***Proiect “Planner”***

1. Cerinte HTML:
   1. Proicetul contine cele 2 pagini html:

“login.html” si “register.html”

Aceste două pagini contin formularele de Logare si Inregistrare, “login.html” contine un script javascript inline care are rolul de a arat parola introdusa cand utilizatorul vine cu curosr-ul pe el si revine la forma de parola cand iese.

“register.html” contine un script javascript mai complex, care este localizat in “Planner/JS/rjs.js”

Are rolul ca si scriptul de la logare dar in plus aici mai exista functia “clearonpaste()” care face ca input-ul sa fie dezactivat si reactivate imediat cand user-ul vrea sa lipeste ceva, ca sa prevenim injectarea unor comenzi SQL la inserarea in bazei de date, de asemenea nici unele caractere nu se pot tasta in inputuri.

Cu toate ca javascript este un limbaj pe partea de client, se poate modifica de client, inputul se poate valida si cu php, verificand fiecare input daca contine anumite caractere sau nu. Nu am facut aasa fiindca Am avut nevoie de 2 pagini html, altfel n-as fi avut niciunul.

1. Cerinte CSS:
   1. Stilizarea este facuta in fisiere separate, link-ul catre aceste fisiere se afla in <head>-ul paginii.

Sunt 4 fisiere css, 3 in “Planner/CSS/” si unul in “Planner/ToDo/”.

Cele 3 fisiere css din “Planner/CSS/” sunt separate folosite:

“css.css” – “index.php”

“login.css” - “login.html”

“register.css – “register.html”

* 1. Site-ul este responsive pe cat am reusit sa-l fac responsive cu @media in CSS.
  2. Layout-ul este realizat cu ajutorul cadrelor, nu am vazut rostul sa folosesc tabele(div, select, form, input, h2, p, ul, li)
  3. Proiectul nu contine un menu-dropdown, fiindca nu am avut unde si pentru ce sa-il pun. Exista doar <select>.
  4. Nu exista transformari de genul:

“transform: rotateY(50deg)”

1. Cerinte Javascript:
   1. Stilul unui element este modificat de fiecare data un utilizator selecteaz a data pe pagina “Planner/index.php”.
   2. O functie este apelat de fiecare data utilizatorul incearca sa introduca un character pe pagina “Planner/register.php”.
   3. Evenimentul de mouseover este utilizat pe paginile “Planner/login.php” si “Planner/register.php” cand se trece peste zona de “Password” in formulare.
   4. Modificare dynamica al unui element daca consideram, ca de fiecare data cand utilizatorul alege o alta luna, unele elemente(zilele(1-31)) sunt reinnoita si actualizate, modificand pozitia de obicei. Data selectata din formularul in “Planner/index.php” se modifica de fiecare data utilizatorul da click pe o data.
   5. setTimeout este utilizat o singura data in “js.js”, pentru ceasul digital de pe pagina “Planner/index.php”
2. Cerinte PHP/MySQL:
   1. Cele 2 formulare se regsesc in “login.html” si “register.html”, functionalitate acestora in “loginuser.php” si “registeruser.php”

Am lasat si “login.php” si “register.php”, cu toate ca nu contin elemente php, astfel am si cu extensia html.

* 1. O singura baza de date este creat automat daca nu exista sub Numele “plannerbybeni” si 2 tabele (“personcredentials” si “usertodo”) ambele au rol diferit.

Tabelul “personcredentials” contine 6 coloane, PersonID(int) care se auto\_incrementeaza de fiecare data un user se inregistreaza, un LastName, FirstName, username, email si pswrd(parola) care sunt stringuri(255 char max).

Tabelul “usertodo” contine 4 coloane, text(string(2000 char max)), username(string 255 char max), month(int), day(int).

* 1. Continutul dinamic este generat doar dupa ce un utilizator s-a inregistrat, a logat, a selectat o data si a planificat pe data selectata.

***Descrierea functionalitatilor***

Pagina principala este “Planner/index.php”, care nu se incarca daca utilizatorul nu este inregistrat/logat, altfel se afiseaza un mesaj cu informația ca trebuie sa fie logat ca sa foloseasca site-ul, si link-uri catre pagina de inregistrarea si logare.

Inregistrarea se face astfel, dupa completarea formularului, se redirectioneaza catre “registeruser.php” unde se valideaza formularul, dupa care se introduce in tabela “personcredentials” datele utilizatorului. Toate acestea doar cand pagina a fost accesat prin submiterea formularului.

Functia createTables() din “registeruser.php” realizeaza o conectiune cu baza de date si creaza tabelele daca acestea nu exista.

Daca inregistrarea a fost realizat cu succes, utilizatorul este redirectionat pe pagina “login.html”.

Dupa conectare utilizatorul este redirectionat pe pagina “Planner/index.php”, unde de data aceasta apare pagina cu toate functionalitatea lui.

De fiecare data cand un utilizator intra pe pagina, se actualizeaza elementele din tabela “usertodo”, adica sterge elementele care sunt din trecut si nu se mai pot accesa. (“removeOld.php”)

Pe top-ul paginii apare un ceas digital, care este realizat cu ajutorul functiei showTime() din “Planner/JS/js.js” unde am folosit de functionalitatile javascript-ului cu date ca sa afisez ora , minutul si secunda curenta.

Dupa care apare o salutare si cu Numel si Prenumel dat de utilizator la inregistrarea(First Name, Last Name).

Sub el este un formular simplu cu un buton care distruge sesiunea curenta si redirectioneaza utilizatorul pe pagina “login.html”.

Sub acest formular apare un <div> cu numele zilelor saptamanii, sub care vin cele “n” zile din luna care suntem, zilele trecute si lunile nu se pot accesa, astfel au o culoarea diferita si nu exista evenimente cand utilizatorul da click pe ele. Aceste zile sunt reinnoite de fiecare data utilizatorul schimba luna sau chiar anul.

De fiecare data cand utilizatorul schimba luna, functia “daysReset()” este apelat unde se reinnoiesc zilele.

De fiecare data cand utilizatorul schimba anul, functia “yearsReset()” este apelat, unde se creeaza calendarul de anul selectat si apoi se apeleaza functia “daysReset()”.

Cand utilizatorul selecteaza o zi ( da click pe o zi ), această zi se va primi o nume de clasa noua, si se va schimba culoarea fundalului elementului, apoi se trece data in formularul de mai jos.

Dupa ce utilizatorul a ales data pe care vrea sa faca planuri, poate sa dea click pe butonul “Plan” in partea de jos a paginii, de unde trece la ultimul site “/ToDo/index.php”.

Aici utilizatorul este informat pe ce data o sa faca planurile, are posibilitatea sa aleaga o alta data, sau chiar sa deconecteze, sau evident, ca sa faca planurile.

Utilizatorul introduce textul / planul in zona de textarea, dupa care da click pe Add, care transmite textul, username-ul la “add.php” unde se introduce datele in tabela “usertodo”, textul, usernameul, luna si ziua adaugarii, dupa care se redirectioneaza inapoi pe site cu datele actualizate.

Sub zona de text se afla o lista neordonata in care apar planurile utilizatorului pe data selectata.

Si in ultimul rand pe partea de jos a paginii exista butonul “Remove Plans” care sterge planurile pe data selectata.

***Technologii utilizate: HTML, CSS, JS, PHP, MYSQL, XAMPP***