Insegnamento di Tecnologie Web

CdS In Informatica

(A.A. 2023-24)

Esame scritto del 16/07/2024

Nome:

Cognome:

Matricola: `OBBLIGATORIA`

Corso di Studi: `e.g. Informatica`

Anno di frequenza: `e.g. 2023-24`

Attenzione:

- Questi computer sono limitati ad accedere solo ad alcuni siti: `eol.unibo.it`, `virtuale.unibo.it`, `developer.mozilla.org`, `getbootstrap.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: nesun erore di ortografia. Cuesta è un universita e non la squola elementale.*

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) CSS

Descrivere la differenza tra *canvas* e *viewport*.

b) Sicurezza Web

Descrivere una tecnica di configurazione degli header che può migliorare la sicurezza delle applicazioni web. Spiegare come questa tecnica può prevenire specifici tipi di attacchi e quali strumenti possono essere utilizzati per verificarne l'efficacia.

c) Mongoose Schema

Scrivere uno (o più) schema/schemata Mongoose che possa/no raccogliere tutte le informazioni della seguente descrizione: "Il 20 luglio, verrà lanciato sul mercato il nuovo smartphone 'TechMaster X100' di Innovatech. Il dispositivo, dotato di un display da 6.5 pollici e una batteria da 5000mAh, sarà disponibile in tre colori: nero, argento e blu. Il prezzo di lancio è fissato a 799 euro e sarà possibile acquistarlo online e nei negozi autorizzati. Ogni acquisto include una garanzia di 2 anni e il supporto clienti 24/7." NB: nel testo sono contenute circa 10 possibili informazioni strutturabili.

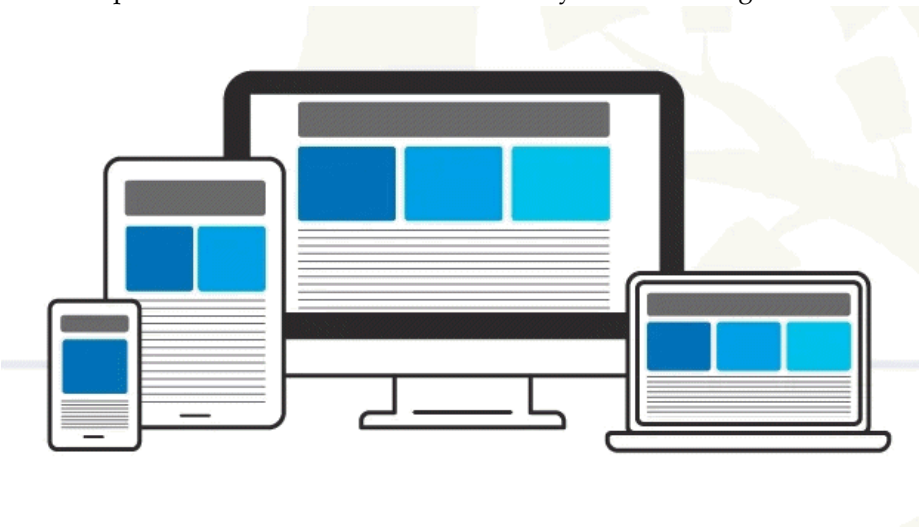
```
const mongoose = require('mongoose');
const Schema = mongoose.Schema;

const eventSchema = new Schema({
... lo schema qui
});

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

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

Riprodurre il seguente layout e comportamento responsive. Il punteggio massimo riflette precisione di forma, struttura e contenuti. La pagina DEVE essere responsive. Si notino le differenze tra i layout nell'immagine.



- È possibile (e consigliato) utilizzare Bootstrap

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

Codice CSS ***(solo se esterno)***

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

Creare una view per gestire degli appuntamenti medici, basato su uno schema che comprenda:

- nome del paziente
- età del paziente
- codice fiscale del paziente
- nome del medico
- data della visita
- luogo della visita
- motivo della visita
- costo

Il form per aggiungere un nuovo appuntamento medico deve prevedere la validazione del codice fiscale. Usare la seguente regex.

```
^([A-Z]{6}[0-9LMNPQRSTUV]{2}[ABCDEHLMPRST]{1}[0-9LMNPQRSTUV]{2}[A-Z]{1}
[0-9LMNPQRSTUV]{3}[A-Z]{1})$|([0-9]{11})$
```

N.B.: in questo file l'espressione regolare è stata spezzata su due righe per esigenze di paginazione. Usare una versione senza ritorni a capo.

Esempio di richiesta:

```
GET /api/appointments
```

Esempio di risposta:

```
{
  "success": true,
  "data": [
    {
      "id": "1",
      "patientName": "John Doe",
      "fiscal_code": "",
      "doctorName": "Dr. Smith",
      "appointmentDate": "2024-08-20T09:00:00.000Z",
      "reason": "General Checkup",
      "location": "Clinic A, Room 101"
    },
    {
      "id": "2",
      "patientName": "Jane Roe",
      "fiscal_code": "",
      "doctorName": "Dr. Adams",
      "appointmentDate": "2024-09-10T14:30:00.000Z",
      "reason": "Dental Cleaning",
      "location": "Dental Clinic, Room 202"
    }
```

```
  ]
}
```

Esempio di POST:

```
POST /api/appointments
Content-Type: application/json

{
  "patientName": "Alice Johnson",
  "fiscal_code": "",
  "doctorName": "Dr. Brown",
  "appointmentDate": "2024-09-15T11:00:00.000Z",
  "reason": "Eye Examination",
  "location": "Eye Clinic, Room 303"
}
```

Esempio di risposta:

```
{
  "success": true,
  "message": "Appointment added successfully",
  "data": {
    "id": "3",
    "patientName": "Alice Johnson",
    "doctorName": "Dr. Brown",
    "appointmentDate": "2024-09-15T11:00:00.000Z",
    "reason": "Eye Examination",
    "location": "Eye Clinic, Room 303"
  }
}
```

Usare un framework (e.g., Angular, React, Vue, etc.)

N.B.: jQuery NON E' considerato un framework, mentre web component lo è.

Esercizio: API

Descrizione:

Creare un'applicazione web per gestire un elenco di visite, basato su uno schema che comprende:

- Nome del paziente
- descrizione della visita
- data della visita
- giorni mancanti alla visita
- luogo della visita (a scelta tra domicilio, ambulatorio in via X, ambulatorio in via Y, etc.)

. L'applicazione deve permettere di aggiungere, visualizzare, modificare e cancellare le visite.

Requisiti obbligatori:

1. **Aggiungere Visita:**
   - Un form per inserire i dati della visita.
   - Un pulsante per salvare la visita.
2. **Visualizzare Eventi:**
   - Le visite mediche salvate devono essere visualizzate in un elenco.
   - Ogni visita nell'elenco deve mostrare nome del paziente e il motivo della visita.

Requisiti aggiuntivi:

3. **Cancellare Visita:**
   - Un pulsante per cancellare una o tutte le visite nell'elenco.
4. **Modificare Visita:**
   - Un pulsante per modificare una visita nell'elenco.
   - I campi devono essere precompilati con i dati della visita selezionata.

- *Suggerimento*: usare il metodo `test()` per validare il codice fiscale.