



# Fondamenti del Web

**Ingegneria del Software e Fondamenti Web**

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

Anno Accademico 2024/2025

Prof. Antonio Ferrara

# Verifica finale

## Presentazione Progetto e Prova Orale

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

## Presentazione Progetto e Prova Orale

### 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  
Per le sessioni, è possibile utilizzare **cookie di sessione** per mantenere l'accesso, autenticazioni basate su **token JWT**, o strategie ibride come **access e session token**

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** per il frontend è facoltativo l'uso di **Axios** per le **richieste HTTP** come alternativa alla classica Fetch  
Si suggerisce l'uso di hook come **useContext** per la creazione di uno stato dell'app (altri hook custom sono possibili purché giustificati) e librerie come **React Router** per la navigazione fra diverse pagine  
Per componenti e layout, si suggeriscono l'uso di **Material UI** come libreria di componenti base per React, eventualmente accompagnato da un **CSS personalizzato**, e a discrezione, tool come **React Bootstrap** e **Tailwind CSS**

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

## Presentazione Progetto e Prova Orale

### 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
- ChatGPT può essere un supporto per capire o imparare a fare qualcosa in più: tutto ciò che è ispirato alle risposte di ChatGPT (e simili) deve essere studiato e compreso a pieno prima di essere inserito nel progetto

# Verifica finale

## Presentazione Progetto e Prova Orale

### 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 aperta per ciascun appello

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

## Presentazione Progetto e Prova Orale

### Valutazione

La valutazione del progetto, per un totale massimo di 10 punti, avverrà secondo la tabella a destra

Si ricorda che la semplice realizzazione dell'obiettivo, in assenza di elementi qualitativi aggiuntivi, sarà valutata con il punteggio minimo all'interno della relativa fascia di valutazione

Tale valutazione sarà assegnata ugualmente a tutti i componenti del gruppo

Inoltre, la discussione del progetto con possibili domande teoriche e pratiche relative agli argomenti del progetto potrà essere valutata individualmente, in casi particolarmente meritevoli, con un bonus fino a 5 punti

Obiettivo	Obbl.	Punteggio minimo	Punteggio massimo
Realizzazione di API con Express	✓	0	2
Realizzazione di frontend con React	✓	0	2
Realizzazione di fogli di stile CSS from scratch		0,5	1
Uso appropriato e consapevole di tool e librerie come Material UI, Bootstrap e Tailwind		0,5	1
Implementazione di meccanismi di autenticazione sicura e mantenimento della sessione (token, cookie, OAuth)		0,5	3
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

# Verifica finale

## Presentazione Progetto e Prova Orale

### Valutazione

Si ricorda che la sessione di Presentazione del Progetto e Prova Orale non deve essere concepita come un semplice racconto divulgativo di ciò che fa il progetto o le linee di codice, ma costituisce a tutti gli effetti una prova d'esame orale, incentrata sulla presentazione tecnica e consapevole del lavoro svolto

Il tempo massimo a disposizione per ciascun gruppo è di 30 minuti, pertanto è consigliato strutturare l'intervento in modo da:

- presentare in pochi minuti le funzionalità principali dell'applicazione sviluppata;
- dedicare alcuni minuti successivi a illustrare aspetti tecnici rilevanti, compresi elementi di codice, possibilmente con un focus individuale da parte di ciascun componente del gruppo sulla parte di progetto di propria competenza: questo contributo dovrà evidenziare con chiarezza e motivazione il raggiungimento di ciascuno degli obiettivi previsti dalla valutazione
- lasciare sufficiente spazio alla parte finale della sessione, dedicata a domande individuali e di gruppo

Si sottolinea che non devono essere incluse nel progetto parti che i membri del gruppo non siano in grado di spiegare in modo adeguato o realizzare autonomamente

A tal proposito, si ricorda che tutti i membri del gruppo devono essere preparati su tutti gli aspetti pratici del corso: ciò implica la capacità di comprendere e argomentare anche sezioni del progetto non realizzate personalmente, nonché di affrontare eventuali richiami teorico-pratici relativi agli argomenti trattati durante il corso

Si sconsiglia, quindi, vivamente la partecipazione alla sessione d'esame a gruppi i cui componenti non siano sufficientemente preparati ad affrontare una discussione orale e tecnica completa sul progetto e sui contenuti del corso