricerca attraverso alcuni filtri;
- **Lista delle creature:** elencate in ordine alfabetico;
- **Classifica delle creature:** possibilità di visualizzare le creature ordinate secondo tre differenti criteri.

#### 2.1.2 Utente autenticato

Questa tipologia di utente possiede un account, e in aggiunta, può usufruire delle seguenti funzionalità:

- **Login e Logout:** possibilità di accedere al sito tramite un form di login, che permette di accedere alle funzionalità riservate agli utenti che possiedono un account. Inoltre, è possibile effettuare il logout, che termina la sessione corrente;
- **Votazione:** per ciascuna creatura è possibile esprimere un voto riguardante la sua esistenza, inoltre vi è la possibilità di retrocedere il voto precedentemente espresso, a causa di un errore di valutazione;
- **Commenti:** per ciascun articolo vi è la possibilità di commentare il suo contenuto.

#### 2.1.3 Utente autore

Questa tipologia di utente, in aggiunta può usufruire della seguente funzionalità:

- **Creazione di un articolo:** possibilità di creare un nuovo articolo, che può essere associato ad una creatura presente nel sito.

#### 2.1.4 Utente amministratore

Questa tipologia di utente, in aggiunta può usufruire delle seguenti funzionalità:

- **Aggiunta di una creatura:** possibilità di aggiungere una nuova creatura al sito;
- **Modifica di un articolo:** possibilità di modificare solamente il contenuto di un articolo;
- **Rimozione di un articolo:** possibilità di rimuovere un articolo;
- **Rimozione di una creatura:** possibilità di rimuovere una creatura;

## 3 Progettazione

### 3.1 Struttura del progetto

#### 3.1.1 Struttura delle cartelle

Il progetto segue la seguente sotto-struttura di cartelle:

- **html**: cartella contenente la struttura delle varie pagine del sito oltre che una cartella **modules** la quale contiene alcuni specifici moduli che vanno caricati all'interno di tutte le pagine, come header e footer, o solo in alcune specifiche, come i commenti.
- **css**: cartella contenente tutti i file che definiscono l'aspetto delle pagine del sito. In più ci sono i css per la visualizzazione mobile e per la stampa;
- **php**: questa cartella è suddivisa in due categorie. I file php che servono a rendere dinamiche le varie pagine del sito, e quelli che servono per l'esecuzione di query di scrittura nel DB;
- **js**: cartella contenente i file javascript, il quale ognuno si occupa di implementare una specifica funzionalità al sito;
- **images**: cartella contenente icone, loghi o sfondi presenti nel sito;
- **config.php**: file di configurazione dei vari percorsi utilizzati nel sito;
- **index.php**: file dell'homepage del sito.

#### 3.1.2 Approccio allo sviluppo

Per lo sviluppo del sito si sono infatti voluti separare al meglio struttura, aspetto e comportamento. Ogni pagina html ha infatti un suo corrispettivo foglio di stile e una pagina php associata, la quale serve ad importare la struttura della pagina e renderla dinamica. Eventualmente le pagine che fanno utilizzo di particolari funzionalità importano al loro interno i file javascript nelle quali sono scritte le funzioni che permettono di svolgere tali operazioni.

Vi sono inoltre altre pagine php, non visibili all'utente, che sono utilizzate per creare connessioni con il database e svolgere le query di scrittura su esso. Questo tipo di pagine servono anche per lo svolgimento dei controlli back-end di validazione dell'input.

Durante lo sviluppo

### 3.2 Database

Il database è composto da un totale di 6 entità:

- **Animale**: che contiene le informazioni di tutte le creature (scoperte, avvistate o ipotizzate) mostrate nel sito;
- **Articolo**: che contiene le informazioni di gli articoli pubblicati. Un articolo può riferire un animale ma non necessariamente;



- **Utente:** che contiene i dati dei vari utenti registrati al sito, compreso il loro ruolo;
- **Commento:** che contiene tutti i vari commenti presenti sotto agli articoli, con le informazioni sul dove si trovino e chi li abbia scritti;
- **Risposta:** Entità che modella la relazione tra commento padre e figli;
- **Voto:** Entità che modella i singoli voti degli utenti per ogni animale.

Sono inoltre state aggiunte delle view al database per semplificare alcune query che vengono spesso eseguite all'interno del sito:

- **View\_animale\_voto:** Che correla ogni animale alla quantità di voti ricevuti;
- **View\_articolo\_commento:** Che correla ogni articolo alla lista di commenti padre presenti in esso;
- **View\_articolo\_commento\_risposta:** Che correla ogni commento padre di uno specifico articolo ai propri figli;
- **View\_articolo\_utente:** Che correla ogni articolo al nome dell'utente che lo ha scritto;
- **Vote\_no:** Che raccoglie il numero di voti a sfavore dell'esistenza di un certo animale;
- **Vote\_yes:** Che raccoglie il numero di voti a favore dell'esistenza di un certo animale;

## 4 Implementazione Back-end

### 4.1 PHP

Nella cartella `/php` sono presenti tutti i file php utilizzati per la gestione del back-end del sito.

Il file `config.php` viene utilizzato dai file php per configurare i percorsi dei file necessari, all'interno di esso la variabile `$root_server_side` serve per impostare la cartella radice del sito.

#### 4.1.1 Pagine

Nella cartella `/php/pages` sono presenti tutti i file php che vengono richiamati dalle pagine del sito per ottenere i dati necessari alla loro visualizzazione:

- `article.php`: pagina dell'articolo, tramite richiesta GET viene passato l'id dell'articolo da visualizzare e il file si occupa di ottenere i dati dell'articolo dal database, i commenti associati ad esso e di visualizzarli nella pagina;
- `article-list.php`: pagina che visualizza tutti gli articoli, tramite richiesta GET si può specificare una stringa di ricerca e un tag utilizzati per filtrare i risultati, il file si occupa di ottenere gli articoli dal database e di visualizzarli nella pagina;
- `animal.php`: pagina della creatura, tramite richiesta GET viene passato il nome della creatura da visualizzare e il file si occupa di ottenere i dati della creatura dal database, gli articoli correlati ad essa e di visualizzarli nella pagina;
- `animal-list.php`: visualizza tutte le creature, il file si occupa di ottenere le creature dal database e di visualizzarle nella pagina;
- `animal-chart.php`: visualizza la classifica creature, tramite richiesta GET si può ordinare per nome, esistenza e non esistenza e si può escludere o no le creature già scoperte, il file si occupa di ottenere le creature ordinate dal database e di visualizzarle nella pagina;
- `form-add-animal.php`: visualizza la form per l'aggiunta di una creatura;
- `form-add-article.php`: visualizza la form per l'aggiunta di un articolo;
- `form-edit-article.php`: visualizza la form per la modifica di un articolo, tramite richiesta GET viene passato l'id dell'articolo da modificare, il file si occupa di ottenere i dati dell'articolo dal database e di visualizzare la form con il testo da modificare;
- `admin-home.php`: visualizza la pagina di amministrazione;
- `admin-animal-list.php`: simile a `animal-list.php` ma con la possibilità di eliminare le creature;
- `admin-article-list.php`: simile a `article-list.php` ma con la possibilità di modificare o eliminare gli articoli;

Per quanto riguarda la pagina di **login**, il file `/php/login.php` si occupa di verificare che le credenziali di accesso al sito inserite siano corrette e di assegnare il ruolo corretto a seconda dell'utente, mentre il file `/php/logout.php` si occupa di chiudere la sessione corrente.

#### 4.1.2 Interazione con il database

Le pagine si connettono al database tramite il file `/php/db-conn.php` che contiene le credenziali di accesso al database

Per gli **articoli** un utente può aggiungere un nuovo articolo, modificarne uno già esistente o eliminarlo:

- L'utente (*writer* o *admin*) aggiunge un nuovo articolo tramite la form presente nella pagina di aggiunta articolo, che invia i dati al file `add-article.php` che inserisce i dati nel database;
- Per **modificare** un articolo viene utilizzato il file `edit-article.php` che si occupa di aggiornare il testo di un articolo nel database;
- Per **eliminare** un articolo viene utilizzato il file `remove-article.php` che si occupa di eliminare l'articolo dal database.

All'interno dei articoli, gli utenti registrati possono aggiungere commenti o rispondere a commenti di altri utenti, Per la creazione di un commento o di una risposta si usa il file `add-comment.php` che si occupa di inserire il commento nel database e, se è una risposta, gli associa l'eventuale commento padre.

Per le **creature** solo gli amministratori possono aggiungere o eliminare creature:

- Per **aggiungere** i dati di una creatura al database si utilizza il file `add-animal.php`;
- Per **eliminare** una creatura dal database si utilizza il file `remove-animal.php`;

#### 4.1.3 Controllo input lato server

Prima di inserire un **articolo**, il file `add-article.php` fa dei controlli sull'input ricevuto:

1. Controlla che il **titolo** non sia vuoto e che non contenga caratteri speciali;
2. Controlla che il **sotto titolo** non sia vuoto;
3. Controlla che abbia ricevuto un **tag** valido;
4. Controlla che il **testo** contenga almeno 20 caratteri;
5. Controlla che un **immagine** sia stata caricata;
6. Se è stato compilato, controlla che la **creatura riferita** esista nel database.

Per la modifica di un **articolo** il file `edit-article.php` controlla solamente che il testo contenga almeno 20 caratteri.

Prima di inserire una **creatura**, il file `add-animal.php` fa dei controlli sull'input ricevuto:

1. Controlla che il **nome** non sia vuoto e che non contenga caratteri speciali;
2. Controlla che la **descrizione** contenga almeno 20 caratteri;

3. Controlla che abbia ricevuto uno **status** valido;
4. Controlla che la **data** non sia vuota e che sia nel formato ++AAAA-MM-GG++;
5. Controlla che un **immagine** sia stata caricata;
6. Infine controlla se la **creatura** non sia già presente nel database.

## 4.2 Javascript

### 4.2.1 Librerie utilizzate

- **jQuery**: libreria utilizzata per semplificare la manipolazione del DOM e l'interazione con il back-end, utilizzata nello script `/js/suggestion.js`;
- **Firebase**: database NoSQL fornito da Google, utilizzato per il salvataggio delle immagini delle creature e di articoli.

### 4.2.2 Immagini

Per il salvataggio delle immagini delle creature da renderizzare nelle pagine, si è deciso di utilizzare un database Firebase, che viene inizializzato nel file `/js/init-db.js` utilizzando il file di configurazione `/js/Firebase-config.js`. Con questo database si possono aggiungere immagini di creature nella form di aggiunta creature tramite lo script `/js/upload-animal.js`, che verranno salvate nella cartella `images/animals` di Firebase, mentre per aggiungere immagini di articoli si utilizza lo script `/js/upload-article.js` che salva le immagini dei articoli nella cartella `images/articles`.

### 4.2.3 Voti

Quando un utente effettua la votazione di una creatura viene chiamata la funzione `addVote` contenuta nello script `js/vote-event.js` che invia il voto al back-end per il salvataggio nel database tramite `/php/add-vote.php`. Sempre nello stesso script, quando un utente vuole rimuovere un voto, viene chiamata la funzione `removeVote` che rimuove il voto dal database tramite `/php/remove-vote.php`.

### 4.2.4 Suggerimenti

Lo script `/js/suggestion.js` viene utilizzato nella form di aggiunta articolo allo scopo di aiutare l'utente a selezionare la creatura riferita dall'articolo, infatti quando l'utente inizia a scrivere il nome della creatura, lo script invia una richiesta al back-end tramite `/php/suggestion.php` che restituisce una lista di creature suggerite sotto il campo della creatura riferito.

## 5 Implementazione Front-end

## 6 Accessibilità

### 6.1 Riferimenti normativi

- **WCAG4All**: <https://web.math.unipd.it/accessibility/test.html>
- **Standard WCAG 2.1**: <https://www.w3.org/Translations/WCAG21-it/>

### 6.2 Strumenti utilizzati

- **WAVE**: <https://wave.webaim.org/>
- **Total validator**: <https://www.totalvalidator.com/>
- **Accessibility Insights for Web**: <https://accessibilityinsights.io/>
- **Siteimprove A.C.**: <https://www.siteimprove.com/toolkit/accessibility-checker/>

### 6.3 Verifica

Per garantire il livello di accessibilità richiesto, sono stati effettuati dei test opportuni elencati, i quali verranno elencati di seguito, si specifica che quasi la totalità dei test per garantire il livello AAA non sono stati presi in considerazione.

#### 6.3.1 HTML

Sono stati verificati, attraverso procedure manuali o automatiche, i seguenti requisiti:

- uso corretto degli elementi semantici di HTML5 per la definizione della struttura delle pagine;
- uso corretto degli elementi semantici per dare enfasi al contenuto;
- lingua principale utilizzata per ogni pagina;
- verifica della presenza di ID duplicati sulla stessa pagina.

#### 6.3.2 CSS

Sono stati verificati, attraverso procedure manuali o automatiche, i seguenti requisiti:

- le immagini a scopo decorativo siano stati applicate via CSS;
- implementazione di unità di misure relative;
- scalabilità del layout del sito, sono stati implementati fogli di stile necessari per la visualizzazione del sito su dispositivi con schermi di dimensioni ridotte.

Inoltre sono stati implementati fogli di stile per la stampa delle pagine e si è verificato che in assenza di fogli di stile il sito sia comunque navigabile.

### 6.3.3 JavaScript

Attraverso *JavaScript* sono stati implementati alcuni script per il comportamento del sito e per la validazione degli input lato client.

Inoltre, in seguito, si è verificato che non vi siano presenti cambi di contesto inaspettati.

### 6.3.4 Immagini

Sono state verificate le seguenti proprietà delle immagini:

- le immagini puramente decorative sono state implementate via *CSS*;
- le immagini di contenuto invece sono state definite via *HTML* e sono state fornite alternative testuali opportune.

### 6.3.5 Link

Sono stati verificati, attraverso procedure manuali o automatiche, i seguenti requisiti:

- il colore non deve essere l'unico mezzo per identificare i link presenti nelle pagine;
- ancora dei link opportuna per far comprendere lo scopo di ciascuno di essi;
- comprendere la differenza tra link già visitati e non ancora visitati.

### 6.3.6 Colori

Sono state rispettate tutte le norme per garantire il contrasto (3:1) tra i colori di testo e lo sfondo, si è verificato, inoltre, che non siano l'unico mezzo per trasmettere informazioni.

Tuttavia ci sono testi che non presentano il grado di contrasto richiesto, in quanto sono dotati di proprietà *text-shadow* che fornisce un effetto di ombreggiatura al testo, per cui non è stato necessario intervenire.

### 6.3.7 Form

Sono stati verificati, attraverso procedure manuali o automatiche, i seguenti requisiti:

- comprensibilità della form, ovvero quali sono i dati richiesti e se opzionali od obbligatori;
- gli errori identificabili e comprensibili, anche attraverso l'uso di *aria-role="alert"* per notificare all'utente, via screed reader, eventuali errori di compilazione;
- presenza di attributi di autocompletamento nei campi delle form;
- compilabilità delle form anche senza l'uso del mouse.

### 6.3.8 Tabelle

Sono stati verificati, attraverso procedure manuali o automatiche, i seguenti requisiti:

- progettazione corretta della struttura delle tabelle;
- presenza della *caption* per descrivere lo scopo della tabella;
- trasformazione elegante delle tabelle in caso di visualizzazione delle stesse in dimensioni ridotte.

### 6.3.9 Modalità di input

Sono stati verificate che le operazioni possano essere svolte con un solo click per tutti i dispositivi di input supportati, ovvero mouse, tastiera e touch screen.

Inoltre è stata implementata la possibilità di retrocedere dal voto espresso per una creatura, in seguito ad un click accidentale oppure ad un errore di valutazione dell'utente votante.

### 6.3.10 Tastiera

Verifica della corretta implementazione dell'attributo *tabindex* per la navigazione del sito attraverso la tastiera nell'ordine corretto, inoltre l'elemento attualmente selezionato viene evidenziato in maniera opportuna.

È possibile usufruire di tutte le funzionalità offerte, servendosi dei controlli interattivi via tastiera, spostando il focus con il tasto *TAB* e selezionando l'elemento desiderato con il tasto *INVIO*.

Infine si specifica che non sono state implementate scorciatoie da tastiera.

### 6.3.11 Ruoli, stati e proprietà WAI ARIA

Ruoli *WAI ARIA* non sono stati necessari per la realizzazione del sito, in quanto sono risultati sufficienti gli elementi semantici di *HTML5*.

Dunque non sono stati implementati in maniera sostanziale neanche gli stati e le proprietà *WAI ARIA*, eccetto per alcuni casi che verranno discussi di seguito:

- **Breadcrumb:** è stato implementato un *aria-label* per specificarne il funzionamento;
- **Validazione input:** è stato implementato un *aria-label* per esplicitare eventuali errori nell'inserimento degli input;
- **Pulsanti:** è stato implementato un *aria-label* nei pulsanti *Submit*, di rimozione e modifica di contenuti per esplicitarne lo scopo.

### 6.3.12 Aiuti alla navigazione

È stato implementato, nelle pagine del sito in cui è necessario uno *scroll* per visualizzarne l'intero contenuto, un pulsante che permette di tornare in cima alla pagina.



### 6.3.13 Disorientamento

Per evitare questo fenomeno, si sono verificate le seguenti proprietà:

- il titolo rispetta la convenzione per cui deve essere dal particolare al generale;
- gli elementi ricorrenti nel sito, sono sempre posizionati nello stesso modo;
- presenza di una breadcrumb per fornire l'utente un orientamento all'interno del sito;
- possibilità di tornare alla home page in qualsiasi momento, cliccando sul logo del sito;
- possibilità di visualizzare il sito in modalità desktop e mobile, grazie all'implementazione di fogli di stile opportuni.

### 6.3.14 Separazione tra struttura, contenuto e comportamento

Per garantire la separazione tra struttura, contenuto e comportamento, si è fatto uso di fogli di stile esterni, in modo da poter modificare l'aspetto del sito senza dover intervenire sul codice *HTML*.

Analogamente, si sono impiegati script *PHP* e *JavaScript* esterni, rispettivamente per la gestione delle funzionalità lato server, templating delle pagine *HTML* e per la gestione delle funzionalità lato client

## 7 Suddivisione del lavoro

- **Francesco Ceccato:** si è occupato di pensare al dominio del sito e di revisionare ad ogni contributo, effettuando test del sito e segnalando errori e problemi.

Settore	Dettaglio
<b>HTML</b>	Struttura base di tutte le pagine ( <i>header</i> , <i>breadcrumb</i> e <i>footer</i> )
<b>CSS</b>	Sviluppo completo dell'aspetto del sito
<b>PHP</b>	Piccoli contributi e correzioni nel comportamento del sito
<b>JS</b>	Piccoli contributi e correzioni
<b>MySQL</b>	Piccoli contributi e correzioni

- **Nicola Ravagnan:**

Settore	Dettaglio
<b>HTML</b>	Struttura di alcune pagine e moduli richiesti
<b>CSS</b>	Piccoli contributi
<b>PHP</b>	Base per effettuare il templating delle pagine <i>html</i> Sistema di validazione lato server Comportamento delle pagine lista articoli e animali
<b>JS</b>	Sistema di validazione lato client Sistema di visualizzazione del gradiente delle votazioni
<b>MySQL</b>	Piccoli contributi e correzioni

- **Matteo Stocco:**

Settore	Dettaglio
<b>HTML</b>	Struttura delle pagine delle form Struttura della sezione commenti e moduli richiesti
<b>CSS</b>	Piccoli contributi
<b>PHP</b>	Comportamento delle pagine di form Comportamento del sistema di commenti
<b>JS</b>	Meccanismo di suggerimento del campo creatura Comportamento visualizzazione sezione commenti
<b>MySQL</b>	Piccoli contributi e correzioni

- **Sebastiano Sanson:**

Compito	Dettaglio
<b>HTML</b>	Struttura delle pagine della sezione admin
<b>CSS</b>	Piccoli contributi e correzioni
<b>PHP</b>	Comportamento della sezione admin Comportamento del carosello Comportamento generale del sito
<b>JS</b>	Sistema di votazione e classifica
<b>MySQL</b>	Ideazione e sviluppo tabelle e views
<b>Firebase</b>	Configurazione e implementazione meccanismo storicizzazione delle immagini

Per quanto riguarda l'accessibilità il lavoro è stato suddiviso equamente tra i componenti del gruppo.

## 8 Note finali

In futuro, si vuole implementare la eliminazione delle immagini presenti su firebase cloud storage, in corrispondenza con l'eliminazione di un articolo o creatura