

---

## 1. Funzione `loadTimer()`

- **Descrizione:** La funzione `loadTimer()` si occupa di recuperare il tempo rimanente salvato nel `localStorage` del browser. Se è presente un valore salvato, viene utilizzato per inizializzare la variabile `timeLeft` (tempo rimanente). Se il valore non è presente, viene impostato un valore predefinito di 600 secondi (10 minuti) e salvato nel `localStorage`.
- **Utilizzo:** Questa funzione viene chiamata all'interno di `DOMContentLoaded` per caricare il timer salvato quando la pagina viene caricata.

## 2. Funzione `startTimer()`

- **Descrizione:** La funzione `startTimer()` avvia il countdown del timer, decrementando ogni secondo il valore di `timeLeft`. Ogni secondo, il tempo rimanente viene aggiornato e mostrato sull'interfaccia utente. Quando il timer arriva a zero, il timer viene fermato e vengono chiamate due funzioni: `showSubmissionMessage()` per visualizzare un messaggio di avviso e `submit()` per inviare automaticamente i dati.
- **Utilizzo:** Viene chiamata quando la pagina viene caricata (al termine del caricamento) per avviare il countdown del timer.

## 3. Funzione `showSubmissionMessage()`

- **Descrizione:** La funzione `showSubmissionMessage()` visualizza un messaggio di "Consegna effettuata" al centro dello schermo per un breve periodo di 3 secondi. Dopo 3 secondi, il messaggio viene nascosto.
- **Utilizzo:** Questa funzione viene invocata quando il timer scade per informare l'utente che il tempo è scaduto e che il test è stato consegnato.

## 4. Funzione `updateCharCount()`

- **Descrizione:** La funzione `updateCharCount()` aggiorna il conteggio dei caratteri della risposta scritta in un campo di testo per una domanda aperta. Ogni volta che l'utente scrive o modifica la risposta, il numero di caratteri viene visualizzato dinamicamente.
- **Utilizzo:** Questa funzione è chiamata quando una domanda aperta è selezionata e l'utente inizia a scrivere la propria risposta. Viene richiamata anche quando il campo di testo viene aggiornato.

## 5. Funzione `loadOpenQuestion()`

- **Descrizione:** La funzione `loadOpenQuestion(data)` carica i dati di una domanda aperta dal file `data.json`. Il file JSON contiene una lista di domande aperte, e questa funzione estrae la domanda corretta in base all'indice `q` presente nell'URL della pagina. Se la domanda è stata precedentemente salvata nel `localStorage`, viene caricata come risposta predefinita.
- **Utilizzo:** Viene utilizzata nella pagina delle domande aperte per caricare la domanda e la risposta preesistente (se presente) quando l'utente visita una determinata pagina delle domande aperte.

## 6. Funzione `saveOpenAnswer()`

- **Descrizione:** La funzione `saveOpenAnswer()` salva la risposta dell'utente per una domanda aperta nel `localStorage`. Ogni volta che l'utente scrive una risposta, questa viene memorizzata e aggiornata in tempo reale. Inoltre, la funzione aggiorna il conteggio dei caratteri tramite la funzione `updateCharCount()`.
- **Utilizzo:** Viene invocata ogni volta che l'utente inserisce una risposta nel campo di testo della domanda aperta.

## 7. Funzione `loadMultipleChoice()`

- **Descrizione:** La funzione `loadMultipleChoice(data)` carica le domande a scelta multipla dal file `data.json`. A seconda dell'index (numero della domanda), le domande vengono visualizzate sulla pagina, con le relative opzioni. Le risposte precedenti vengono recuperate dal `localStorage` per segnare le opzioni selezionate, se presenti.
- **Utilizzo:** Viene invocata nella pagina delle domande a scelta multipla per caricare le domande e le risposte selezionate in base all'indice di domanda.

## 8. Funzione `saveAnswer()`

- **Descrizione:** La funzione `saveAnswer(sectionIndex, questionIndex, answerIndex)` salva la risposta selezionata per una domanda a scelta multipla nel `localStorage`. La risposta dell'utente viene memorizzata con una chiave che include l'indice della sezione, della domanda e della risposta.
- **Utilizzo:** Viene invocata ogni volta che l'utente seleziona una risposta in una domanda a scelta multipla. La funzione salva l'opzione selezionata (con un indice da 0 a 3) nel `localStorage`.

## 9. Funzione `submit()`

- **Descrizione:** La funzione `submit()` si occupa di raccogliere tutte le risposte salvate nel `localStorage` (sia per le domande aperte che per quelle a scelta multipla). Le risposte vengono formattate in un file di testo (TXT), che viene poi scaricato automaticamente dall'utente. Una volta completato il download, il `localStorage` viene pulito e il timer viene fermato.

- **Utilizzo:** Viene chiamata quando il timer scade, per inviare le risposte e fare il download del file contenente le risposte dell'utente.

## 10. Funzione `loadPage()`

- **Descrizione:** La funzione `loadPage(page, questionIndex)` permette di navigare tra le diverse pagine del test. Modificando l'URL con il parametro `q` o `t`, si carica una pagina diversa con la domanda successiva o precedente.
- **Utilizzo:** Viene chiamata per navigare tra le pagine delle domande aperte e a scelta multipla.

## 11. Funzione `goForward()`

- **Descrizione:** La funzione `goForward()` consente di navigare in avanti tra le domande del test, passando da una domanda aperta alla domanda successiva o da una domanda a scelta multipla alla successiva. Se si raggiunge la fine delle domande di una sezione, la funzione naviga alla sezione successiva.
- **Utilizzo:** Viene chiamata quando l'utente vuole andare alla domanda successiva, a seconda della sezione e della tipologia di domanda (aperta o a scelta multipla).

## 12. Funzione `goBack()`

- **Descrizione:** La funzione `goBack()` consente di navigare indietro tra le domande, tornando alla domanda precedente o alla sezione precedente. Viene utilizzata per spostarsi indietro nelle domande, sia per le domande aperte che quelle a scelta multipla.
- **Utilizzo:** Viene invocata quando l'utente desidera tornare alla domanda precedente.

## 13. Funzione `goHome()`

- **Descrizione:** La funzione `goHome()` reindirizza l'utente alla pagina principale (`index.html`).
- **Utilizzo:** Viene chiamata per portare l'utente alla home page, ad esempio dopo aver completato il test.

---

## Conclusioni

Il codice gestisce un'applicazione di test con timer e salvataggio delle risposte. Ogni funzione è progettata per gestire un aspetto specifico dell'esperienza utente, inclusi il caricamento delle domande, il salvataggio delle risposte, la gestione del timer e la navigazione tra le domande. Il codice fornisce una base robusta per un'applicazione di test online che può essere facilmente estesa o modificata per includere nuove funzionalità o adattarsi a diverse esigenze.