

# 1. Studio di Fattibilità

Lo studio di fattibilità valuta se il progetto è realizzabile considerando tutte le funzionalità del sito vetrina.

## Studio di Fattibilità - Cybersecurity Lab Site

### 1. Introduzione

Il progetto "Cybersecurity Lab" consiste nello sviluppo di un sito web vetrina per un laboratorio di cybersecurity. Il sito ha l'obiettivo di presentare il laboratorio, fornendo informazioni su:

- La storia del laboratorio.
  - I contatti (es. email, telefono, indirizzo).
  - I progetti di ricerca.
  - Le pubblicazioni accademiche del laboratorio.
- Inoltre, il sito consente agli amministratori di caricare nuove pubblicazioni (inclusi file PDF).

Gli obiettivi principali sono:

- Fornire un'interfaccia user-friendly per promuovere il laboratorio.
- Consentire agli utenti (studenti, ricercatori, pubblico generico) di accedere a informazioni sul laboratorio.
- Garantire un sistema sicuro per l'upload di pubblicazioni da parte degli amministratori.
- Implementare un'architettura scalabile e manutenibile.

### 2. Fattibilità Tecnica

- Tecnologie scelte:
  - Frontend: Next.js (React framework) per un'interfaccia reattiva, performante e ottimizzata per SEO (importante per un sito vetrina);  
  
Shadcn: una libreria open-source per i componenti UI (bottoni, modali, toggle, card, ecc.)  
  
Tailwind: un framework utility-first che consente di applicare stili CSS direttamente all'interno del markup JSX tramite classi predefinite
  - Backend: FastAPI (Python) per un'API RESTful veloce e con validazione automatica.
  - Database: SQLite per la gestione di pubblicazioni e progetti (scelto per semplicità, ma scalabile a PostgreSQL se necessario).
  - Autenticazione: JWT (JSON Web Token) per gestire l'accesso degli amministratori.
  - Hosting dei file: I file PDF delle pubblicazioni saranno salvati nella directory public/pdf del frontend.

- Competenze richieste:
  - Conoscenza di JavaScript/TypeScript e React per il frontend.
  - Conoscenza di Python e FastAPI per il backend.
  - Competenze di base in gestione di database relazionali.
  - Competenze di base in design web per creare un'interfaccia accattivante.

### **3. Fattibilità Economica**

- Costi stimati:
  - Sviluppo: Nessun costo diretto, poiché il progetto è svolto da uno studente come progetto di tesi.
  - Infrastruttura: Hosting su un server locale durante lo sviluppo (costo zero). Per un'eventuale produzione, si stima un costo di 5-10€/mese per un hosting base (es. Vercel per il frontend, Heroku per il backend).
  - Manutenzione: Minima, stimata in 2-3 ore al mese per aggiornamenti e bug fixing.
- Benefici attesi:
  - Promuovere il laboratorio di cybersecurity, aumentando la visibilità online.
  - Fornire un servizio utile per la consultazione di progetti e pubblicazioni.
  - Acquisire competenze pratiche in sviluppo web e gestione di progetti software.
- Valutazione: Il progetto è economicamente fattibile, con costi minimi e benefici formativi e promozionali significativi.

### **4. Fattibilità Operativa**

- Impatto sull'organizzazione: Il sito sarà utilizzato dal laboratorio di cybersecurity per promuovere le proprie attività. Gli amministratori del laboratorio (es. ricercatori o personale) potranno gestire le pubblicazioni.
- Facilità d'uso:
  - Gli utenti finali (pubblico generico, studenti, ricercatori) possono accedere alle sezioni del sito (storia, contatti, progetti, pubblicazioni) senza autenticazione.
  - Gli amministratori possono caricare pubblicazioni tramite un'interfaccia semplice, previa autenticazione.
- Valutazione: Il sito è intuitivo e non richiede formazione complessa per gli utenti o gli amministratori.

### **5. Fattibilità Temporale**

- Tempistiche stimate:

- Analisi dei requisiti: 2 giorni.
  - Progettazione (inclusi diagrammi UML: 2 giorni.
  - Sviluppo del backend (API per pubblicazioni e progetti): 5 giorni.
  - Sviluppo del frontend (pagine statiche e dinamiche): 5 giorni.
  - Testing e debugging: 5 giorni.
  - Documentazione finale: 2 giorni.
  - Totale: 21 giorni.
- Scadenze: Supponendo una scadenza a fine maggio 2025, il progetto è completabile entro il termine.
  - Valutazione: Le tempistiche sono realistiche per un progetto di questa portata.

## 6. Conclusioni e Raccomandazioni

Il progetto "Cybersecurity Lab Site" è fattibile da tutti i punti di vista: tecnico, economico, operativo e temporale.

---

# 2. Analisi dei Requisiti

L'analisi dei requisiti include tutte le funzionalità del sito vetrina, non solo il caricamento delle pubblicazioni.

## Analisi dei Requisiti - Cybersecurity Lab Site

### 1. Introduzione

Questo documento descrive i requisiti per il sistema "Cybersecurity Lab Site", un sito web vetrina per un laboratorio di cybersecurity. Il sito presenta il laboratorio (storia, contatti, progetti, pubblicazioni) e consente agli amministratori di caricare nuove pubblicazioni.

### 2. Stakeholder

- *Utenti finali* (pubblico generico, studenti, ricercatori):
  - Bisogno: Accedere a informazioni sul laboratorio (storia, contatti, progetti, pubblicazioni).
- *Amministratori* (personale del laboratorio):
  - Bisogno: Caricare nuove pubblicazioni (titolo, autori, anno, categoria, abstract, file PDF) previa autenticazione.
- *Sviluppatore*:
  - Bisogno: Implementare un sistema manutenibile e scalabile, con documentazione chiara.

### 3. Requisiti Funzionali

- RF1: Il sistema deve mostrare una pagina "Chi siamo" con informazioni sul laboratorio (es. anno di fondazione, mission, obiettivi).
- RF2: Il sistema deve mostrare una pagina "Contatti" con informazioni di contatto (es. email, telefono, indirizzo).
- RF3: Il sistema deve mostrare una pagina "Progetti" con un elenco di progetti di ricerca (titolo, descrizione, stato: in corso/completato).
- RF4: Il sistema deve mostrare una pagina "Pubblicazioni" con un elenco di pubblicazioni (titolo, autori, anno, categoria, abstract, link al PDF).
- RF5: Il sistema deve consentire agli amministratori di caricare una nuova pubblicazione, includendo:
  - Titolo (stringa).
  - Autori (stringa).
  - Anno (numero intero).
  - Categoria (stringa).
  - Descrizione (testo).
  - File PDF.
- RF6: Il sistema deve autenticare gli amministratori tramite username e password.
- RF7: Il sistema deve salvare i file PDF caricati in una directory accessibile (public/pdf).
- RF8: Il sistema deve fornire un link al file PDF per ogni pubblicazione.

### 4. Requisiti Non Funzionali

- RNF1 (Sicurezza): Solo gli amministratori autenticati possono caricare pubblicazioni.
- RNF2 (Usabilità): L'interfaccia deve essere intuitiva, responsive e accessibile su dispositivi mobili.
- RNF3 (Prestazioni): Le pagine devono caricarsi in meno di 2 secondi con un database di 1000 progetti e pubblicazioni.
- RNF4 (Scalabilità): Il sistema deve essere progettato per supportare un futuro passaggio a un database più robusto (es. PostgreSQL).
- RNF5 (Manutenibilità): Il codice deve essere ben strutturato e documentato.
- RNF6 (SEO): Il sito deve essere ottimizzato per i motori di ricerca (es. metadati, URL leggibili).

### 5. Casi d'Uso

- Caso d'Uso 1: Visualizza Chi siamo

- Attore: Utente finale.
    - Descrizione: L'utente accede alla pagina /chi-siamo e visualizza informazioni sul laboratorio.
    - Precondizioni: Nessuna.
    - Postcondizioni: L'utente ha letto la storia del laboratorio.
  - Caso d'Uso 2: Visualizza Contatti
    - Attore: Utente finale.
    - Descrizione: L'utente accede alla pagina /contatti e visualizza le informazioni di contatto.
    - Precondizioni: Nessuna.
    - Postcondizioni: L'utente può contattare il laboratorio.
  - Caso d'Uso 3: Visualizza Progetti
    - Attore: Utente finale.
    - Descrizione: L'utente accede alla pagina /progetti e visualizza un elenco di progetti.
    - Precondizioni: Nessuna.
    - Postcondizioni: L'utente può leggere i dettagli dei progetti.
  - Caso d'Uso 4: Visualizza Pubblicazioni
    - Attore: Utente finale.
    - Descrizione: L'utente accede alla pagina /pubblicazioni e visualizza un elenco di pubblicazioni.
    - Precondizioni: Nessuna.
    - Postcondizioni: L'utente può cliccare sul link per scaricare il PDF.
  - Caso d'Uso 5: Carica Pubblicazione
    - Attore: Amministratore.
    - Descrizione: L'amministratore accede alla pagina /admin, inserisce i dati della pubblicazione e clicca su "Carica".
    - Precondizioni: L'amministratore deve essere autenticato.
    - Postcondizioni: La pubblicazione è salvata nel database e il file PDF è salvato nella directory public/pdf.
-

## 3. Schema UML

Aggiorniamo gli schemi UML per includere tutte le funzionalità del sito.

### 3.1 Diagramma dei Casi d'Uso

*Attori:*

- Utente finale.
- Amministratore.

*Casi d'uso:*

- Visualizza Storia (Utente finale).
- Visualizza Contatti (Utente finale).
- Visualizza Progetti (Utente finale).
- Visualizza Pubblicazioni (Utente finale).
- Carica Pubblicazione (Amministratore).
- Autenticazione (Amministratore).

*Descrizione testuale:*

- L'Utente finale è collegato ai casi d'uso "Visualizza Storia", "Visualizza Contatti", "Visualizza Progetti" e "Visualizza Pubblicazioni".
- L'Amministratore è collegato ai casi d'uso "Autenticazione" e "Carica Pubblicazione".
- Il caso d'uso "Carica Pubblicazione" include "Autenticazione" come precondizione.

### 3.3 Diagramma di Sequenza (Carica Pubblicazione)

*Attori e Oggetti:*

- Amministratore.
- Frontend (/admin).
- Proxy (/api/publications).
- Backend (/publications).
- Database.

*Flusso:*

1. L'Amministratore compila il form e clicca su "Carica".
2. Il Frontend invia una richiesta POST a /api/publications con i dati e il token.
3. Il Proxy inoltra la richiesta al Backend.
4. Il Backend verifica il token e valida i dati.
5. Il Backend salva i dati nel Database e il file nella directory public/pdf.
6. Il Backend restituisce una risposta.
7. Il Proxy inoltra la risposta al Frontend.
8. Il Frontend mostra un messaggio di successo.

### 3.4 Diagramma di Sequenza (Visualizza Progetti)

*Attori e Oggetti:*

- Utente finale.
- Frontend (/progetti).
- Proxy (/api/projects).
- Backend (/projects).
- Database.

*Flusso:*

1. L'Utente finale accede alla pagina /progetti.
2. Il Frontend invia una richiesta GET a /api/projects.
3. Il Proxy inoltra la richiesta al Backend (/projects).
4. Il Backend recupera i progetti dal Database.
5. Il Backend restituisce l'elenco dei progetti.
6. Il Proxy inoltra la risposta al Frontend.
7. Il Frontend mostra l'elenco dei progetti.