

# **APPLICAZIONI WEB**

2019/2020



## **Sistema per la gestione degli esami universitari**

RICCARDO DEANA  
PIERLUIGI FABBRO  
ENRICO GERUSSI

115739  
116461  
124592

## Obiettivi del progetto:

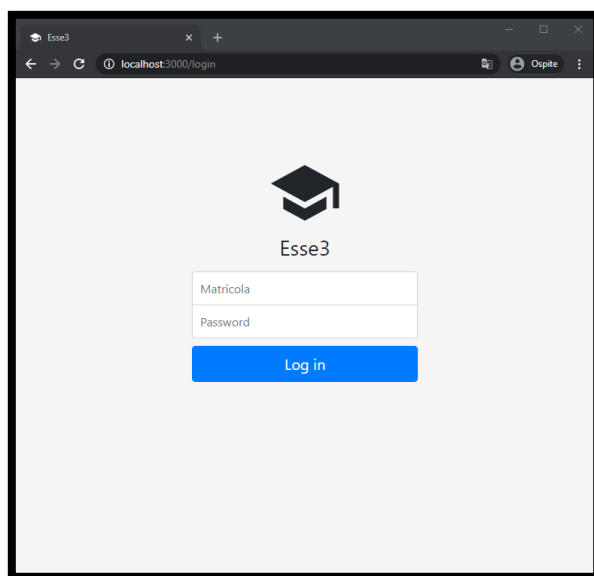
Realizzare un'applicazione web in grado di gestire la prenotazione degli esami universitari. Gli studenti devono avere la possibilità di eseguire il login e logout ad una pagina per prenotarsi agli appelli d'esame inerenti al loro corso di studi. Gli esami disponibili alla prenotazione non devono essere già prenotati o superati dallo studente che ne fa la richiesta. Ogni appello d'esame ha un numero massimo di partecipanti, è pertanto necessario effettuare un controllo per non superare tale limite prima della prenotazione.

## Tecnologie utilizzate:

- JavaScript: linguaggio di programmazione per lo sviluppo front end e back end dell'applicazione web;
- Npm: gestore di pacchetti utilizzato per l'ambiente di runtime JavaScript Node.js;
- Express: framework utilizzato per il processo di routing;
- MongoDB Atlas: database non relazionale per l'archiviazione delle informazioni. È stata utilizzata la libreria mongoose per MongoDB che permette la gestione relazionale dei dati e la validazione degli schemi mantenendo comunque la flessibilità offerta dai database NoSQL;
- EJS: template engine che consente l'utilizzo del codice JavaScript nelle views;
- Bootstrap: framework per l'interfaccia grafica;
- Bcryptjs: utilizzato per eseguire l'hashing delle password;
- Body-parser: middleware per l'analisi sintattica del body delle richieste http;
- JWT: standard utilizzato per l'autenticazione tramite token.

## Scelte progettuali:

È stata realizzata un'applicazione web che consente da un'unica schermata il login come studente oppure come amministratore. Dopo aver verificato le credenziali, l'utente viene reindirizzato alla pagina corretta in base al suo ruolo. Per sicurezza l'applicazione genera un token JWT che scade dopo 15 minuti di inattività dell'utente oppure se viene effettuato il logout. Ogni volta che l'utente cambia pagina o compie un'azione viene generato un nuovo token e viene cancellato quello vecchio. I token JWT vengono salvati nel database MongoDB.

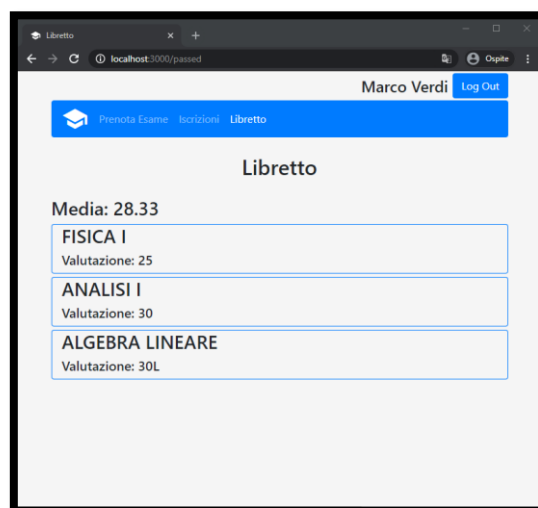
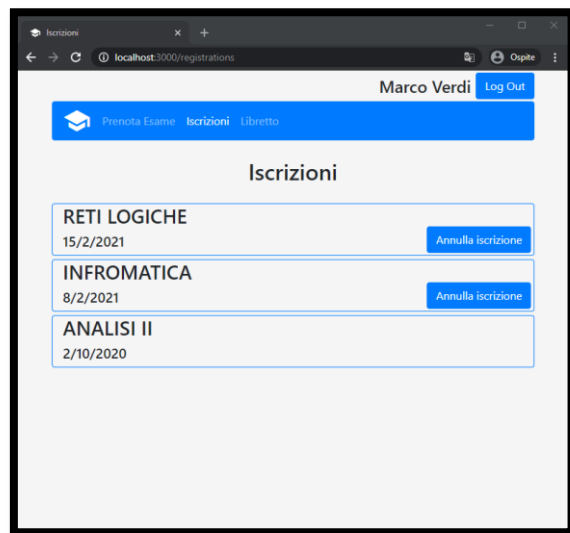
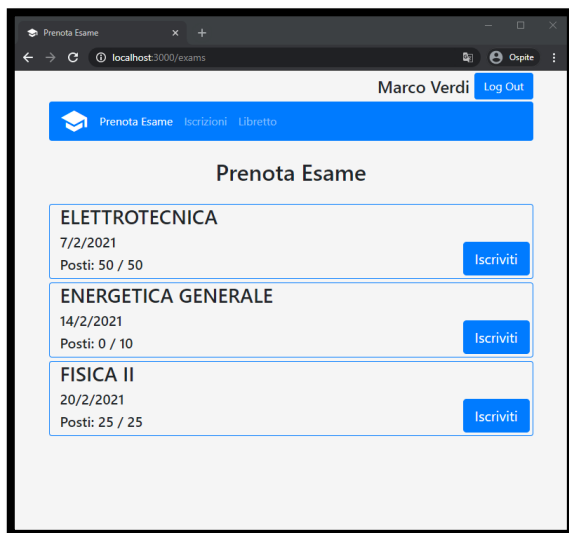


L'interfaccia grafica, realizzata utilizzando bootstrap, è responsive e si adatta alle varie risoluzioni. Quando lo spazio non è sufficiente, come nel caso della visualizzazione da smartphone, la navbar collassa, assumendo l'aspetto di un menu a tendina. Gli altri elementi invece vengono scalati, mantenendo le proporzioni e le posizioni reciproche, dato che sono state utilizzate misure relative alla risoluzione per il loro posizionamento.

Dopo aver eseguito il login l'utente vede il proprio nome in alto a destra accanto al pulsante di logout.

Lo studente può eseguire le seguenti operazioni:

- Iscrivere agli esami che fanno parte del suo corso di studi, a cui non si è già iscritto e non ha già sostenuto. Si verifica se il limite dei posti disponibili non è stato superato. Se l'iscrizione va a buon fine, viene decrementato il numero di posti liberi.
- Visualizzare l'elenco delle prenotazioni con la possibilità di annullarle se la data dell'appello d'esame non è già stata superata, in tal caso il bottone per l'annullamento non viene visualizzato. Quando un'iscrizione viene annullata si aggiunge un posto libero all'appello corrispondente.
- Visualizzare il libretto con i voti degli esami sostenuti e la media.



L'amministratore può eseguire le seguenti operazioni:

- Aggiungere uno studente utilizzando una matricola che non è già stata utilizzata.
- Rimuovere uno studente esistente inserendo la sua matricola.
- Aggiungere un appello d'esame. Non è possibile aggiungere due esami con lo stesso nome, la stessa facoltà e la stessa data.
- Registrare il voto di un esame solo se lo studente si è effettivamente iscritto ad un appello corrispondente. Il voto deve essere compreso tra 18 e 30, e la lode è possibile solo se abbinata ad un 30.

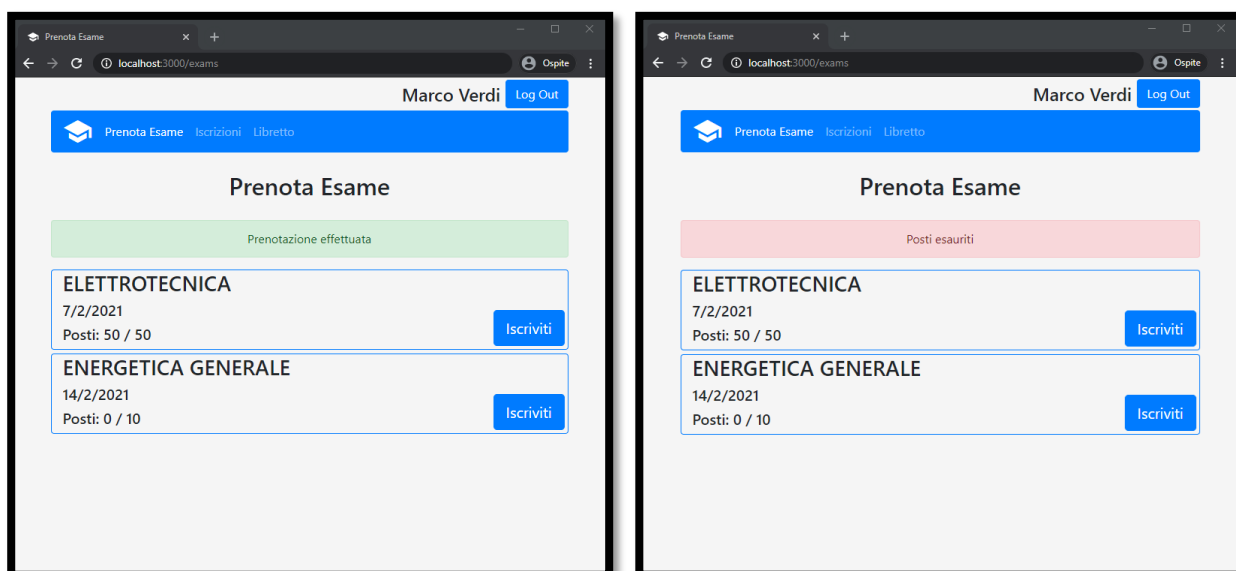
The screenshot shows the 'Aggiungi Studente' page in a web browser. The browser's address bar shows 'localhost:3000/addStudent'. The page has a header with 'Mario Rossi' and a 'Log Out' button. A navigation bar contains links: 'Aggiungi Studente' (active), 'Rimuovi Studente', 'Aggiungi Esame', and 'Registra Voto'. The main form is titled 'Aggiungi Studente' and includes fields for 'Nome:', 'Cognome:', 'Facoltà:' (a dropdown menu with '-- selezionare una facoltà --'), 'Matricola:', and 'Password:'. A blue 'Aggiungi' button is at the bottom.

The screenshot shows the 'Rimuovi Studente' page in a web browser. The browser's address bar shows 'localhost:3000/remStudent'. The page has a header with 'Mario Rossi' and a 'Log Out' button. A navigation bar contains links: 'Aggiungi Studente', 'Rimuovi Studente' (active), 'Aggiungi Esame', and 'Registra Voto'. The main form is titled 'Rimuovi Studente' and includes a 'Matricola:' field. A blue 'Rimuovi' button is at the bottom.

The screenshot shows the 'Aggiungi Esame' page in a web browser. The browser's address bar shows 'localhost:3000/addExam'. The page has a header with 'Mario Rossi' and a 'Log Out' button. A navigation bar contains links: 'Aggiungi Studente', 'Rimuovi Studente', 'Aggiungi Esame' (active), and 'Registra Voto'. The main form is titled 'Aggiungi Esame' and includes fields for 'Esame:', 'Facoltà:' (a dropdown menu with '-- selezionare una facoltà --'), 'Data:' (with a date picker showing 'gg/mm/aaaa'), and 'Numero posti:'. A blue 'Aggiungi' button is at the bottom.

The screenshot shows the 'Registra Voto' page in a web browser. The browser's address bar shows 'localhost:3000/regGrade'. The page has a header with 'Mario Rossi' and a 'Log Out' button. A navigation bar contains links: 'Aggiungi Studente', 'Rimuovi Studente', 'Aggiungi Esame', and 'Registra Voto' (active). The main form is titled 'Registra Voto' and includes fields for 'Matricola studente:', 'Esame:', 'Voto:', and 'Lode:'. A blue 'Registra' button is at the bottom.

Alle pagine dell'applicazione web vengono inviati opportuni messaggi per segnalare l'esito positivo o negativo delle operazioni richieste dall'utente.



### Struttura del database:

- *Studenti* (nome, cognome, matricola, facoltà, password)
- *Amministratori* (nome, cognome, matricola, password)
- *Token studenti* (idStudente, token, dataCreazione)
- *Token amministratori* (idAdmin, token, dataCreazione)
- *Esami* (nome, facoltà, postiTot, postiLiberi, data)
- *Iscrizioni* (idEsame, nomeEsame, studente, data)
- *Superati* (studente, esame, voto, lode)

### Esecuzione dell'applicazione web:

- Installare i pacchetti necessari con il comando da terminale: `npm install`;
- Avviare l'esecuzione dell'applicazione con il comando: `npm start`;
- Accedere dal browser in locale sulla porta 3000: <http://localhost:3000>;
- È possibile testare tutte le funzionalità dell'applicazione eseguendo il login con le credenziali di un amministratore o uno studente riportate nella tabella:

	Matricola	Password
<b>Amministratore</b>	111111	Amministrativo1
<b>Studente</b>	222222	studente1

### Fonti utilizzate per lo sviluppo:

- Nozioni teoriche e tutorial seguiti durante le lezioni del corso di "Applicazioni Web"
- [https://www.mrw.it/javascript/esempio-applicazione-web-realizzata-express\\_12446.html](https://www.mrw.it/javascript/esempio-applicazione-web-realizzata-express_12446.html)