



# Fondamenti del Web

**Ingegneria del Software e Fondamenti Web**

**Corso di Laurea in Ingegneria Informatica e dell'Automazione**

Anno Accademico 2023/2024

Prof. Antonio Ferrara

# Verifica finale

## Modalità 2 (Progetto)

Si vuole realizzare un'**applicazione web** che ricada in un **qualunque dominio o scenario applicativo** a completa scelta del gruppo

L'applicazione dovrà essere realizzata come **Single Page Application** con **client-side rendering** e dovrà essere composta da:

- un **backend** che fornisce ai diversi attori le corrette API
- un **frontend** che permette il loro agevole utilizzo

### Requisiti tecnici

- Il backend dell'applicazione dovrà essere realizzato mediante Node.js ed Express
- I dati dovranno essere memorizzati all'interno di un database MongoDB in cloud
- Il frontend dovrà essere realizzato con React

# Verifica finale

## Modalità 2 (Progetto)

### Alcuni suggerimenti tecnici

Si suggerisce di prevedere sia **API autenticate** sia un **sistema di login** lato frontend per l'autorizzazione dei diversi attori  
L'autenticazione sicura può essere effettuata con password con hashing, Passport, OAuth, o altri meccanismi  
È possibile utilizzare **cookie di sessione** per mantenere l'accesso

Se si decide di creare qualche forma di **interazione real-time** (es. chat, aggiornamenti in tempo reale, strumenti collaborativi real-time), si suggerisce di utilizzare **Socket.IO** per la sua semplicità (<https://socket.io/>), ma è possibile utilizzare altre strategie come Server-Sent Events oppure i classici WebSocket

Nell'uso di **React** si suggerisce (consigliato) l'uso di **Axios** per le **richieste HTTP** come alternativa alla classica Fetch  
Si suggerisce, inoltre, l'uso di **Material UI** come libreria per layout e componenti base per React, eventualmente accompagnato da un **CSS personalizzato**  
Alternative sono **React Bootstrap** o un CSS scritto da zero

L'eventuale **deployment** dell'applicazione potrà essere realizzato con un qualunque servizio (es. Vercel, Render, AWS, Netlify, ...)  
Può facoltativamente essere adottata qualunque strategia avanzata per l'**architettura dell'applicazione**: microservizi, microfrontend, uso di container e strumenti come Kubernetes, ...)

# Verifica finale

## Modalità 2 (Progetto)

### Istruzioni pratiche per lo svolgimento

- Il progetto potrà essere realizzato mediante l'IDE di sviluppo preferito (consigliati WebStorm e Visual Studio Code)
- Sono da evitare ricevimenti o richieste di consigli e aiuto al docente: si tratta di una prova d'esame!
- È consentito consultare materiale e risorse esterne
- Non è consentito copiare da materiale e risorse esterne
- È severamente vietato il riuso di codice fra gruppi, che sarà punito con l'annullamento di entrambe le prove progettuali

# Verifica finale

## Modalità 2 (Progetto)

### Istruzioni pratiche per la consegna

Sarà necessario consegnare:

- Una breve descrizione dello scenario applicativo e dell'architettura dell'applicazione che includa anche:
  - Un diagramma UML dei casi d'uso (e opzionalmente anche delle sequenze)
  - Una descrizione del modello dei dati
  - Una documentazione delle API (backend) e una breve descrizione dei componenti React (frontend)
- I file del progetto

Tutti questi file dovranno essere compressi in un unico file .zip e allegati come soluzione dell'attività su Microsoft Teams che sarà aperta prossimamente

Nota bene. La cartella del progetto dovrà essere composta da due sottocartelle per backend e frontend. Da tali cartelle andranno rimosse le sottocartelle **node\_modules**, non necessarie nella condivisione. Inoltre, dovranno essere fornite le eventuali credenziali per il test delle diverse funzionalità.

# Verifica finale

## Modalità 2 (Progetto)

### Valutazione

La valutazione del progetto, per un totale massimo di 10 punti, avverrà secondo la tabella a destra  
Tale valutazione sarà assegnata ugualmente a tutti i componenti del gruppo

Inoltre, la discussione del progetto con possibili domande teoriche relative agli argomenti del progetto sarà valutata individualmente fino a 5 punti

Obiettivo	Obbl.	Punteggio minimo	Punteggio massimo
Realizzazione di API	✓	0	2
Uso di React	✓	0	2
Realizzazione e uso di fogli di stile CSS		0,5	1
Uso di Material UI e Bootstrap		0,5	1
Implementazione di meccanismi di autenticazione sicura		0,5	2
Uso di cookie per la sessione		0,5	1
Uso di tecnologie per l'interazione real-time		1	2
Deployment dell'applicazione		1	1
Uso di tecniche avanzate di qualunque genere (es. uso di microfrontend, strategie per scalabilità/disponibilità, uso di webhook, container, microservizi, cluster Kubernetes, streaming API ...)		1	2
Documentazione	✓	0	1