

# Insegnamento di Tecnologie Web

## CdS In Informatica

(A.A. 2023-24)

Esame scritto del 24/01/2025

**Nome:**

**Cognome:**

**Matricola:** OBBLIGATORIA

**Corso di Studi:** e.g. Informatica

**Anno di frequenza:** e.g. 2024-25

**Attenzione:**

- Questi computer sono limitati ad accedere solo ad alcuni siti: eol.unibo.it, virtuale.unibo.it, developer.mozilla.org, getbootstrap.com, tailwindcss.com e site212248.tw.cs.unibo.it. Non funzionano Google, stack overflow, etc.
- *Rispondere solo negli spazi delimitati dai blocchi `` ` qui la risposta ` ``, senza modificarli o eliminarli.*
- *Consegnare solo questo file. Copiare ed incollare dentro agli appositi spazi la risposta per intero.*
- *Si può decidere se inserire il CSS inline nel file HTML o metterlo in un file esterno. Nel secondo caso inserire l'elemento nella posizione corretta e mettere il CSS in un blocco separato.*
- *Utilizzare strumenti non concessi e telefoni cellulari per rispondere alla domanda ha un tasso di tolleranza pari a zero.*
- *You can use either English or Italian for your answers. In this case, ask your Professor for the translated document*
- *Per favore, per favore, per favore: nessun errore di ortografia. Questa è un università e non la scuola elementare.*

### **FAQ sull'esame:**

- *Dove trovo i file che ho scaricato?*

Nella cartella download sul vostro computer. Per navigare tra le cartelle aprite l'applicazione *Nautilus*. Per aprire l'applicazione *Nautilus* portate il mouse in una zona senza finestre attive e fate click destro. Compare l'elenco delle applicazioni che potete aprire. Scegliete *Nautilus*. \* *Ho ridotto l'editor ad icona per sbaglio, che faccio?*

Con la combinazione CTRL+Tab accedete e scorrete tra tutte le applicazioni aperte. Trovate l'icona della finestra chiusa per errore e selezionatela. La finestra ricomparirà.

- *Ho chiuso l'editor per sbaglio, che faccio?*

Come sopra: portate il mouse in un'area senza finestre, fate click destro e aprite *gedit* oppure *jedit*. Alternativamente usate *Nautilus* per navigare fino alla finestra dove avete il file di testo e fate doppio click.

- *Come faccio a testare HTML/CSS/JS?*

Aprite *Firefox*. Una volta dentro, fate CTRL+O per aprire un file all'interno del browser. Navigare fino alla cartella in cui avete il filee selezionate il documento HTML di vostro interesse. Il documento HTML deve avere impostati al suo interno i link ai file CSS e JS di vostro interesse.

## Domanda # 1: Domande di base (9 punti totali)

### a) JS

Spiega la funzione del DOM nello sviluppo Front-End, rispondendo alle seguenti domande: \* Per cosa sta l'acronimo \* A cosa ci si riferisce con "manipolazione del DOM" \* Cosa è lo Shadow DOM Fornire esempi laddove necessario.

### b) API

Il seguente URL

<https://dummyimage.com/500x600/fa0546/000000.png&text=WebTechnologies>

ritorna l'immagine qui presente:



Analizzare gli elementi dell'API DummyImage fornendo un'ipotesi sul significato di ciascun componente (fa0546 corrisponde al colore rosso).

### c) Mongoose Schema

Scrivere lo/gli schema(ta) Mongoose che possa(no) raccogliere tutte le informazioni della seguente descrizione:

“Edward Said terrà una conferenza intitolata ‘Orientalismo e Cultura’ il 15 maggio 2025 presso l’Università di Columbia. La conferenza, della durata di 2 ore, includerà un’analisi del suo libro ‘Orientalismo’ e una sessione di dibattito. L’evento sarà disponibile sia in presenza (max 200 posti) che in streaming. I partecipanti riceveranno materiale digitale sulla critica post-coloniale e accesso alla registrazione. Il biglietto costa 15\$, con il ricavato devoluto all’UNRWA. Saranno presenti interpreti per arabo e inglese.”

NB: nel testo possono essere identificati più di uno schema. Consiglio: Per fare riferimento ad uno schema dal campo di un altro, specificare come type “Schema.Types.ObjectId”

```
const mongoose = require('mongoose');
const Schema = mongoose.Schema;

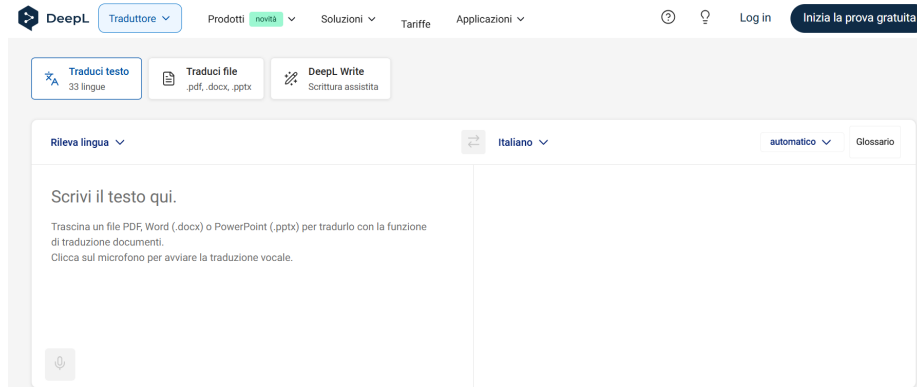
const eventSchema = new Schema({
  ...
});

const otherSchema = new Schema({
  ...
})

module.exports = {
  Event: mongoose.model('Event', eventSchema),
  ...
};
...
```

## Domanda #2 - HTML + CSS (massimo 9 punti)

Riprodurre la interfaccia di un traduttore online. E' possibile utilizzare Bootstrap e/o Tailwind



- È possibile (e consigliato) utilizzare Bootstrap e/o tailwindcss
- Le icone e il logo sono disponibili anche come svg nella cartella immagini
- Non è necessario rispettare colori e testo in maniera precisa (e.g. usare lorem ipsum)
- Non è necessario usare con precisione le icone al posto giusto (e.g. una icona qualunque va bene purché la posizione sia corretta)

**Codice HTML (ed eventualmente CSS interno)**

**Codice CSS (solo se esterno)**

### Domanda #3 - JS + Framework (max 15 punti)

Creare un sistema di gestione delle chiamate di emergenza per la centrale dei vigili del fuoco. Usare un framework JS.

#### Endpoints:

- **GET** Info operatore:  
firedept.org/emergency/operators/:operatorId
- **GET** Lista chiamate:  
firedept.org/emergency/operators/:operatorId/calls
- **GET** Dettagli chiamata:  
firedept.org/emergency/operators/:operatorId/calls/:callId
- **POST** Segna chiamata come gestita:  
firedept.org/emergency/calls/:callId/handle
- **POST** Riapri chiamata:  
firedept.org/emergency/calls/:callId/reopen

#### Struttura Info Operatori/Operatrici:

```
{  
  "operatorId": "FD123456",  
  "name": "Lucia Bianchi",  
  "phoneNumber": "333555125",  
  "station": "Bologna Centrale",  
  "shift": "Notturmo"  
}
```

#### Struttura lista chiamate:

```
[  
  {  
    "callId": 1,  
    "description": "Incendio appartamento",  
    "priority": "Alta",  
    "handled": false,  
    "timestamp": "2025-03-15T18:30:00Z"  
  },  
  {  
    "callId": 2,  
    "description": "Incidente stradale",  
    "priority": "Media",  
    "handled": true,  
    "timestamp": "2025-03-15T20:00:00Z"  
  },  
  {  
    "callId": 3,
```

```
    "description": "Fuoriuscita di gas",
    "priority": "Alta",
    "handled": false,
    "timestamp": null
  }
]
```

#### Struttura Dettagli Chiamata:

```
{
  "callId": 2,
  "description": "Incidente stradale",
  "priority": "Media",
  "handled": true,
  "timestamp": "2025-03-15T20:00:00Z",
  "location": "Via Roma, 25",
  "unitsDispatched": 2
}
```

#### Funzionalità da implementare:

##### 1. Homepage (4 punti)

- Input ID operatore
- Lista chiamate in attesa (descrizione, priorità, orario)
- L'ordine della lista preimpostato è per ordine di urgenza
- Le chiamate possono essere ordinate in (1) ordine di urgenza (2) data e per numero di unità mandate (3). Un checkbox consente di visualizzare solamente quelle già gestite o quelle non ancora gestite ("handled": true/false)
- I campi null devono essere gestiti con testo in rosso per indicare che manca una informazione.

##### 2. Gestione Chiamate (4 punti)

- Bottone "Segna come gestita" per chiamate non ancora risolte
- Bottone "Riapri chiamata" per chiamate già gestite

##### 3. Vista Dettaglio (2 punti)

- Mostra dettagli chiamata
- Numero unità inviate
- Gestione stato chiamata (gestita/non gestita)

