

# TEHNICI WEB FORMS

*Claudia Chiriță . 2023/2024*

**CE ÎNVĂȚĂM SĂ FACEM AZI?**

# CLIENT-SIDE WEB STORAGE

- un obiect `Storage` este un array asociativ în care cheile și valorile sunt stringuri
- accesăm obiecte `Storage` prin proprietăți ale obiectului *window*:

```
localStorage // permanent  
sessionStorage // până la închiderea tabului
```

# STORAGE

```
localStorage.length // nr. intrări în Storage
localStorage.key(i) // numele cheii cu indexul i
localStorage.setItem(numCheie, numeValoare)
/* adaugă o cheie și valoarea ei sau înlocuiește
   valoarea unei chei existente */
localStorage.getItem(numCheie) // valoarea cheii
localStorage.removeItem("x") // șterge cheia din Storage
localStorage.clear() // șterge toate cheile
localStorage.propNoua = valoare
```

# CLIENT-SIDE WEB STORAGE

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="demo.css">
    <style>
      .brb {
        height: 350px;
        width: 350px;
        border-radius: 50%;
        display: inline-block;
        margin: 10px 0 10px 0;
        background-color: red;
      }
      div { text-align:center; }
    </style>
    <script>
      window.onload = function() {
        var buton =
document.getElementById("brb");
        buton.onclick= function() {
          let x =
Number(localStorage.getItem("nrc"));
          if (x) {
            localStorage.setItem("nrc", x +
1); }

```

>

# COOKIES?



# INCOGNITO?



# FORMS

elementul HTML `<form>` reprezintă o secțiune a unei pagini care conține elemente interactive (*controls*) și e folosit pentru a trimite informații către un server web

```
<form action="/my-handling-form-page" method="post">
<!-- poate conține orice element HTML, însă relevante sunt
      elementele de control -->
<!-- bad practice: imbricarea formularelor -->
</form>
```



# FORMS

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="demo.css">
  </head>
  <body>
    <div>
      <form action="/my-handling-form-page"
method="post">
        <ul>
          <li>
            <label for="name">Name:</label>
            <input type="text" id="name"
name="user_name" />
          </li>
          <li>
            <label for="mail">Email:</label>
```



- după completare, datele sunt trimise serverului ca o listă de perechi cheie/valoare într-un request HTTP de tip post

# FORMS: CONTROLS

```
<form id="myform">

<!-- elemente specifice formularelor -->
<input>
<textarea>
<fieldset>
<select>
<button type="submit"> Submit </button>

</form>

<input type="submit" form="myform" value="Submit">
<button type="submit" form="myform"> Submit </button>
```

# STRUCTURARE: LABEL

folosit pentru etichetarea widget-urilor

```
<label for="name">Name:</label>  
<input type="text" id="name" name="user_name">
```

sau

```
<label for="name">  
  Name: <input type="text" id="name" name="user_name">  
</label>
```

# STRUCTURARE: LABEL

etichetare **multiplă** a unui widget

```
<div>  
  <label for="name">Name:  
    <span aria-label="required">*</span>  
  </label>  
  <input type="text" id="name" name="user_name">  
</div>
```

# STRUCTURARE: FIELDSET

folosit pentru gruparea elementelor într-un form

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="demo.css">
  </head>
  <body>
    <div>
      <form>
        <fieldset>
          <legend>Cookies cu</legend>
          <p>
            <input type="radio" name="ingredient"
id="ingred_1" value="stafide" />
            <label for="ingred_1">Stafide</label>
          </p>
          <p>
            <input type="radio" name="ingredient"
```



# INPUT

```
<input type="..." name="..." id="...">
```

- poate lua multe forme, în funcție de valoarea atributului *type*
- pentru crearea majorității tipurilor de widget-uri: câmpuri de text pe o singură linie, controale pentru timp & dată, checkboxuri, butoane radio, selectoare de culoare, butoane

# TEXT INPUT

```
<input type="text" id="comment" name="comment" value="câmp tex
```

- attribute specifice: readonly, placeholder, required, size (width pentru câmp), maxlength (numărul maxim de caractere)

# TEXT INPUT

text input cu pattern:

```
<label for="cnp">  
  <input type="text" id="cnp" pattern="[0-9]{13}">  
  CNP</label>
```

```
<label for="ci">  
  <input type="text" id="ci" pattern="[A-Z]{2}[0-9]{4}">  
  C.I  
</label>
```



# TEXT INPUT

- password type

```
<input type="password" id="pwd" name="pwd" maxlength="8">
```

- hidden type: control invizibil pentru user; este trimis serverului odată cu restul formularului

```
<input type="hidden" id="timestamp" name="timestamp" value="1286
```

# EMAIL

```
<input type="email" id="email" name="email" multiple="" require
```

- are validare client-side built-in
- atributul *multiple* permite scrierea mai multor adrese separate prin virgulă

# SEARCH

```
<input type="search" id="search" name="search">
```

- principala diferență față de input text este stilul:  
buton X de ștergere a conținutului, colțuri rotunjite

# CÂMP NUMERIC

```
<input type="number" name="age" id="age" min="1" max="10" step
```

- acceptă doar numere floating-point
- attribute: min, max, step (default 1, acceptă "any")

# SLIDER

```
<input type="range" name="price" id="price" min="5" max="500"
```

- se recomandă setarea atributelor: min, max, step

# FILE SELECTOR

```
<input type="file" name="file" id="file" accept="image/*" mult
```

- dispozitive mobile: selectorul de fișiere poate accesa fotografii, video și audio înregistrate direct de cameră și microfon

```
<input type="file" accept="image/*;capture=camera">
```

# RADIO

```
<input type="radio" name="ingredient" id="ingred_1" value="sta">  
<label for="ingred_1">Stafide</label>  
  
<input type="radio" name="ingredient" id="ingred_2" value="chi">  
<label for="ingred_2">Choc chips</label>
```

- doar unul din butoanele radio poate fi selectat;  
atributul *name* trebuie să aibă aceeași valoare  
pentru toate opțiunile

# CHECKBOX

```
<input type="checkbox" id="cookie" name="cookie" checked="">  
<label for="cookie">Cookies</label>
```

```
<input type="checkbox" id="biscuit" name="biscuit">  
<label for="biscuit">Biscuits</label>
```

- permit selectarea mai multor valori, spre deosebire de butoanele radio



# HTML5 INPUT

- **color**: alegerea unei culori
- **date** : introducerea datei în formatul yyyy-mm-dd
- **time** : introducerea orei în formatul hh:mm
- **week** : introducerea datei în formatul yyyy-Www

# HTML5 INPUT

- **image**: buton grafic folosind o imagine
- **tel** : introducerea unui număr de telefon; fără validare implicită
- **url** : introducerea unui URL în formatul `urlscheme://restofurl`

# TEXTAREA

```
<textarea name="comments" rows="10" cols="48" wrap="hard" resi
```

- câmp de editare a textului pe mai multe linii
- implicit: cols = 20

# SELECT

```
<select id="ingrediente" name="ingrediente">  
  <option>stafide</option>  
  <option selected="">choc chips</option>  
  <option>afine</option>  
</select>
```

- control care oferă un meniu de opțiuni selectabile

# TRIMITEREA FORMULARELOR: INPUT

- **submit**: trimiterea datelor din formular către server

```
<input type="submit" value="Send Request">
```

- **reset**: resetarea tuturor câmpurilor din formular (rar folosit)

```
<input type="reset" value="Reset the form">
```

# TRIMITEREA FORMULARELOR: BUTTON

- submit: trimiterea datelor din formular către server

```
<button type="submit" value="Send Request" />
```

- reset: resetarea tuturor câmpurilor din formular

```
<button type="reset" value="Reset the form" />
```

- poate fi folosit în afara formularelor cu atributul

```
type="button"
```

# TRIMITEREA FORMULARELOR: ATRIBUTE

```
<form target="_blank" method="post" action="URL">
```

```
target = "_self" / "_blank" / "numeiframe"  
/* locul în care este afișat răspunsul după  
   trimiterea formularului (implicit: _self) */
```

```
method = "post"/"get"  
/* metoda folosită pentru comunicarea cu serverul  
   (implicit: get)  
   get: query string în URL;  
   post: query string în corpul requestului */
```

```
action = URL-ul care procesează formularul  
/* (implicit: URL paginii curente) */
```

# FORMS

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="demo.css">
  </head>
  <body>
    <div>
      <form action="" method="post"
target="numeiframe">
        <ul>
          <li>
            <label for="name">Name:</label>
            <input type="text" id="name"
name="user_name" />
          </li>
          <li>
            <label for="mail">Email:</label>
```





**VALIDARE FORMULAR BUILT-IN**

**VALIDARE FORMULAR: E REZONABIL?**

# FORMS & JAVASCRIPT

- un form este reprezentat în JavaScript de un obiect *HTMLFormElement*
- proprietatea **elements** întoarce colecția câmpurilor formularului (array-like object)
- proprietatea *length* întoarce numărul de câmpuri

# FORMS & JAVASCRIPT

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="demo.css">
    <script>
      /* window.onload=function() {
        var f =
document.getElementById("formular");
        alert(f.elements);
//[objectHTMLFormControlsCollection]
        alert(f.length); // 4
        alert(f.elements[2].value); // I love
cookies!
      } */
    </script>
  </head>
  <body>
```



# PROPRIETATEA VALUE

- pentru `<input type="text">` și `<textarea>` reprezintă valoarea introdusă de utilizator la trimiterea formularului și este un String
- pentru `<option>` reprezintă valoarea opțiunii alese
- altfel, reprezintă valoarea atributului html *value*

```
<input type="text" value="5">  
<input type="submit" value="Trimite">  
<input type="radio" value="v1">  
<input type="checkbox" value="v1">
```

## EVENIMENTE FORM

- **submit**: evenimentul lansat la trimiterea formularului
- **reset**: evenimentul lansat când un formular este resetat

# EVENIMENTE FORM

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="demo.css">
    <script>
      /* window.onload = function() {

document.getElementById("read").onsubmit =
suma;
      }
      function suma() {
        let x =
document.getElementById("nr").value;
        let s = 0;
        for (let i = 1; i <= parseInt(x);
i++)
          s = s + i;
```



# PROPRIETĂȚI SPECIFICE CÂMPURILOR

- proprietatea *checked* (boolean) pentru elemente

```
<input type="radio" value="v1">  
<input type="checkbox" value="v1">
```

- proprietatea *elements* a câmpului *fieldset*:  
array-like care îi conține componentele



# PROPRIETĂȚI SPECIFICE CÂMPURILOR

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="demo.css">
    <script>
      /* window.onload = function() {

document.getElementById("quiz").onsubmit =
checkQuiz;
    alert("dd");
  }
  function checkQuiz() {
    let fe =
document.getElementById("quiz").elements;
    let q1 = fe[0].elements;
    let q2 = fe[4].elements;
    let v = 0;
```



# EVENIMENTE CONTROALE: INPUT

- pentru elemente cu conținut editabil  
ex: <input>, <textarea>
- se declanșează la fiecare modificare  
ex: fiecare caracter scris într-un input

# EVENIMENTE CONTROALE: CHANGE

- pentru `<input>`, `<textarea>`, `<select>`
- se declanșează după alegerea unei noi valori
- pentru `<input type="text">` și `<textarea>` se declanșează când încetează focusul pe element

# EVENIMENTE CONTROALE: CHANGE

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="demo.css">
    <script>
      /* window.onload = function() {

document.getElementById("tt").onchange =
function () {

alert(document.getElementById("tt").value); }
      } */
    </script>
  </head>
  <body>
    <div>
      <input type="text" id="tt" value="demo" />
    </div>
  </body>
</html>
```

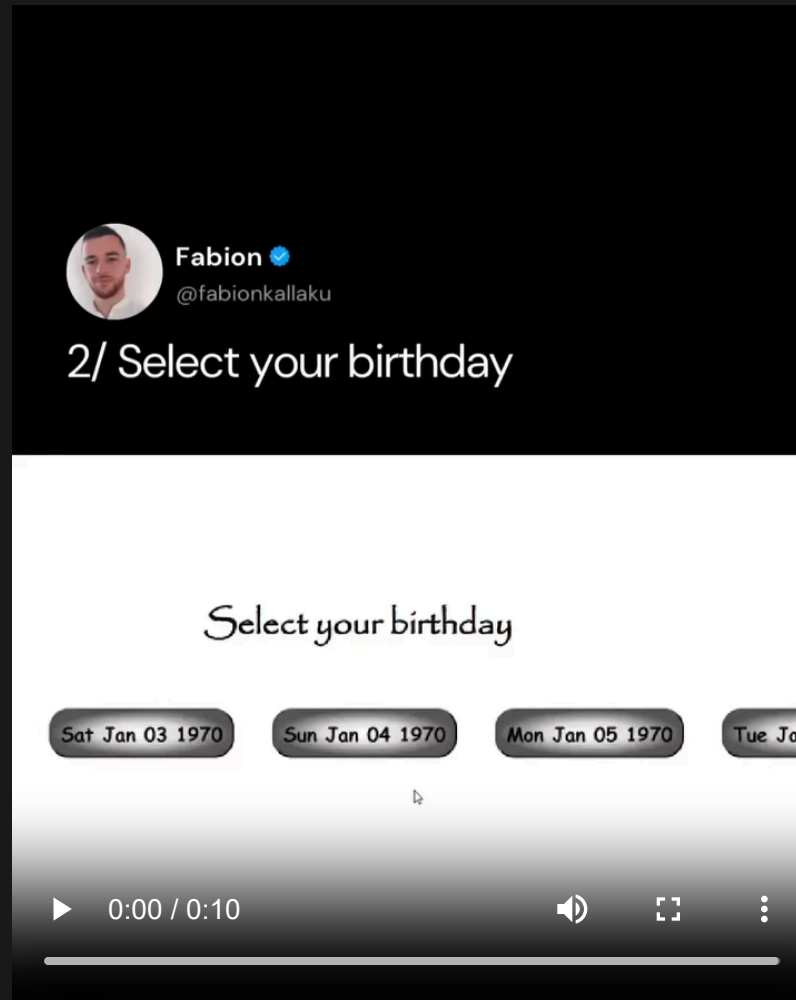
>

# EVENIMENTE CONTROALE: CHANGE

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="demo.css">
    <script>
      /* window.onload = function() {
        obp =
document.getElementById("hello");
        obsel =
document.getElementById("sel");
        obsel.onchange = function () {
          obp.innerHTML +=
obsel.options[obsel.selectedIndex].value;
        }
      } */
    </script>
  </head>
```


>

# AI-AI, BAD UI!



# AI-AI, BAD UI!



Fabion   
@fabionkallaku

3/ Please enter your phone  
number:

## Bad Phone Selector

Phone: (14) 419-642-857



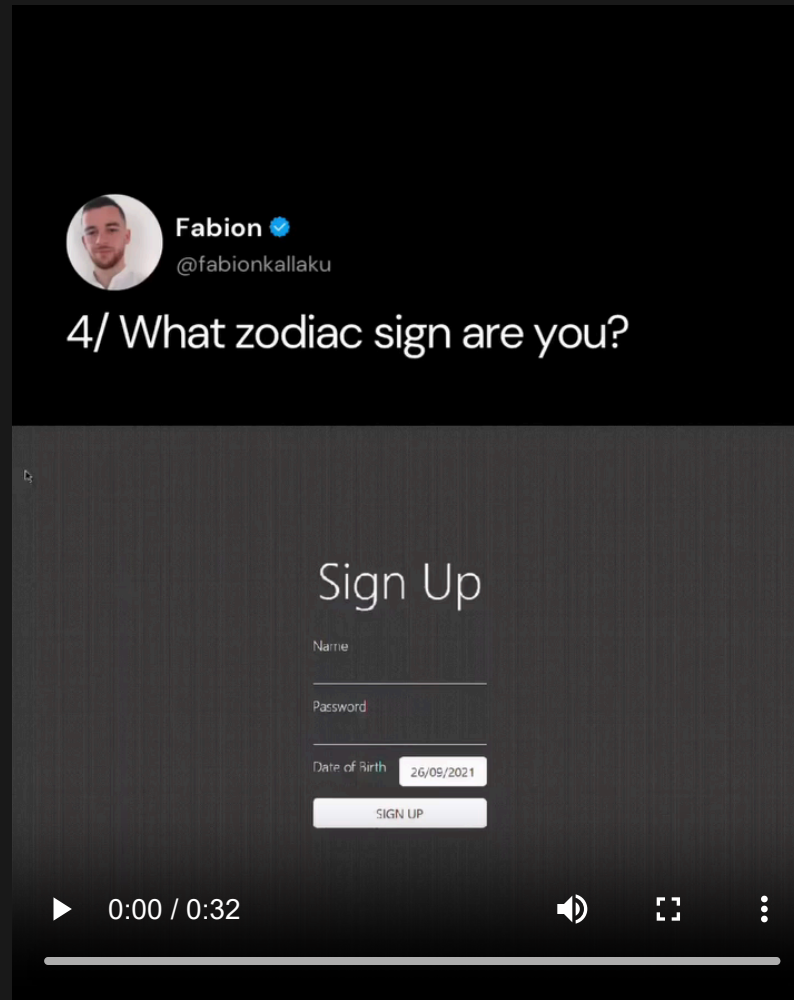
☐ Advanced Mode  
Submit



0:00 / 0:12

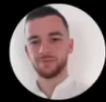



# AI-AI, BAD UI!



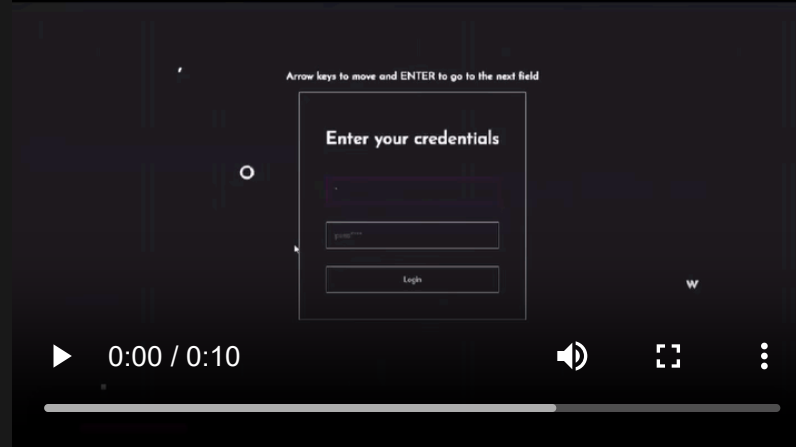


# AI-AI, BAD UI!




Fabion   
@fabionkallaku

5/ Please enter your credentials

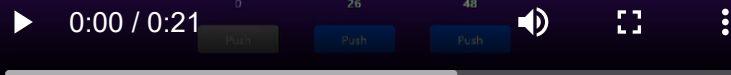
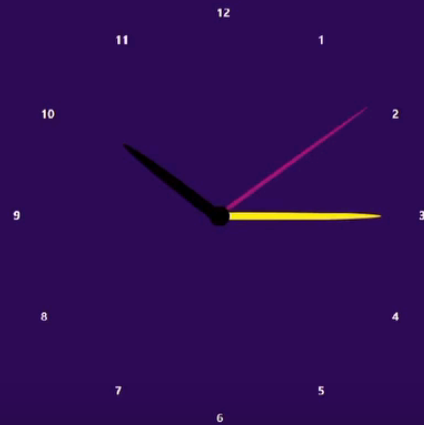


# AI-AI, BAD UI!

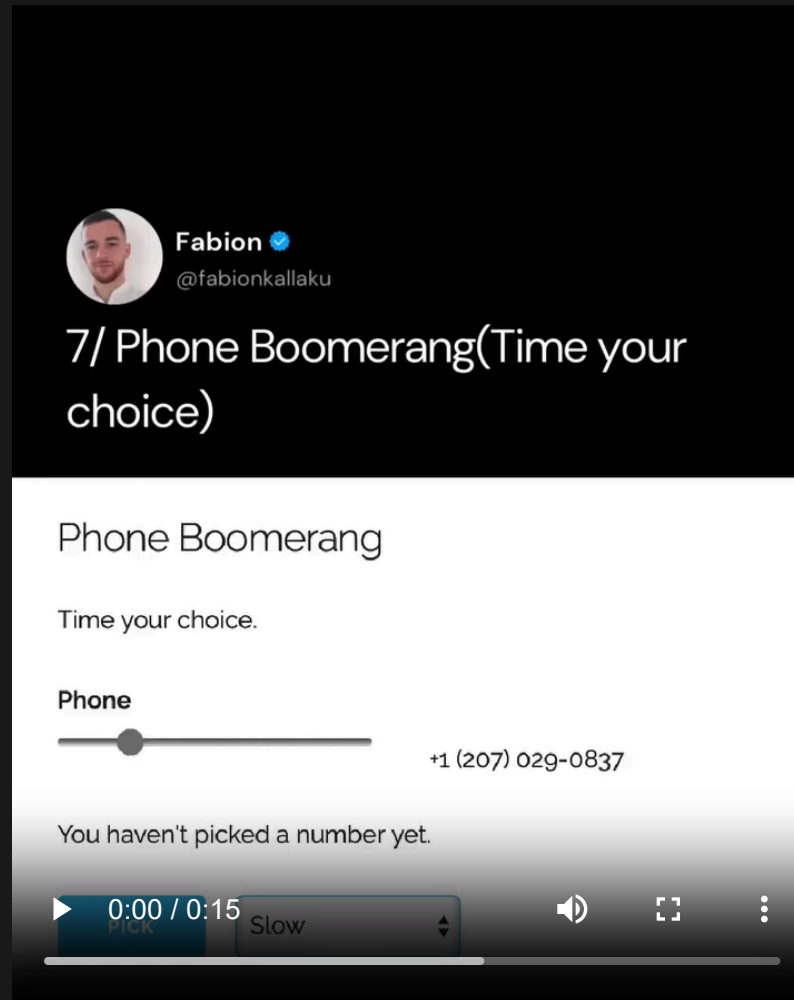


Fabion   
@fabionkallaku

6/ Full analog clock



# AI-AI, BAD UI!



**AI-AI, BAD UI!**

```

      WW
    /  _  \
   |  oo  |
  (| _ ( ) _ |) / _ oo o _ \
W"          "W  \ _ / (| _ o _ |)
W  -=====  W  / | \ / | \ \ \ _ / /
    "W          W" | | | | | | | /-----
W" " " " " " " " "W | | | | | | | ===== |
W                      W | | | | | | | ===== |

```

întrebări?

