

Akademija tehničko-umetničkih strukovnih studija Beograd Odsek Visoka ICT škola

Dokumentacija za sajt

<http://berix-agency.epizy.com/>

Predmet: Praktikum PHP

Student: Aleksa Berisavac 32/18

Sadrzaj

[1. Sadrzaj 5](#_Toc111583245)

[2. Slike stranica i opis 6](#_Toc111583246)

[2.1 Home stranica 6](#_Toc111583247)

[2.2 Reservations 7](#_Toc111583248)

[2.3 Make a Reservation 8](#_Toc111583249)

[2.4 Admin Panel 9](#_Toc111583250)

[2.5 Login 10](#_Toc111583251)

[2.6 Register 11](#_Toc111583252)

[2.7 Author 12](#_Toc111583253)

[3. Dizajn Baze Podataka 13](#_Toc111583254)

[4. Struktura Projekta 14](#_Toc111583255)

[5. Svi Kodovi (moguce da ima razlike u odnosu na hostovanu verziju sajta) 15](#_Toc111583256)

[5.1 assets 15](#_Toc111583257)

[5.1.1 sass 15](#_Toc111583258)

[5.1.2 main.js 37](#_Toc111583259)

[5.2 config 82](#_Toc111583260)

[5.2.1 .env 82](#_Toc111583261)

[5.2.2 config.php 83](#_Toc111583262)

[5.2.3 connection.php 84](#_Toc111583263)

[5.3 data (sadrzi i bazu podataka kao backup, za svaki slucaj) 85](#_Toc111583264)

[5.3.1 logAccessedPages.txt (napisacu samo par linija) 85](#_Toc111583265)

[5.3.2 logSuccessfullLogin.php (napisacu samo par linija) 86](#_Toc111583266)

[5.3.3 logUnsuccessfullLogin (napisacu samo par linija) 87](#_Toc111583267)

[5.4 models 88](#_Toc111583268)

[5.4.1 contact.php 88](#_Toc111583269)

[5.4.2 deleteRow.php 90](#_Toc111583270)

[5.4.3 functions.php 91](#_Toc111583271)

[5.4.4 getActivities.php 103](#_Toc111583272)

[5.4.5 getDestinationInformations.php 104](#_Toc111583273)

[5.4.6 getDestinations.php 105](#_Toc111583274)

[5.4.7 getDestinationsIdName.php 106](#_Toc111583275)

[5.4.8 getSpecificDestination.php 107](#_Toc111583276)

[5.4.9 getTable.php 108](#_Toc111583277)

[5.4.10 insertActivity.php 109](#_Toc111583278)

[5.4.11 insertDestination.php 111](#_Toc111583279)

[5.4.12 insertHotel.php 113](#_Toc111583280)

[5.4.13 insertReservation.php 115](#_Toc111583281)

[5.4.14 insertTravelDate.php 117](#_Toc111583282)

[5.4.15 insertTravelPrice.php 119](#_Toc111583283)

[5.4.16 login.php 121](#_Toc111583284)

[5.4.17 logout.php 123](#_Toc111583285)

[5.4.18 register.php 124](#_Toc111583286)

[5.4.19 searchAdminPanel.php 126](#_Toc111583287)

[5.4.20 showPagesActivity.php 127](#_Toc111583288)

[5.4.21 uploadActivityPicture.php 128](#_Toc111583289)

[5.4.22 uploadDestinationPicture.php 130](#_Toc111583290)

[5.5 views 132](#_Toc111583291)

[5.5.1 fixed 132](#_Toc111583292)

[5.5.1.1 registration 132](#_Toc111583293)

[5.5.1.2 serverCommunication 134](#_Toc111583294)

[5.5.1.3 footer.php 136](#_Toc111583295)

[5.5.1.4 head.php 137](#_Toc111583296)

[5.5.1.5 header.php 138](#_Toc111583297)

[5.5.1.6 navigation.php 139](#_Toc111583298)

[5.5.1.7 scripts.php 141](#_Toc111583299)

[5.5.2 pages 142](#_Toc111583300)

[5.5.2.1 admin.php 142](#_Toc111583301)

[5.5.2.2 author.php 145](#_Toc111583302)

[5.5.2.3 index.php 146](#_Toc111583303)

[5.5.2.4 makeAReservation.php 148](#_Toc111583304)

[5.5.2.5 showReservations.php 150](#_Toc111583305)

[5.5.3 partials 151](#_Toc111583306)

[5.5.3.1 destination.php 151](#_Toc111583307)

[5.5.3.2 formInputElement.php 152](#_Toc111583308)

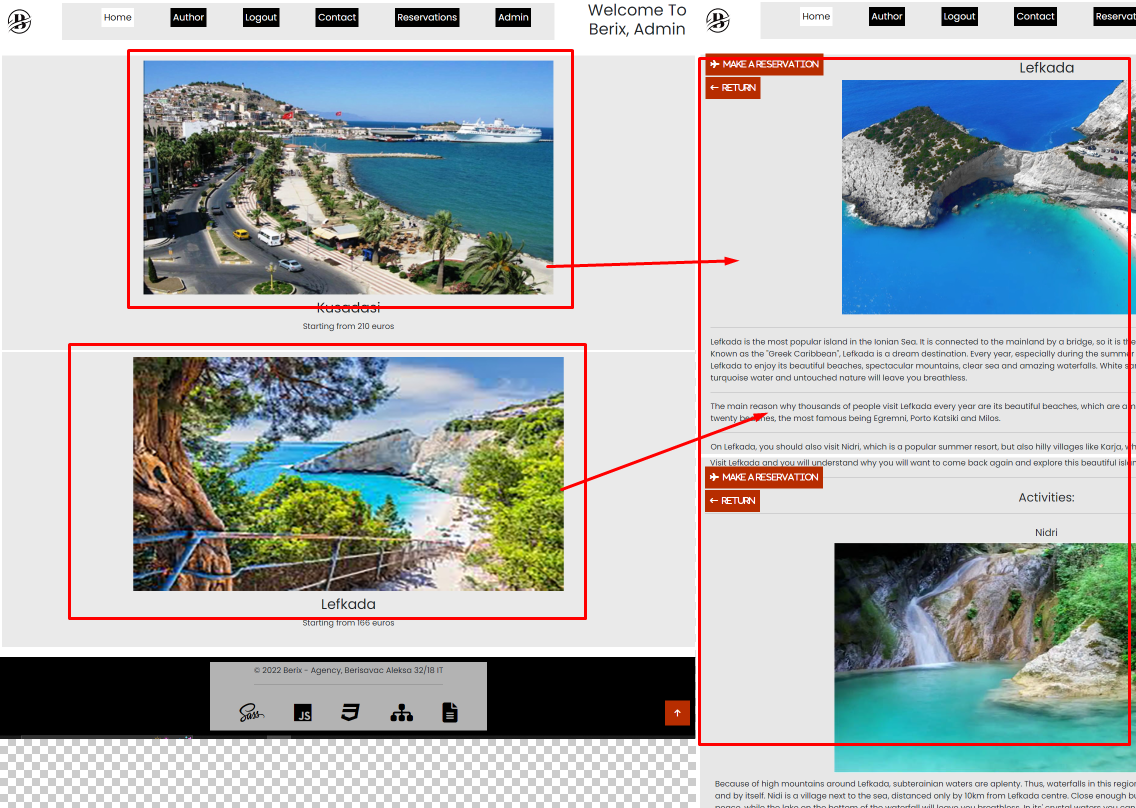
[5.6 index.php 153](#_Toc111583309)

# Sadrzaj

Sajt simulira jednu turisticku agenciju. Sto se tice korisnicke strane, mogu se videti destinacije, moze se uci u njih i videti njihovi detalji. Odavde, kao i sa stranice gde se prikazuju rezervacije koje dati korisnik vec ima, moze se otici na stranu za rezervaciju. Moze se obaviti rezervacija. Sve provere se rade na backu kako se podaci koji se koriste za ispis ovih polja dobijaju iz backa. Nisu odradjene provere koje proveravaju da li taj korisnik vec ima rezervaciju ciji se vremenski interval preklapa sa trenutnim. Postojanje podataka i njihova ispravnost se proverava. Na stranici za prikaz rezervacija korisnika korisnik moze uklanjati svoje rezervacije. Sto se tice administratora, on na admin panelu ima mogucnost potpunog upravljanja sadrzajem bitnih tabela baze podataka. Odradjen je select, delete, insert. Update nije odradjen posto ima previse tabela i moze doci do problema sa referencama, ali moguce je obrisati red (ukoliko se ne krsi referencijalni integritet, u kom slucaju potrebno je rucno brisati sve redove iz drugih tabela koji se referenciraju) i ubaciti novi red. Moguce je pretrazivati tabelu po apsolutno bilo kojoj koloni. Zahtev za pretragu se salje bazi, jer je ona optimizovanija za to. Takodje se ovde mogu videti statisticki podaci posecivanja stranice, i koliko je korisnika ulogovano u zadnjih 24h. Dosta ispisa se radi preko ajaxa slanjem zahteva na back, posto je za mnoge ispise potrebno dohvatati podatke iz odredjenih tabela.

# Slike stranica i opis

## Home stranica



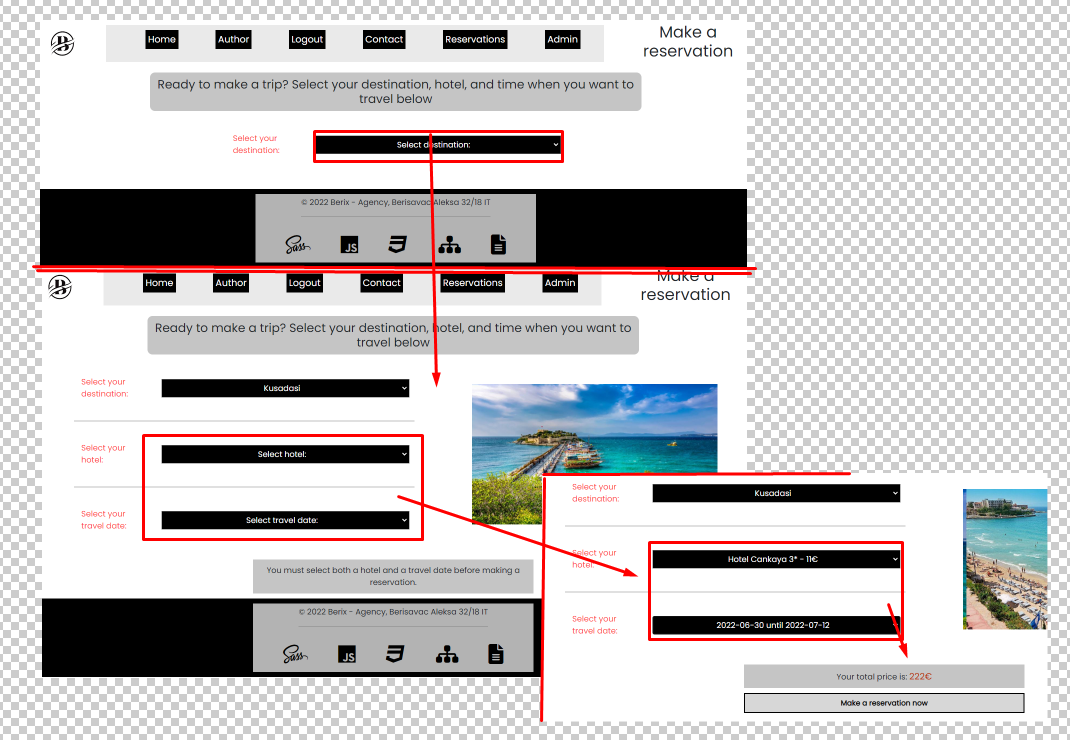
Klikom na neku od destinacija obavlja se ispis javaskriptom sa podacima koji su dohvaceni sa backa. Sve je dinamicki u zavisnosti od same destinacije i njenog broja aktivnosti, slika destinacije, slika aktivnosti etc. Odavde je moguce otici na stranicu za obavljanje rezervacije, ili se vratiti i ponovno ispisati sve destinacije. Naravno, sto se tice samog nav bara, ovde se ispisuje „Reservations“ jer sam ulogovan, inace ne bi, takodje se ispisuje „Admin“ jer sam ulogovan na admin nalogu, inace ne bi. Takodje na samom ulazu u stranicu preko url-a proverava se ponovo sve i redirektuje se, tako da je na admin stranicu nemoguce uci ako nismo u sesiji. Takodje se u headeru ispisuje dobrodosao [ime korisnika], ali prikazuje se samo na index stranici. Inace je drugaciji text headera u zavisnosti od stranice u kojoj se nalazimo. Naravno, prikaz detalja o selektovanoj destinaciji se menja u zavisnosti od toga da li smo ulogovani. Tacnije, dugme koje vodi na obavljanje rezervacije ce voditi na login ako nismo ulogovani, i promenice se njegov opis iz „Make a reservation“ u to da korisnik mora biti ulogovan.

## Reservations



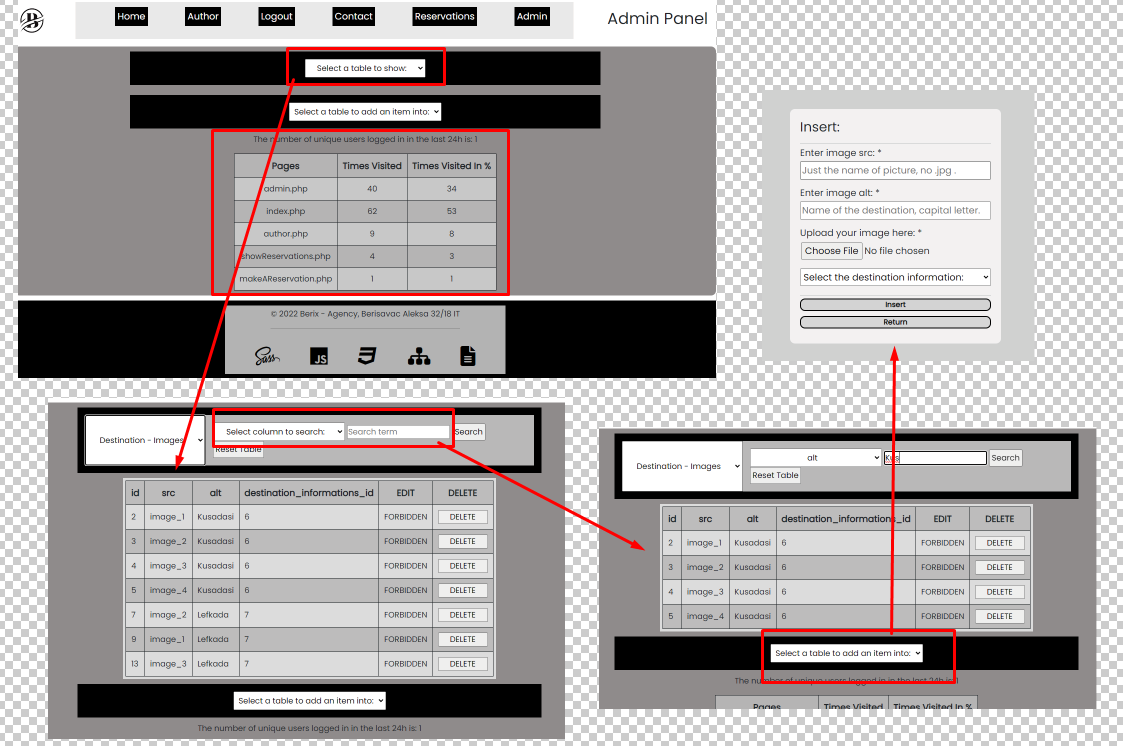
Ovde korisnik moze videti sve rezervacije koje je napravio. Ispisuje se dinamicki iz tabele. Moze i da brise svoje rezervacije ako zeli. Zabranjen mu je update svojih rezervacija, a sa obzirom da sam iskoristio istu funkciju za ispis tabele kao i na admin panelu nisam video potrebu da ponovo pisem funkciju za ispis tabele bez ovog dela. Odavde se moze otici na stranicu za obavljanje rezervacije.

## Make a Reservation



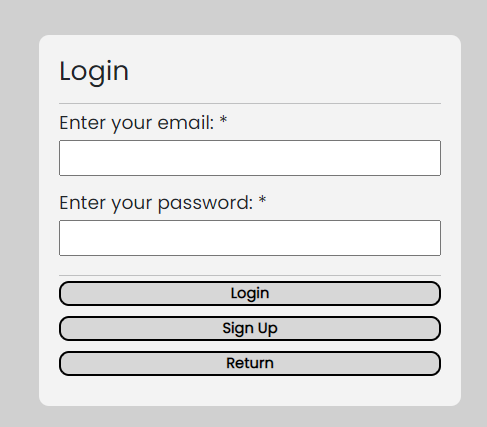
Prvo se prikazuju destinacije koje postoje. Nakon sto se izabere destinacija ispisuju se select za hotele i select za datum putovanja. Nakon sto se oba izaberu ispisuje se ukupna cena i dugme za rezervaciju. U slucaju da se nesto od-Izabere, u zavisnosti od toga sta je odredjene stvari ce se ispisati. Sve je odradjeno preko javaskripta i ajaxa u komunikaciji sa bazom podataka. Ako dodje do greske ispisace se koja je greska bila na serveru, odmah ispod dugmeta za obavljanje rezervacije.

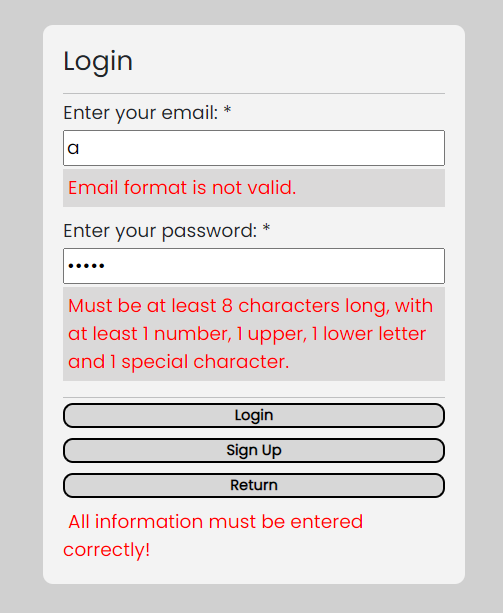
## Admin Panel



Ovde Administrator inicijalno vidi prikaz statistike i ima opcije da prikaze bilo koju od tabela iz baze, da insertuje u jednu od 7 bitnih tabela, i da brise iz bilo koje od tabela. U slucaju da nema redova u tabeli, ili da dodje do greske u toku brisanja zbog narusavanja referencijalnog integriteta ispisace se poruka na dnu ekrana. 2 tabele u koje se moze raditi insert sadrze i insert slike. Sa obzirom na to da je tema takva kakva jeste nemam nikakve potrebe za thumbnailovima, pa njih nisam ni radio. Ali slika se sasvim uredno moze postaviti, i koristiti kasnije kada se poveze sa odgovarajucim aktivnostima etc, sto je sve lepo ispunjeno u insert formi, koja je posebna, kao popup, i dinamicki se ispisuje u zavisnosti od tabele u koju se insertuje. Sve provere sto se tice samog inserta se rade i na frontu i na backu. Nakon sto dodje do izmene tabele, ukoliko je bas ta tabela prikazana ponovo ce se ispisati kako bi se uocila promena. Bilo to insert ili delete.

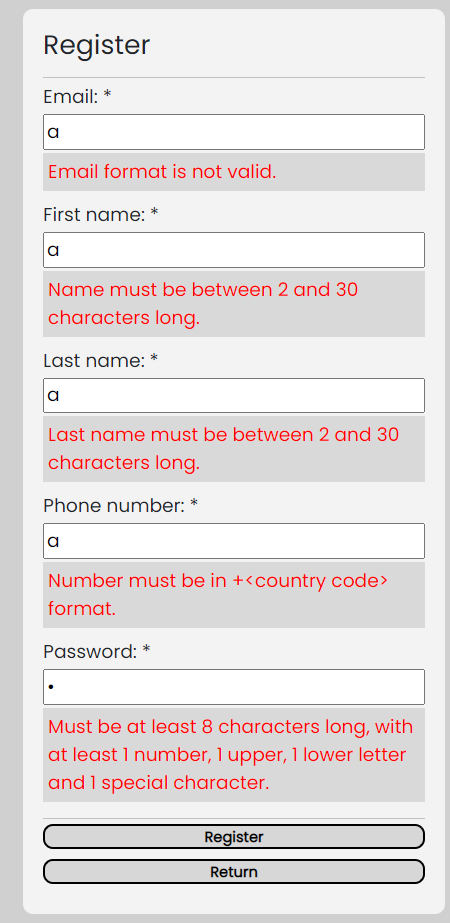
## Login

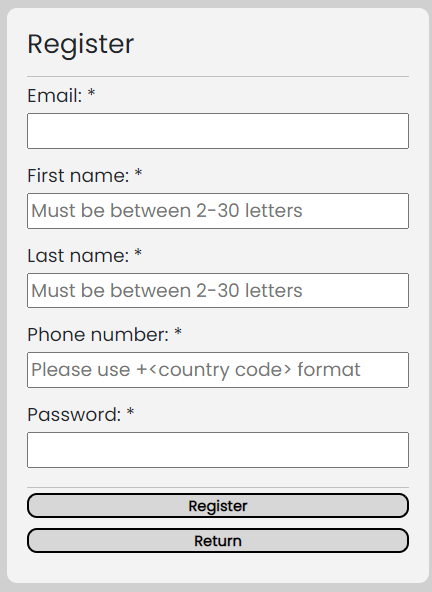




Ovde se moze logovati. Login forma je odradjena kao popUp. Sve greske do kojih moze doci i na serveru bivaju uredno ispisane. Provera se vrsi i na frontu i na backu.

## Register





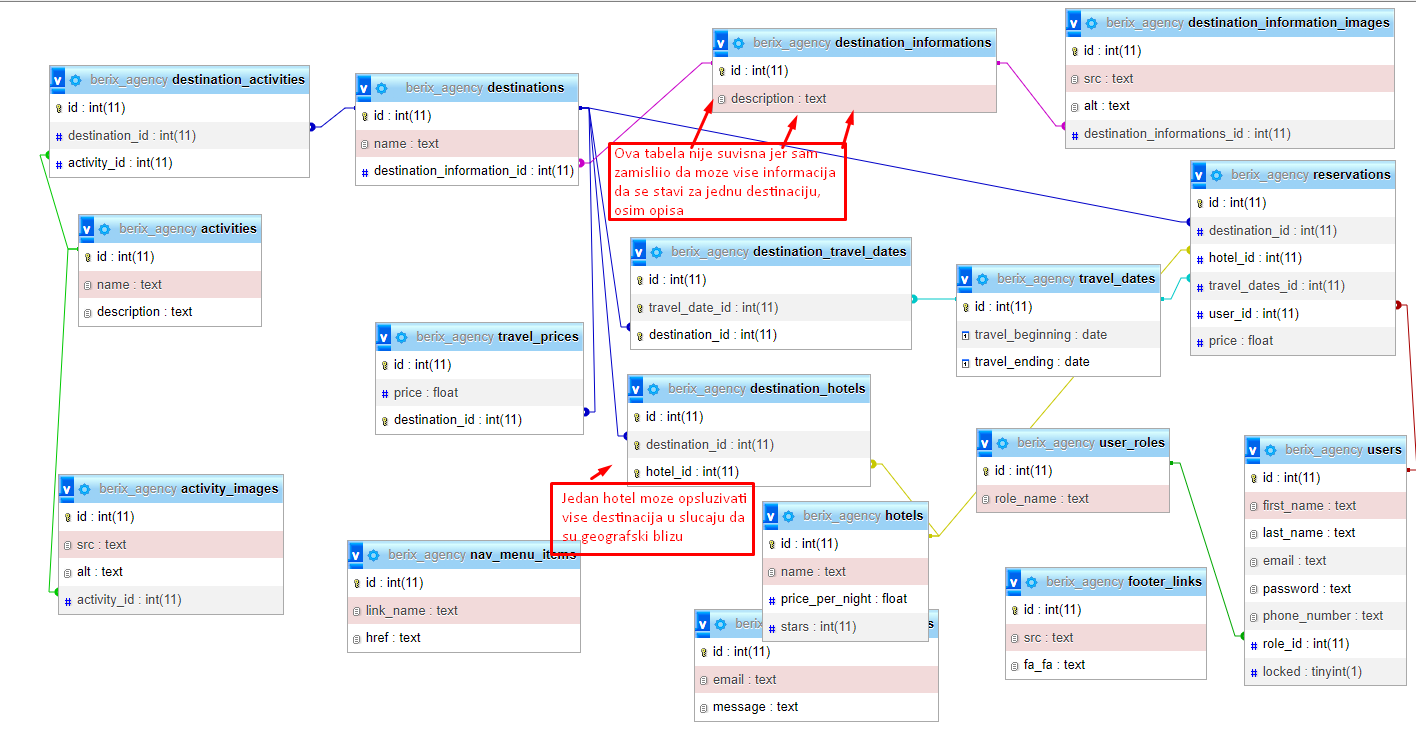
Slicno kao sto je bila situacija sa loginom, ovde se korisnik moze registrovati. Provere se vrse i na frontu i na backu. Admin ne moze da se registruje. Admin se moze dodati samo direktno unosom u tabelu posto je to vrlo kriticno.

## Author

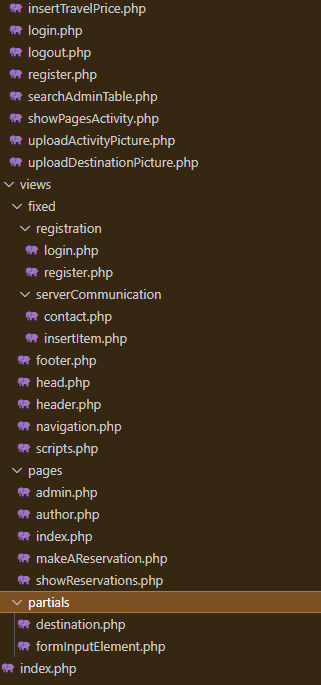


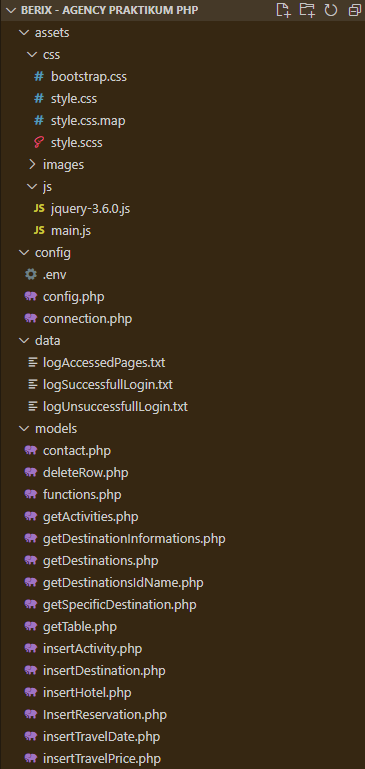
Klasicno o autoru. Nije odradjen export u dokument fajl. Odmah da napisem da nije odradjen ni eksport bilo kojeg dela sajta u excel formatu. Sve ostalo je uradjeno osim update-a neke od tabela.

# Dizajn Baze Podataka



# Struktura Projekta





# Svi Kodovi (moguce da ima razlike u odnosu na hostovanu verziju sajta)

## assets

### sass

@import url('https://fonts.googleapis.com/css2?family=Poppins&family=Racing+Sans+One&display=swap');

\*{

margin:0px;

padding:0px;

font-family: 'Poppins';

}

.holder{

width:100%;

}

header{

background-color: #fff;

display:flex;

flex-direction: row;

justify-content: space-between;

align-items: center;

position:sticky;

top:0px;

z-index: 2;

div, nav{

margin: 5px 20px;

width:15%;

h2 {

text-align: center;

}

}

nav{

width:70%;

background-color: #eaeaea;

}

}

#scroll-to-top{

display: none;

flex-direction: column;

align-items: center;

justify-content: center;

background-color: #b72d00;

color:#fff;

width:50px;

height:50px;

z-index:1;

position:fixed;

bottom:20px;

right:20px;

transition: all ease-in-out 0.3s;

&:hover{

transform: scale(1.1);

background-color: #e73a00;

border-radius: 15px;

}

}

#return-to-index{

background-color: #b72d00;

color:#fff;

padding:10px;

z-index:1;

position:fixed;

top:17%;

left:20px;

transition: all ease-in-out 0.3s;

&:hover{

transform: scale(1.1);

background-color: #e73a00;

border-radius: 15px;

}

}

#go-to-reservations{

background-color: #b72d00;

color:#fff;

padding:10px;

z-index:1;

position:fixed;

top:12%;

left:20px;

transition: all ease-in-out 0.3s;

&:hover{

transform: scale(1.1);

background-color: #e73a00;

border-radius: 15px;

}

}

#error-message{

background-color: #b72d00;

color:#fff;

padding:10px;

z-index:3;

position:fixed;

width:100%;

bottom:0px;

left:0px;

text-align: center;

transition: all ease-in-out 0.3s;

&:hover{

background-color: #e73a00;

}

}

nav{

padding:10px;

#btnMenu{

display:none;

}

ul{

display:flex;

flex-direction: row;

justify-content: space-around;

align-items: center;

list-style: none;

a{

text-decoration: none;

color: #fff;

background-color: #000;

transition: all .3s ease-in-out;

&:hover{

animation: tilt-shaking 0.25s 2;

color:#fff;

}

li{

font-size:14pt;

padding:5px;

}

&.active-nav-item{

color: #000;

background-color: #fff;

animation: none;

&:hover {

transform: scale(1.1);

}

}

}

}

}

.popup-whole-screen{

z-index: 3;

position:fixed;

background-color: #d0d0d0;

display: flex;

flex-direction: column;

align-items: center;

justify-content: center;

display:none;

// .visible{

// }

.invisible{

display: none;

}

.popup-wrapper{

background-color: #f3f3f3;

width:30%;

padding:20px;

border-radius: 10px;

h3, hr{

margin-bottom: 5px;

}

h3{

font-size: 20pt;

}

.form-input-element{

display:flex;

flex-direction: column;

align-items: center;

justify-content: center;

font-size: 14pt;

margin-bottom: 10px;

label{

width:100%;

}

input, select{

width:100%;

margin-top: 3px;

padding: 2px;

}

#image{

color:red;

}

span{

width:100%;

margin-top: 3px;

background-color: #dad9d9;

color:red;

}

}

.button-group{

span{

font-size: 14pt;

width:100%;

margin-top: 3px;

color:red;

}

input{

width: 100%;

border-radius: 10px;

padding: 4px;

height:25px;

font-weight: bold;

font-size:11pt;

transition: all .5s ease-in-out;

background-color: #d7d7d7;

margin-bottom:10px;

color:#000;

display: flex;

flex-direction: column;

align-items: center;

justify-content: center;

&:hover{

background-color: #000;

color:#fff;

}

}

}

}

}

footer{

display:flex;

flex-direction: column;

align-items: center;

justify-content: center;

background-color: #000;

padding:10px;

#footer-wrapper{

width:40%;

background-color: rgba(224, 224, 224, 0.8);

display:flex;

flex-direction: column;

align-items: center;

justify-content: center;

div{

margin: 5px 0px;

&#footer-copyright{

hr{

margin-top:3px;

border-top: 1px solid #000;

}

}

&#footer-links{

font-size: 30pt;

display:flex;

flex-direction: row;

justify-content: space-evenly;

align-items: center;

a{

color: #000;

transition: all .3s ease-in-out;

&:hover{

transform: scale(1.3);

text-shadow:

-1px -1px 0 #000,

1px -1px 0 #000,

-1px 1px 0 #000,

1px 1px 0 #000;

color:rgb(255, 61, 61);

}

}

}

}

}

}

main{

padding:10px;

}

#main-author{

div{

display:flex;

flex-direction: column;

justify-content: center;

align-items: center;

img{

width:500px;

}

p{

width:80%;

margin-top:10px;

font-size: 15pt;

padding:5px;

background-color: #d3d3d3;

a{

color:orangered;

text-decoration: none;

}

}

}

}

#main-index{

.destination-holder{

display:flex;

flex-direction: column;

align-items: center;

justify-content: center;

background-color: #eaeaea;

margin: 10px 0px;

padding:10px;

.destination-information{

margin:10px;

display:flex;

flex-direction: column;

justify-content: center;

align-items: center;

}

.carousel{

width:60%;

transition: all .3s ease-in-out;

&:hover{

transform: scaleX(1.05);

}

}

}

#specific-destination-holder{

display:flex;

flex-direction: column;

align-items: center;

justify-content: center;

background-color: #e7e7e7;

margin: 10px 0px;

padding:10px;

.carousel{

width:60%;

transition: all .3s ease-in-out;

&:hover{

transform: scaleX(1.05);

}

}

#destination-information-holder{

margin:10px;

}

#destination-activities-holder{

h4{

text-align: center;

}

.activity-holder{

display:flex;

flex-direction: column;

align-items: center;

justify-content: center;

background-color: #e7e7e7;

margin: 10px 0px;

padding:10px;

.activity-information{

margin:10px;

}

}

}

}

}

#main-make-a-reservation{

#reservation-holder{

display:flex;

flex-direction: column;

justify-content: center;

align-items: center;

.reservation-region-holder{

width : 100%;

display:flex;

flex-direction: row;

justify-content: center;

align-items: center;

h4{

width:70%;

padding:10px;

background-color:rgb(197, 197, 197);

border-radius: 10px;

text-align: center;

}

#reservation-select-region-holder{

width : 100%;

display:flex;

flex-direction: row;

justify-content: space-around;

align-items: center;

// #reservation-destination-carousel{

// }

.reservation-select-holder{

padding: 10px;

border-radius: 10px;

width:50%;

}

#reservation-destination-carousel{

width:35%;

}

.reservation-select-holder, #reservation-destination-carousel {

display:flex;

flex-direction: column;

align-items: center;

hr{

width:100%;

border-top: 2px solid #000;

}

#reservation-inputs-holder{

display:flex;

flex-direction: column;

align-items: center;

width:100%;

.reservation-input-holder{

width:100%;

padding:10px;

margin:10px 0px;

display:flex;

flex-direction: row;

justify-content:space-between;

align-items: center;

}

}

label{

margin-right:10px;

padding:5px;

width:20%;

word-wrap: break-word;

color: rgb(255, 61, 61);

}

select{

width: 75%;

text-align: center;

padding:5px;

background: #000;

color: #fff;

}

}

}

#submit-reservation-holder{

width:100%;

margin-top:20px;

display:flex;

flex-direction: column;

align-items: center;

label{

padding:10px;

width:40%;

background-color:rgb(197, 197, 197);

text-align: center;

}

span{

color: #b72d00;

font-size: large;

}

#reservation-submit-request-span{

color:#f00;

}

input{

width: 40%;

margin-top:10px;

height:40px;

font-weight: bold;

font-size:11pt;

transition: all .5s ease-in-out;

background-color: #d7d7d7;

margin-bottom:10px;

color:#000;

display: flex;

flex-direction: column;

align-items: center;

justify-content: center;

&:hover{

background-color: #000;

color:#fff;

}

}

}

}

}

}

#main-show-reservations{

display: flex;

flex-direction: column;

align-items: center;

justify-content: center;

padding:10px 30px;

min-width: 400px;

background-color: rgb(143, 139, 139);

border-radius: 10px;

#go-to-reservations{

top:10%;

}

#table-information-wrapper{

display:none;

background-color: rgba(228, 228, 228, 0.9);

padding:5px;

margin-top:50px;

text-align: center;

//#isDeleted{

// color: rgb(197, 42, 42);

// text-shadow: -1px -1px 0 #000, 1px -1px 0 #000,-1px 1px 0 #000,1px 1px 0 #000;

// font-size: 20pt;

// text-align: center;

//}

table{

padding:5px;

border:1px solid;

thead tr{

background-color: rgba(128, 128, 128, 0.9);

}

tr:nth-child(even){

background-color: rgba(185, 184, 184, 0.9);

}

th{

font-size:large;

text-align: center;

}

td, th{

border:1px solid;

padding:10px;

input{

width: 100px;

text-align: center;

&:hover{

transition: all 0.3s ease-in-out;

background-color: #000;

color:#fff;

}

}

}

}

}

}

#main-admin-panel{

#tables-wrapper{

display: flex;

flex-direction: column;

align-items: center;

justify-content: center;

padding:10px 30px;

min-width: 400px;

background-color: rgb(143, 139, 139);

border-radius: 10px;

#statistics{

table{

padding:5px;

border:1px solid;

tr{

text-align: center;

}

thead tr{

background-color: rgba(185, 184, 184, 0.9);

}

tr:nth-child(even){

background-color: rgba(206, 206, 206, 0.9);

}

th{

font-size:large;

text-align: center;

}

td, th{

border:1px solid;

padding:10px;

input{

width: 100px;

text-align: center;

&:hover{

transition: all 0.3s ease-in-out;

background-color: #000;

color:#fff;

}

}

}

}

}

#select-table-to-show{

margin-bottom:10px;

padding:15px;

width:70%;

background-color: #000;

display: flex;

flex-direction: row;

justify-content: space-around;

#admin-search{

display: none;

background-color: rgba(185, 184, 184, 1);

padding:15px;

span{

width:100%;

margin-top: 3px;

background-color: #dad9d9;

color:red;

}

#search-button, #search-reset{

padding:3px;

}

}

select{

text-align: center;

padding:5px;

}

}

#table-information-wrapper{

display:none;

background-color: rgba(228, 228, 228, 0.9);

padding:5px;

//#isDeleted{

// color: rgb(197, 42, 42);

// text-shadow: -1px -1px 0 #000, 1px -1px 0 #000,-1px 1px 0 #000,1px 1px 0 #000;

// font-size: 20pt;

// text-align: center;

//}

table{

padding:5px;

border:1px solid;

thead tr{

background-color: rgba(185, 184, 184, 0.9);

}

tr:nth-child(even){

background-color: rgba(185, 184, 184, 0.9);

}

th{

font-size:large;

text-align: center;

}

td, th{

border:1px solid;

padding:10px;

input{

width: 100px;

text-align: center;

&:hover{

transition: all 0.3s ease-in-out;

background-color: #000;

color:#fff;

}

}

}

}

}

#select-item-to-insert{

margin-bottom:10px;

margin-top:10px;

padding:15px;

width:70%;

background-color: #000;

display: flex;

flex-direction: row;

justify-content: space-around;

select{

text-align: center;

padding:5px;

}

}

}

}

@media screen and (max-width: 1000px) and (min-width:0px){

header{

display:block;

#header-logo{

display:none;

}

#header-text{

display:none;

}

nav {

position:fixed;

top:0px;

padding:0px;

margin:0px;

width:100%;

display:block;

#btnMenu{

display:block;

width:100%;

background-color: #aeaeae;

height:40px;

}

ul{

display:none;

padding:0px;

a{

li{

text-align: center;

text-decoration: none;

color: #fff;

background-color: #000;

transition: all .3s ease-in-out;

border-style: solid;

border-width: 0px 0px 2px 0px;

&:hover{

transform: scale(1.1);

border-width: 2px 0px 2px 0px;

color: #f00;

}

}

&.active-nav-item{

li{

color: #000;

background-color: #fff;

animation: none;

&:hover {

color: #f00;

transform: scale(1.1);

}

}

}

}

&.show{

display:block;

}

}

}

}

.popup-whole-screen{

.popup-wrapper{

width:auto;

}

}

footer{

#footer-wrapper{

width:95%;

}

}

#main-author{

div{

img{

width:200px;

}

}

}

#main-make-a-reservation{

padding-top:50px;

#reservation-holder{

.reservation-region-holder{

h4{

width:100%;

}

.carousel{

width:100%;

}

#reservation-select-region-holder{

flex-direction: column;

.reservation-select-holder, #reservation-destination-carousel{

width:100%;

#reservation-inputs-holder{

.reservation-input-holder{

flex-direction: column;

align-items: center;

text-align: center;

width:100%;

select{

width:100%;

}

label{

width:100%;

}

}

}

}

}

#submit-reservation-holder{

label{

width:100%;

}

input{

width:100%;

}

}

}

}

}

#main-show-reservations{

#table-information-wrapper{

margin-top:80px;

}

#go-to-reservations{

top:42px;

right:auto;

left:2px;

}

}

#main-index{

padding-top:100px;

#return-to-index, #go-to-reservations{

top:50px;

}

#go-to-reservations{

right:0px;

left:auto;

}

#return-to-index{

left:0px;

}

#destinations-holder, #specific-destination-holder{

.carousel{

width:95%;

}

}

}

}

@keyframes tilt-shaking {

0% { transform: rotate(0deg); }

25% { transform: rotate(3deg); }

50% { transform: rotate(0deg); }

75% { transform: rotate(-3deg); }

100% { transform: rotate(0deg); }

}

### main.js

$(document).ready(function(){

$(".carousel").carousel({

'pause': false

});

//Navigation Menu Functionalities

//region

function getUrlPath(){

let urlPath = window.location.pathname.split("/");

urlPath = urlPath[urlPath.length - 1];

return urlPath;

}

$("#scroll-to-top").click(()=>window.scrollTo(0,0));

$(document).scroll(function(){

if($("header")[0].offsetTop>300){

$("#scroll-to-top").css('display', 'flex');

} else{

$("#scroll-to-top").css('display', 'none');

}

});

function setActiveNavMenuItem(currentNavMenuItem){

let arrayOfNavMenuItems = $("nav a");

let i=0;

for (navMenuItem of arrayOfNavMenuItems){

let navMenuItemPath = navMenuItem.href.split("/");

navMenuItemPath = navMenuItemPath[navMenuItemPath.length - 1];

if(currentNavMenuItem==navMenuItemPath){

arrayOfNavMenuItems[i].classList.add('active-nav-item');

break;

}

i++;

}

}

setActiveNavMenuItem(getUrlPath());

//endregion

//End Navigation Menu Functionalities

//Responsive

//region

function showNavMenuItemsViaButtonClick(){

$("nav ul").toggleClass("show");

}

$("#btnMenu").click(()=>showNavMenuItemsViaButtonClick());

//endregion

//End Responsive

//Popup functionalities

//region

function showPopUp(id) {

let screenWidth = $(window).width();

let screenHeight = $(window).height();

$("#" + id).css({

'width': screenWidth + 'px',

'height': screenHeight + 'px',

'display': 'flex'

});

$("#" + id).children().addClass('visible');

$("#" + id).children().removeClass('invisible');

}

function dropPopUp() {

$(".popup-whole-screen").css("display", "none");

$(".popup-whole-screen").children().removeClass('visible');

$(".popup-whole-screen").children().addClass('invisible');

}

$(window).resize(function () {

screenWidth = $(window).width();

screenHeight = $(window).height();

if ($("#login-form").css('display') == "flex") {

dropPopUp();

showPopUp("login-form");

}

if ($("#register-form").css('display') == "flex") {

dropPopUp();

showPopUp("register-form");

}

if ($("#contact-form").css('display') == "flex") {

dropPopUp();

showPopUp("contact-form");

}

if ($("#insert-item-form").css('display') == "flex") {

dropPopUp();

showPopUp("insert-item-form");

}

});

$(document).keyup(function(e){

if(e.keyCode === 27)

dropPopUp();

if(e.keyCode=== 13){

$(".popup-whole-screen .visible input[type='button']")[0].click()

}

});

//Show popups

function addPopupButtonListeners(){

$("nav ul li:contains('Login')").click(function(){

showPopUp('login-form');

})

$("nav ul li:contains('Contact')").click(function(){

showPopUp('contact-form');

})

$("#login-form .register").click(function(){

dropPopUp();

showPopUp('register-form');

})

$("#login-form .return").click(()=>dropPopUp());

$("#insert-form .return").click(()=>dropPopUp());

$("#contact-form .return").click(()=>dropPopUp());

$("#register-form .return").click(function(){

dropPopUp();

showPopUp('login-form');

});

}

addPopupButtonListeners();

//initial presentation

$("#register-password").attr('type', 'password');

$("#login-password").attr('type', 'password');

$("#contact-message").attr('type', 'textarea');

if($("#register-span").text().trim()!="") showPopUp("register-form");

if($("#login-span").text().trim()!="") showPopUp("login-form");

if($("#contact-span").text().trim()!="") showPopUp("contact-form");

//endregion

//End Popup Functionalities

//Register Functionalities

//region

//Checking Fields

//Regular Expressions

let reFirstName = /^[a-z ,.'-]{2,30}$/;

let reLastName = /^[a-z ,.'-]{2,30}$/;

let rePassword = /^(?=.\*[a-z])(?=.\*[A-Z])(?=.\*\d)(?=.\*[@$!%\*?&])[A-Za-z\d@$!%#\*?&]{8,}$/;

let rePhoneNumber = /(^\+[\d]{10,13})|(^\+[\d]{3,5}(\s\d{2,4}){1,4})/;

let reEmail = /(?:[a-z0-9!#$%&'\*+\/=?^\_`{|}~-]+(?:\.[a-z0-9!#$%&'\*+\/=?^\_`{|}~-]+)\*|"(?:[\x01-\x08\x0b\x0c\x0e-\x1f\x21\x23-\x5b\x5d-\x7f]|\\[\x01-\x09\x0b\x0c\x0e-\x7f])\*")@(?:(?:[a-z0-9](?:[a-z0-9-]\*[a-z0-9])?\.)+[a-z0-9](?:[a-z0-9-]\*[a-z0-9])?|\[(?:(?:(2(5[0-5]|[0-4][0-9])|1[0-9][0-9]|[1-9]?[0-9]))\.){3}(?:(2(5[0-5]|[0-4][0-9])|1[0-9][0-9]|[1-9]?[0-9])|[a-z0-9-]\*[a-z0-9]:(?:[\x01-\x08\x0b\x0c\x0e-\x1f\x21-\x5a\x53-\x7f]|\\[\x01-\x09\x0b\x0c\x0e-\x7f])+)\])/;

let reMessage = /^[\w .,!?:;]{2,200}$/

//End Regular Expressions

//If input is correct remove span text.

$("#register-form .form-input-element input").blur(function(){

checkAllRegistrationFields()

});

//Self explanatory

function checkAllRegistrationFields() {

let firstName = $("#register-first-name").val().toLowerCase();

let lastName = $("#register-last-name").val().toLowerCase();

let password = $("#register-password").val();

let phoneNumber = $("#register-phone-number").val();

let email = $("#register-email").val().toLowerCase();

let alltrue=0

alltrue+= checkRegularExpression(reFirstName, firstName, "Name must be between 2 and 30 characters long.", 'register-first-name-span');

alltrue+= checkRegularExpression(reLastName, lastName, "Last name must be between 2 and 30 characters long.", 'register-last-name-span');

alltrue+= checkRegularExpression(rePassword, password, "Must be at least 8 characters long, with at least 1 number, 1 upper, 1 lower letter and 1 special character.", 'register-password-span');

alltrue+= checkRegularExpression(rePhoneNumber, phoneNumber, "Number must be in +<country code> format.", 'register-phone-number-span');

alltrue+= checkRegularExpression(reEmail, email, "Email format is not valid.", 'register-email-span');

return alltrue==5;

}

function checkRegularExpression(regEx, ex, errMessage, spanId){

if(!regEx.test(ex)&&ex!="") {

$("#"+spanId).text(errMessage);

$("#"+spanId).css('padding', '5px');

return 0;

} else if(ex==""){

$("#"+spanId).text("");

$("#"+spanId).css('padding', '0px');

return 0;

}

else {

$("#"+spanId).text("");

$("#"+spanId).css('padding', '0px');

return 1;

};

}

function checkOptionalRegularExpression(regEx, ex, errMessage, spanId){

if(!regEx.test(ex)&&ex!="") {

$("#"+spanId).text(errMessage);

$("#"+spanId).css('padding', '5px');

return 0;

} else if(ex==""){

$("#"+spanId).text("");

$("#"+spanId).css('padding', '0px');

return 1;

}

else {

$("#"+spanId).text("");

$("#"+spanId).css('padding', '0px');

return 1;

};

}

//End Checking Fields

//Submit form

$("#register-form .register").click(function(){

if(checkAllRegistrationFields()){

let email = $("#register-email").val().toLowerCase();

let firstName = $("#register-first-name").val().toLowerCase();

let lastName = $("#register-last-name").val().toLowerCase();

let phoneNumber = $("#register-phone-number").val();

let password = $("#register-password").val();

let data = {

"email": email,

"password": password,

"firstName": firstName,

"lastName": lastName,

"phoneNumber": phoneNumber

};

$.ajax({

"url": "../../models/register.php",

"method": "post",

"data" : data,

success: function(response){

$("#register-span").text("Please wait...");

$("#register-span").css('padding', '5px');

sessionStorage.setItem("registerIsSuccessfull", "true");

setTimeout(()=>{

location.reload();

}, 1000);

},

error: function(xhr){

if(xhr.status==422){

writeErrors(JSON.parse(xhr.responseText).errors, "register-span");

} else {

$("#register-span").text("Unexpected error. Contact support.");

$("#register-span").css('padding', '5px');

}

}

});

} else{

$("#register-span").text("All information must be entered correctly!");

$("#register-span").css('padding', '5px');

}

})

if(sessionStorage.getItem("registerIsSuccessfull")=="true"){

sessionStorage.removeItem("registerIsSuccessfull");

showPopUp("login-form");

$("#login-span").text("You have successfully registered.");

$("#login-span").css('padding', '5px');

}

//endregion

//End Register Functionalities

//Write error messages in span

function writeErrors(errors, spanId){

let text = "";

for (error of errors){

text+=error + "</br>";

};

$(`#${spanId}`).html(text);

$(`#${spanId}`).css('padding', '5px');

}

//Login Functionalities

//region

//If input is correct remove span text.

$("#login-form-field .form-input-element input").blur(function(){

checkAllLoginFields()

});

function checkAllLoginFields(){

let email = $("#login-email").val().toLowerCase();

let password = $("#login-password").val();

let alltrue=0

alltrue+= checkRegularExpression(rePassword, password, "Must be at least 8 characters long, with at least 1 number, 1 upper, 1 lower letter and 1 special character.", 'login-password-span');

alltrue+= checkRegularExpression(reEmail, email, "Email format is not valid.", 'login-email-span');

return alltrue==2;

}

//Submit form

$("#login-form #login").click(function(){

if(checkAllLoginFields()){

let email = $("#login-email").val().toLowerCase();

let password = $("#login-password").val();

let data = {"email": email, "password": password};

$.ajax({

"url": "../../models/login.php",

"method": "post",

"data" : data,

success: function(response){

localStorage.setItem("isLoggedIn", "true");

$("#login-span").text("Please wait...");

$("#login-span").css('padding', '5px');

setTimeout(()=>location.reload(), 1000);

},

error: function(xhr){

if(xhr.status==422){

writeErrors(JSON.parse(xhr.responseText).errors, "login-span");

} else {

$("#login-span").text("Unexpected error. Contact support.");

$("#login-span").css('padding', '5px');

}

}

});

} else{

$("#login-span").text("All information must be entered correctly!");

$("#login-span").css('padding', '5px');

}

})

//endregion

//End Login Functionalities

//Logout Functionalities

//region

$("ul a li:contains('Logout')").click(()=>{

localStorage.setItem("isLoggedIn", "false");

$.ajax({

"url": "../../models/logout.php",

"method": "get",

success: function(response){

location.reload();

},

error: function(xhr){

}

})

})

//endregion

//End Logout Funcionalities

//Contact Popup

//region

//If input is correct remove span text.

$("#contact-form .form-input-element input").blur(function(){

checkAllContactFields();

});

//Self explanatory

function checkAllContactFields() {

let message = $("#contact-message").val();

let email = $("#contact-email").val();

let alltrue=0

alltrue+= checkRegularExpression(reMessage, message, "Message is either too long or you used characters which are not allowed.", 'contact-message-span');

alltrue+= checkOptionalRegularExpression(reEmail, email, "Email format is not valid.", 'contact-email-span');

return alltrue==2;

}

//Submit

$("#contact-form #contact").click(function(){

if(checkAllContactFields()){

let message = $("#contact-message").val().trim();

let email = $("#contact-email").val().trim();

let data = {

message,

email

};

$.ajax({

"url": "../../models/contact.php",

"method": "get",

data,

success: function(response){

$("#contact-message").val("");

$("#contact-email").val("");

$("#contact-span").text("Successfuly sent your message. Send another one?");

$("#contact-span").css('padding', '5px');

},

error: function(xhr){

if(xhr.status==422){

writeErrors(JSON.parse(xhr.responseText).errors, "contact-span");

} else {

$("#contact-span").text("Unexpected error. Contact support.");

$("#contact-span").css('padding', '5px');

}

}

});

} else{

$("#contact-span").text("All information must be entered correctly!");

$("#contact-span").css('padding', '5px');

}

})

//endregion

//End Contact Popup

//Show Specific Destination

//region

function addShowSpecificDestinationListener(){

$("#main-index .carousel").click(function(){

let id = $(this).attr('data-id');

$.ajax({

'url': '../../models/getSpecificDestination.php',

'method': 'get',

"data": {id},

success: function(response){

writeSpecificDestination(response.destination);

$("#go-to-reservations").click(()=>{

if(localStorage.getItem("isLoggedIn")!="true") showPopUp("login-form");

})

},

error: function(xhr){

}

});

});

}

function writeSpecificDestination(destination){

$("#header-text h2").html("Welcome to Berix");

let i=0;

let html;

//Make a reservation button

if(localStorage.getItem("isLoggedIn")=="true"){

html=`<a href="makeAReservation.php"><div id="go-to-reservations">

<i class="fa fa-plane" aria-hidden="true">&nbsp;&nbsp;Make a reservation</i>

</div></a>`;

} else{

html=`<a href="#"><div id="go-to-reservations">

<i class="fa fa-plane" aria-hidden="true">&nbsp;&nbsp;Login to make a reservation</i>

</div></a>`;

}

//Return to index button

html+=`<a href="index.php"><div id="return-to-index">

<i class="fa-solid fa-arrow-left-long">&nbsp;&nbsp;Return</i>

</div></a><div id="specific-destination-holder" >`;

//ispis naslova

html+=`<h3> ${destination.name} </h3>`

//ispis karousela

html+=`

<div class="carousel slide" data-bs-ride="carousel" data-id="${destination.id}">

<div class="carousel-inner">

`;

for(let image of destination.information\_images){

html+=`

<div class="carousel-item ${i==0?'active':''}">

<img src="../../assets/images/Destinations/` + destination.name+`/`+image.src+`.jpg" class="d-block w-100" alt="${image.alt}">

</div>

`;

i++;

}

i=0;

//gotov ispis carousela

//ispis informacija o destinaciji

let allDestinationInformation = destination.information.description;

allDestinationInformation=allDestinationInformation.split('\n');

html+=`</div></div><div id="destination-information-holder"><hr/>`;

for(let destinationInformation of allDestinationInformation){

html+=`<p>${destinationInformation}</p><hr/>`;

}

html+=`</div>`;

//gotov ispis informacija o destinaciji

//ispis aktivnosti

html+=`<div id="destination-activities-holder" class="holder"><h4>Activities:</h4><hr/>`;

for(let activity of destination.activities){

html+=`<div class="activity-holder"><h5>${activity.name}</h5>`;

//ispis karousela

html+=`

<div class="carousel slide" data-bs-ride="carousel" data-id="${destination.id}">

<div class="carousel-inner">

`;

for(let image of activity.images){

html+=`

<div class="carousel-item ${i==0?'active':''}">

<img src="../../assets/images/Activities/` + activity.name+`/`+image.src+`.jpg" class="d-block w-100" alt="${image.alt}">

</div>

`;

i++;

}

i=0;

html+=`</div>`;

html+=`</div>`;

//gotov ispis karousela

//ispis informacija o aktivnosti

html+=`<div class="activity-information"><p>${activity.description}</p>`;

html+=`<hr/></div>`;

//gotov ispis informacija o aktivnosti

html+=`</div>`;

}

//gotov ispis aktivnosti

html+=`</div>`;

$("#main-index").html(html);

$(".carousel").carousel({

'pause': false

});

window.scrollTo(0,0);

}

addShowSpecificDestinationListener();

//endregion

//End Show Specific Destination

//Admin Panel

//region

//SELECT SVIH TABELA

//region

function writeTableData(table){

let text = `<table><thead><tr>`;

if (table != false) {

columns = Object.keys(table[0]);

for ( let column of columns) {

text += `<th>` + column + `</th>`;

}

text += `

<th>EDIT</th>

<th>DELETE</th></tr></thead><tbody>

`;

for (let row of table) {

text += "<tr>"

for (let column of columns) {

text += `<td>` + row[column] + `</td>`;

}

//ako treba da se za odredjenu tabelu zabrani update

//text += localStorage.getItem("tableChosenForManipulating") == "User" ? "<td>FORBIDDEN</td>" : `<td><input type="button" data-action="update" data-id="${row.id}" value="EDIT"/></td>`;

text += localStorage.getItem("tableChosenForManipulating") != "" ? "<td>FORBIDDEN</td>" : `<td><input type="button" data-action="update" data-id="${row.id}" value="EDIT"/></td>`;

text += `

<td><input type="button" data-action="delete" data-id="${row.id}" value="DELETE"/></td>

`;

text += `</tr>`

}

text += `</tbody></table>`;

} else {

text = "There are no items left in this table";

}

$("#table-information-wrapper").html(text);

$("#table-information-wrapper").css("display", "inherit");

addDeleteListener();

}

function writeTable(table){

localStorage.setItem("tableChosenForManipulating", table);

//ispis za pretragu ako tabela postoji

if(table=="default") {

$("#admin-search").css('display', 'none');

$("#table-information-wrapper").html(`Select a table`);

return;

} else{

$("#admin-search").css('display', 'block');

}

$.ajax({

'url': '../../models/getTable.php',

'method' : 'get',

'data':{table},

success: function(response){

writeTableData(response.table);

writeSearchSelect(response.table);

},

error: function(xhr){

}

});

}

$("#select-table-to-show #table-select").change(function(){

writeTable($(this).val());

});

//endregion

//END SELECT SVIH TABELA

//PRETRAGA PO KOLONI TABELE

//region

function writeSearchSelect(table){

let text=`<option value="default">Select column to search:</option>`;

if(localStorage.getItem("tableChosenForManipulating") != "default"){

let columns = Object.keys(table[0])

for ( let column of columns) {

text += `<option value="` + column + `">` + column + `</option>`;

}

}

$("#select-search").html(text);

}

function addSearchTermEvent(){

$("#search-button").click(function(){

searchTerm($("#select-search")[0].value, $("#search-term").val());

});

$("#search-reset").click(function(){

writeTable($("#select-table-to-show #table-select").val())

});

}

function resetSearch(){

$("#select-search")[0].value = "default";

$("#search-term").val("");

}

function searchTerm(column, term){

if (term.trim()==""||column=="default"){

$("#invalid-search-term").html("Invalid search term!");

setTimeout(()=>$("#invalid-search-term").html(""),2000);

return;

}

let table = localStorage.getItem("tableChosenForManipulating");

$.ajax({

'url': "../../models/searchAdminTable.php",

'method': "get",

"data":{

table,

column,

term

},

success: function(response){

writeTableData(response.table);

resetSearch();

},

error: function(xhr){

}

})

}

addSearchTermEvent();

//endregion

//END PRETRAGA PO KOLONI TABELE

//DELETE IZ TABELE

//region

function addDeleteListener(){

$("input[data-action='delete']").click(function(){

deleteItem($(this).attr("data-id"));

})

}

function deleteItem(id){

let table = $("#select-table-to-show #table-select").val();

if(localStorage.getItem("tableChosenForManipulating")=="showReservations"){

table = "reservations";

};

$.ajax({

'url': '../../models/deleteRow.php',

'method' : 'post',

data : {

table,

id

},

success: function(response){

if(localStorage.getItem("tableChosenForManipulating")=="showReservations"){

writeTable("showReservations");

} else{

writeTable($("#select-table-to-show #table-select").val());

}

if(response.isSuccessfull==false){

writeErrorMessage("An error has occured when trying to delete the item. Probably because of referential integrity. Try to delete the item where the selected row primary key is being referenced first, then try again.");;

}

},

error: function(xhr){

writeErrorMessage("An error has occured when trying to delete the item. Probably because of referential integrity. Try to delete the item where the selected row primary key is being referenced first, then try again.");

}

})

}

$("#error-message").css('display', 'none');

$("#error-message").click(()=>$("#error-message").css('display', 'none'));

function writeErrorMessage(message){

$("#error-message").html(message);

$("#error-message").css('display', 'block');

}

addDeleteListener()

//endregion

//END DELETE IZ TABELE

//INSERT PODATAKA

//region Ispis Insert Popup-a

$("#table-select-to-insert").change(()=>{

let tableChosenForInsertion = $("#table-select-to-insert").val();

localStorage.setItem("tableChosenForInsertion", tableChosenForInsertion);

if(tableChosenForInsertion!="default"){

switch(tableChosenForInsertion){

case "activities":

$.ajax({

"url":"../../models/getDestinations.php",

"method":"get",

success: function(response){

if(response.destinations==undefined){

writeErrorMessage("There is not enough data about some destinations. Please fulfill all necessary data before inserting. Especially if you have added new destinations.");

} else{

insertActivity(response.destinations);

showPopUp("insert-item-form");

}

},

error: function(xhr){

}

});

break;

case "travel\_prices":

$.ajax({

"url":"../../models/getDestinationsIdName.php",

"method":"get",

success: function(response){

if(response.destinations==undefined){

writeErrorMessage("There is not enough data about some destinations. Please fulfill all necessary data before inserting. Especially if you have added new destinations.");

} else{

insertTravelPrice(response.destinations);

showPopUp("insert-item-form");

}

},

error: function(xhr){

}

});

break;

case "destinations":

insertDestination();

showPopUp("insert-item-form");

break;

case "destination\_information\_images":

$.ajax({

"url":"../../models/getDestinations.php",

"method":"get",

success: function(response){

if(response.destinations==undefined){

writeErrorMessage("There is not enough data about some destinations. Please fulfill all necessary data before inserting. Especially if you have added new destinations.");

} else{

insertDestinationImages(response.destinations);

showPopUp("insert-item-form");

}

},

error: function(xhr){

}

});

break;

case "hotels":

$.ajax({

"url":"../../models/getDestinations.php",

"method":"get",

success: function(response){

if(response.destinations==undefined){

writeErrorMessage("There is not enough data about some destinations. Please fulfill all necessary data before inserting. Especially if you have added new destinations.");

} else{

insertHotel(response.destinations);

showPopUp("insert-item-form");

}

},

error: function(xhr){

}

});

break;

case "travel\_dates":

$.ajax({

"url":"../../models/getDestinationsIdName.php",

"method":"get",

success: function(response){

if(response.destinations==undefined){

writeErrorMessage("There is not enough data about some destinations. Please fulfill all necessary data before inserting. Especially if you have added new destinations.");

} else{

insertTravelDate(response.destinations);

showPopUp("insert-item-form");

}

},

error: function(xhr){

}

});

break;

case "activity\_images":

$.ajax({

"url":"../../models/getActivities.php",

"method":"get",

success: function(response){

insertActivityImages(response.activities);

showPopUp("insert-item-form");

},

error: function(xhr){

}

});

break;

}

$("#table-select-to-insert").val("default");

}

});

function insertTextInput(inputId, inputLabelText, inputPlaceholderText, inputSpanId, inputSpanText, counter){

let html="";

for(let i=0; i<counter;i++){

html+=`

<div class="form-input-element">

<label for="${inputId[i]}">${inputLabelText[i]}</label>

<input type="text" placeholder="${inputPlaceholderText[i]}" id="${inputId[i]}" name="${inputId[i]}"/>

<span id="${inputSpanId[i]}">${inputSpanText[i]}</span>

</div>

`;

}

return html;

}

function insertSelectinput(selectId, spanId, defaultText, optionValue, optionText, counter){

let html="";

html+=`

<div class="form-input-element">

<select id="${selectId}" name="${selectId}">

<option value="default"> ${defaultText} </option>

`;

for(let i=0; i<counter;i++){

html+=`

<option value="${optionValue[i]}"> ${optionText[i]} </option>

`;

}

html+=`

</select>

<span id="${spanId}"></span>

</div>

`;

return html;

}

function insertPopupButtons(html, id){

html+=`

<hr/>

<div class="button-group">

<input type="button" id="insert-${id}" name="${id}"value="Insert"/>

<input type="button" class="return" value="Return"/>

<span id="insert-span">

</span>

</div>

`;

return html;

}

//endregion Ispis Insert Popup-a

//region Insert Hotela

function checkAllInsertHotelFields(){

let rePricePerNight = /^[1-9][0-9]\*(\.[0-9]([1-9]?))?$/;

let reHotelName = /^([a-zA-Z ]){2,50}$/;

let reStars = /^[1-5]$/;

let hotelName = $("#hotel-name").val();

let pricePerNight = $("#hotel-price-per-night").val();

let stars = $("#hotel-stars").val();

let alltrue=0

alltrue+= checkRegularExpression(reHotelName, hotelName, "Hotel name must be between 2 and 50 characters long with only letters.", 'hotel-name-span');

alltrue+= checkRegularExpression(rePricePerNight, pricePerNight, "You can only use formats: XY[.Z[V]]", 'hotel-price-per-night-span');

alltrue+= checkRegularExpression(reStars, stars, "Number of stars must be between 1 and 5.", 'hotel-stars-span');

return alltrue==3;

}

function sendInsertHotelRequest(){

if(checkAllInsertHotelFields()){

let hotelName = $("#hotel-name").val();

let pricePerNight = $("#hotel-price-per-night").val();

let stars = $("#hotel-stars").val();

let destinationId = $("#destination-select").val();

let data = {

"hotelName": hotelName,

"pricePerNight": pricePerNight,

"destinationId": destinationId,

"stars": stars

};

$.ajax({

"url": "../../models/insertHotel.php",

"method": "post",

"data" : data,

success: function(response){

$("#insert-span").text("Successfully inserted hotel. Insert another one?");

$("#insert-span").css('padding', '5px');

if($("#select-table-to-show #table-select").val()=="hotels" || $("#select-table-to-show #table-select").val()=="destination\_hotels"){

writeTable($("#select-table-to-show #table-select").val())

}

},

error: function(xhr){

if(xhr.status==422){

writeErrors(JSON.parse(xhr.responseText).errors, "insert-span");

} else {

$("#insert-span").text("Unexpected error. Contact support.");

$("#insert-span").css('padding', '5px');

}

}

});

} else{

$("#insert-span").text("All information must be entered correctly!");

$("#insert-span").css('padding', '5px');

}

}

function insertHotel(destinations){

let inputId = ['hotel-name', 'hotel-price-per-night', "hotel-stars"];

let inputLabelText = ['Enter hotel name: \*', 'Enter price per night (€) : \*', 'Enter the number of stars: \*'];

let inputPlaceholderText = ['', 'Example: 19.39', ''];

let inputSpanId = ['hotel-name-span', 'hotel-price-per-night-span', 'hotel-stars-span'];

let inputSpanText = ['', '', ''];

let html = insertTextInput(inputId, inputLabelText, inputPlaceholderText, inputSpanId, inputSpanText, inputId.length);

let optionValue = [];

let optionText = [];

for(let destination of destinations){

optionValue.push(destination.id);

optionText.push(destination.name);

}

html+= insertSelectinput("destination-select", "destination-select-span", "Select location (optional): ",optionValue, optionText, optionText.length);

html = insertPopupButtons(html, "hotel");

$("#insert-form").html(html);

addPopupButtonListeners();

$("#insert-form .form-input-element input").blur(function(){

checkAllInsertHotelFields();

});

$("#insert-hotel").click(()=>{

sendInsertHotelRequest();

})

}

//endregion Insert Hotela

//region Insert Aktivnosti

function checkAllInsertActivityFields(){

let reActivityName = /^([a-zA-Z ]){2,50}$/;

let reActivityDescription = /^(.){10,}$/;

let activityName = $("#activity-name").val();

let activityDescription = $("#activity-description").val();

let alltrue=0

alltrue+= checkRegularExpression(reActivityName, activityName, "Activity name must be between 2 and 50 characters long with only letters.", 'activity-name-span');

alltrue+= checkRegularExpression(reActivityDescription, activityDescription, "Description can't be empty, or shorter then 10 characters.", 'activity-description-span');

return alltrue==2;

}

function sendInsertActivityRequest(){

if(checkAllInsertActivityFields()){

let activityName = $("#activity-name").val();

let activityDescription = $("#activity-description").val();

let destinationId = $("#destination-select").val();

let data = {

"activityName": activityName,

"activityDescription": activityDescription,

"destinationId": destinationId

};

$.ajax({

"url": "../../models/insertActivity.php",

"method": "post",

"data" : data,

success: function(response){

$("#insert-span").text("Successfully inserted activity. Insert another one?");

$("#insert-span").css('padding', '5px');

if($("#select-table-to-show #table-select").val()=="activities" || $("#select-table-to-show #table-select").val()=="destination\_activities"){

writeTable($("#select-table-to-show #table-select").val())

}

},

error: function(xhr){

if(xhr.status==422){

writeErrors(JSON.parse(xhr.responseText).errors, "insert-span");

} else {

$("#insert-span").text("Unexpected error. Contact support.");

$("#insert-span").css('padding', '5px');

}

}

});

} else{

$("#insert-span").text("All information must be entered correctly!");

$("#insert-span").css('padding', '5px');

}

}

function insertActivity(destinations){

let inputId = ['activity-name', 'activity-description'];

let inputLabelText = ['Enter activity name: \*', 'Enter activity description: \*'];

let inputPlaceholderText = ['', ''];

let inputSpanId = ['activity-name-span', 'activity-description-span'];

let inputSpanText = ['', ''];

let html = insertTextInput(inputId, inputLabelText, inputPlaceholderText, inputSpanId, inputSpanText, inputId.length);

let optionValue = [];

let optionText = [];

for(let destination of destinations){

optionValue.push(destination.id);

optionText.push(destination.name);

}

html+= insertSelectinput("destination-select", "destination-select-span", "Select location (optional): ",optionValue, optionText, optionText.length);

html = insertPopupButtons(html, "activity");

$("#insert-form").html(html);

addPopupButtonListeners();

$("#insert-form .form-input-element input").blur(function(){

checkAllInsertActivityFields();

});

$("#insert-activity").click(()=>{

sendInsertActivityRequest();

});

}

//endregion Insert Aktivnosti

//region Insert Slika Za Aktivnosti

function checkAllInsertActivityImageFields(){

let reActivityImageSrc = /^([a-zA-Z1-9\_.]){3,50}$/;

let reActivityImageAlt = /^([a-zA-Z1-9 ]){2,50}$/;

let activityImageSrc = $("#activity-image-src").val().toLowerCase();

let activityImageAlt = $("#activity-image-alt").val();

let activityId = $("#activity-select").val();

let alltrue=0

if($("#activity-image")[0].files.length==1){

alltrue+= 1;

$("#activity-image-span").html(" ");

$("#activity-image-span").css("padding", "0px");

} else{

writeErrors(["You must upload a picture."], "activity-image-span");

}

if(activityId!="default") {

alltrue+= 1;

$("#activity-select-span").html(" ");

$("#activity-select-span").css("padding", "0px");

} else{

writeErrors(["You must chose an activity!"], "activity-select-span");

}

alltrue+= checkRegularExpression(reActivityImageSrc, activityImageSrc, "Only letters, underscores, dots and numbers allowed. Length is 3-50.", 'activity-image-src-span');

alltrue+= checkRegularExpression(reActivityImageAlt, activityImageAlt, "Only letters and numbers allowed, with spaces. Length is 2-50.", 'activity-image-alt-span');

return alltrue==4;

}

function sendInsertActivityimageRequest(){

if(checkAllInsertActivityImageFields()){

$("#insert-form").attr("action", "../../models/uploadActivityPicture.php");

$("#insert-form").submit();

} else{

$("#insert-span").text("All information must be entered correctly!");

$("#insert-span").css('padding', '5px');

}

}

function insertActivityImages(activities){

let inputId = ['activity-image-src', 'activity-image-alt'];

let inputLabelText = ['Enter image src: \*', 'Enter image alt: \*'];

let inputPlaceholderText = ['Just the name of picture, no .jpg .', 'Name of the activity, capital letter.'];

let inputSpanId = ['activity-image-src-span', 'activity-image-alt-span'];

let inputSpanText = ['', ''];

let html = insertTextInput(inputId, inputLabelText, inputPlaceholderText, inputSpanId, inputSpanText, inputId.length);

html+=`

<div class="form-input-element">

<label for="activity-image">Upload your image here: \*</label>

<input type="file" id="activity-image" name="activity-image"/>

<span id="activity-image-span"></span>

</div>

`;

let optionValue = [];

let optionText = [];

for(let activity of activities){

optionValue.push(activity.id);

optionText.push(activity.name);

}

html+= insertSelectinput("activity-select", "activity-select-span", "Select the activity: \*", optionValue, optionText, optionText.length);

html=insertPopupButtons(html, "activity-image");

$("#insert-form").html(html);

addPopupButtonListeners();

$("#insert-form .form-input-element input").blur(function(){

checkAllInsertActivityImageFields();

});

$("#insert-form .form-input-element #activity-select").change(function(){

checkAllInsertActivityImageFields();

});

$("#insert-activity-image").click(()=>{

sendInsertActivityimageRequest();

});

}

//endregion Insert Slika Za Aktivnosti

//region Insert Datuma Putovanja

function checkAllInsertTravelDateFields(){

let reTravelDate = /^\d{4}\-(0[1-9]|1[012])\-(0[1-9]|[12][0-9]|3[01])$/;

let travelDateBeginning = $("#travel-beginning").val();

let travelDateEnding = $("#travel-ending").val();

let alltrue=0

alltrue+= checkRegularExpression(reTravelDate, travelDateBeginning, "Choose correct date. ", 'travel-beginning-span');

alltrue+= checkRegularExpression(reTravelDate, travelDateEnding, "Choose correct date. ", 'travel-ending-span');

return alltrue==2;

}

function sendInsertTravelDateRequest(){

if(checkAllInsertTravelDateFields()){

// if(true){

let travelDateBeginning = $("#travel-beginning").val();

let travelDateEnding = $("#travel-ending").val();

let destinationId = $("#destination-select").val();

let data = {

"travelDateBeginning": travelDateBeginning,

"travelDateEnding": travelDateEnding,

"destinationId": destinationId

};

$.ajax({

"url": "../../models/insertTravelDate.php",

"method": "post",

"data" : data,

success: function(response){

$("#insert-span").text("Successfully inserted travel date.");

$("#insert-span").css('padding', '5px');

if($("#select-table-to-show #table-select").val()=="travel\_dates" || $("#select-table-to-show #table-select").val()=="destination\_travel\_dates"){

writeTable($("#select-table-to-show #table-select").val())

}

},

error: function(xhr){

if(xhr.status==422){

writeErrors(JSON.parse(xhr.responseText).errors, "insert-span");

} else {

$("#insert-span").html("Unexpected error. Contact support. </br></br>"+xhr.responseText);

$("#insert-span").css('padding', '5px');

}

}

});

} else{

$("#insert-span").text("All information must be entered correctly!");

$("#insert-span").css('padding', '5px');

}

}

function insertTravelDate(destinations){

let inputId = ['travel-beginning', 'travel-ending'];

let inputLabelText = ['Enter beginning date of travel: \*', 'Enter ending date of travel: \*'];

let inputPlaceholderText = ['Example: 2022-10-13', 'Example: 2022-10-19'];

let inputSpanId = ['travel-beginning-span', 'travel-ending-span'];

let inputSpanText = ['', ''];

let html = insertTextInput(inputId, inputLabelText, inputPlaceholderText, inputSpanId, inputSpanText, inputId.length);

let optionValue = [];

let optionText = [];

for(let destination of destinations){

optionValue.push(destination.id);

optionText.push(destination.name);

}

html+= insertSelectinput("destination-select", "destination-select-span", "Select the destination: ",optionValue, optionText, optionText.length);

html = insertPopupButtons(html, "travel-date");

$("#insert-form").html(html, "travel-date");

addPopupButtonListeners();

$("#insert-form .form-input-element input").blur(function(){

checkAllInsertTravelDateFields();

});

$("#insert-travel-date").click(()=>{

sendInsertTravelDateRequest();

})

}

//endregion Insert Datuma Putovanja

//region Insert Cene Prevoza

function checkAllInsertTravelPriceFields(){

let reTravelPrice = /^[1-9][0-9]\*(\.[0-9]([1-9]?))?$/;

let travelPrice = $("#travel-price").val();

let destinationId = $("#destination-select").val();

let alltrue=0

alltrue+= checkRegularExpression(reTravelPrice, travelPrice, "You can only use formats: XY[.Z[V]]", 'travel-price-span');

if(destinationId!="default") {

alltrue+=1;

$("#destination-select-span").html("");

$("#destination-select-span").css("padding", "0px");

} else{

writeErrors(["Some destination must be selected."], "destination-select-span");

}

return alltrue==2;

}

function sendInsertTravelPriceRequest(){

if(checkAllInsertTravelPriceFields()){

let travelPrice = $("#travel-price").val();

let destinationId = $("#destination-select").val();

let data = {

"travelPrice": travelPrice,

"destinationId": destinationId

};

$.ajax({

"url": "../../models/insertTravelPrice.php",

"method": "post",

"data" : data,

success: function(response){

$("#insert-span").text("Successfully inserted Travel Price. Insert another one?");

$("#insert-span").css('padding', '5px');

if($("#select-table-to-show #table-select").val()=="travel\_prices" || $("#select-table-to-show #table-select").val()=="travel\_prices"){

writeTable($("#select-table-to-show #table-select").val())

}

},

error: function(xhr){

if(xhr.status==422){

writeErrors(JSON.parse(xhr.responseText).errors, "insert-span");

} else {

$("#insert-span").html("Unexpected error. Contact support. </br></br>"+xhr.responseText);

$("#insert-span").css('padding', '5px');

}

}

});

} else{

$("#insert-span").text("All information must be entered correctly!");

$("#insert-span").css('padding', '5px');

}

}

function insertTravelPrice(destinations){

let inputId = ['travel-price'];

let inputLabelText = ['Enter travel price (€): \*'];

let inputPlaceholderText = [''];

let inputSpanId = ['travel-price-span'];

let inputSpanText = [''];

let html = insertTextInput(inputId, inputLabelText, inputPlaceholderText, inputSpanId, inputSpanText, inputId.length);

let optionValue = [];

let optionText = [];

for(let destination of destinations){

optionValue.push(destination.id);

optionText.push(destination.name);

}

html+= insertSelectinput("destination-select", "destination-select-span", "Select the destination: ",optionValue, optionText, optionText.length);

html= insertPopupButtons(html, "travel-price");

$("#insert-form").html(html);

addPopupButtonListeners();

$("#insert-form .form-input-element input").blur(function(){

checkAllInsertTravelPriceFields();

});

$("#insert-form .form-input-element select").change(function(){

checkAllInsertTravelPriceFields();

});

$("#insert-travel-price").click(()=>{

sendInsertTravelPriceRequest();

})

}

//endregion Insert Cene Prevoza

//region Insert Destinacije

function checkAllInsertDestinationFields(){

let reDestinationName = /^([a-zA-Z ]){2,50}$/;

let reDestinationDescription = /^(.){10,}$/;

let destinationName = $("#destination-name").val().toLowerCase();

let destinationDescription = $("#destination-description").val();

let alltrue=0

alltrue+= checkRegularExpression(reDestinationName, destinationName, "Destination name must be between 2 and 50 characters long with only letters.", 'destination-name-span');

alltrue+= checkRegularExpression(reDestinationDescription, destinationDescription, "Destination description must be at least 10 characters long.", 'destination-description-span');

return alltrue==2;

}

function sendInsertDestinationRequest(){

if(checkAllInsertDestinationFields()){

let destinationName = $("#destination-name").val().toLowerCase();

let destinationDescription = $("#destination-description").val();

let data = {

"destinationName": destinationName,

"destinationDescription": destinationDescription

};

$.ajax({

"url": "../../models/insertDestination.php",

"method": "post",

"data" : data,

success: function(response){

$("#insert-span").text("Successfully inserted Destination. Insert another one?");

$("#insert-span").css('padding', '5px');

if($("#select-table-to-show #table-select").val()=="destinations" || $("#select-table-to-show #table-select").val()=="destination\_informations"){

writeTable($("#select-table-to-show #table-select").val())

}

},

error: function(xhr){

if(xhr.status==422){

writeErrors(JSON.parse(xhr.responseText).errors, "insert-span");

} else {

$("#insert-span").text("Unexpected error. Contact support.");

$("#insert-span").css('padding', '5px');

}

}

});

} else{

$("#insert-span").text("All information must be entered correctly!");

$("#insert-span").css('padding', '5px');

}

}

function insertDestination(){

let inputId = ['destination-name'];

let inputLabelText = ['Enter destination name: \*'];

let inputPlaceholderText = [''];

let inputSpanId = ['destination-name-span'];

let inputSpanText = [''];

let html = insertTextInput(inputId, inputLabelText, inputPlaceholderText, inputSpanId, inputSpanText, inputId.length);

html += `

<div class="form-input-element">

<label for="destination-description">Enter the destination description: \*</label>

<input type="textarea" id="destination-description" name="destination-description"/>

<span id="destination-description-span"></span>

</div>

`;

html= insertPopupButtons(html, "destination");

$("#insert-form").html(html);

addPopupButtonListeners();

$("#insert-form .form-input-element input").blur(function(){

checkAllInsertDestinationFields();

});

$("#insert-destination").click(()=>{

sendInsertDestinationRequest();

})

}

//endregion Insert Destinacije

//region Insert Slika Destinacije

function checkAllInsertDestinationImageFields(){

let reDestinationImageSrc = /^([a-zA-Z1-9\_.]){3,50}$/;

let reDestinationImageAlt = /^([a-zA-Z1-9 ]){2,50}$/;

let destinationImageSrc = $("#destination-image-src").val().toLowerCase();

let destinationImageAlt = $("#destination-image-alt").val();

let destinationId = $("#destination-select").val();

let alltrue=0

if($("#destination-image")[0].files.length==1){

alltrue+= 1;

$("#destination-image-span").html(" ");

$("#destination-image-span").css("padding", "0px");

} else{

writeErrors(["You must upload a picture."], "destination-image-span");

}

if(destinationId!="default") {

alltrue+= 1;

$("#destination-select-span").html(" ");

$("#destination-select-span").css("padding", "0px");

} else{

writeErrors(["You must chose an activity!"], "activity-select-span");

}

alltrue+= checkRegularExpression(reDestinationImageSrc, destinationImageSrc, "Only letters, underscores, dots and numbers allowed. Length is 3-50.", 'destination-image-src-span');

alltrue+= checkRegularExpression(reDestinationImageAlt, destinationImageAlt, "Only letters and numbers allowed, with spaces. Length is 2-50.", 'destination-image-alt-span');

return alltrue==4;

}

function sendInsertDestinationimageRequest(){

if(checkAllInsertDestinationImageFields()){

$("#insert-form").attr("action", "../../models/uploadDestinationPicture.php");

$("#insert-form").submit();

} else{

$("#insert-span").text("All information must be entered correctly!");

$("#insert-span").css('padding', '5px');

}

}

function insertDestinationImages(infos){

let inputId = ['destination-image-src', 'destination-image-alt'];

let inputLabelText = ['Enter image src: \*', 'Enter image alt: \*'];

let inputPlaceholderText = ['Just the name of picture, no .jpg .', 'Name of the destination, capital letter.'];

let inputSpanId = ['destination-image-src-span', 'destination-image-alt-span'];

let inputSpanText = ['', ''];

let html = insertTextInput(inputId, inputLabelText, inputPlaceholderText, inputSpanId, inputSpanText, inputId.length);

html+=`

<div class="form-input-element">

<label for="image">Upload your image here: \*</label>

<input type="file" id="destination-image" name="destination-image"/>

<span id="destination-image-span"></span>

</div>

`;

let optionValue = [];

let optionText = [];

for(let info of infos){

optionValue.push(info.id);

optionText.push(info.name);

}

html+= insertSelectinput("destination-select", "destination-select-span", "Select the destination information: ", optionValue, optionText, optionText.length);

html=insertPopupButtons(html, "destination-image");

$("#insert-form").html(html);

addPopupButtonListeners();

$("#insert-form .form-input-element input").blur(function(){

checkAllInsertDestinationImageFields();

});

$("#insert-form .form-input-element #destination-select").change(function(){

checkAllInsertDestinationImageFields();

});

$("#insert-destination-image").click(()=>{

sendInsertDestinationimageRequest();

});

};

//endregion Insert Slika Destinacije

//END INSERT PODATAKA

//endregion

//End Admin Panel

//Make A Reservation

//region

function addReservationSelectDestinationListener(){

$("#select-reservation-destination").change(()=>{

let destinationId=$("#select-reservation-destination").val();

if(destinationId!="default"){

$.ajax({

"url":"../../models/getDestinations.php",

"method":"get",

success: function(response){

$("#reservation-destination-carousel").remove();

writeReservationDestinationInput(destinationId, response.destinations);

},

error: function(xhr){

}

});

} else{

$("#reservation-destination-carousel").remove();

$("#reservation-inputs-holder .reservation-input-holder")[1].remove();

$("#reservation-inputs-holder .reservation-input-holder")[1].remove();

$("#reservation-inputs-holder hr")[0].remove();

$("#reservation-inputs-holder hr")[0].remove();

$("#submit-reservation-holder").html("");

}

})

}

function addReservationSelectHotelDateListener(){

$.ajax({

"url":"../../models/getDestinations.php",

"method":"get",

success: function(response){

$("#select-reservation-travel-date").change(()=>{

getSubmitReservationRequestButton(response.destinations);

})

$("#select-reservation-hotel").change(()=>{

getSubmitReservationRequestButton(response.destinations);

})

},

error: function(xhr){

}

});

}

function addReservationSubmitRequestListener(){

$("#reservation-submit-request").click(()=>{

sendReservationRequest();

})

}

addReservationSelectDestinationListener();

function sendReservationRequest(){

let destinationId = $("#select-reservation-destination").val();

let hotelId = $("#select-reservation-hotel").val();

let travelDateId = $("#select-reservation-travel-date").val();

let travelPrice = localStorage.getItem("travelPrice");

let data={

destinationId,

hotelId,

travelPrice,

travelDateId

}

$.ajax({

"url":"../../models/insertReservation.php",

"method":"post",

data,

success: function(response){

$("#reservation-submit-request-span").html("You have successfully made a reservation.")

},

error: function(xhr){

if(xhr.status==403){

writeErrors(JSON.parse(xhr.responseText).errors, "reservation-submit-request-span");

} else if(xhr.status==422){

writeErrors(JSON.parse(xhr.responseText).errors, "reservation-submit-request-span");

} else{

$("#reservation-submit-request-span").html("An unexpected error has occured, please contact support.<br/>"+xhr.responseText);

}

}

});

}

function getReservationSelectDestinationInputHtml(destinations, id){

let html=`

<div class="reservation-input-holder">

<label for="select-reservation-destination">Select your destination: </label>

<select id="select-reservation-destination" name="select-reservation-destination">

<option value="default">Select destination: </option>

`;

for(destination of destinations){

html+=`<option value="${destination.id}" ${destination.id==id?"selected":""}>${destination.name}</option>`;

}

html+=`

</select>

</div>

`;

return html;

}

function getReservationSelectHotel(destinations, destinationId){

let destination;

for (let dest of destinations){

if(dest.id==destinationId) {

destination=dest;

}

}

let html=`

<div class="reservation-input-holder">

<label for="select-reservation-hotel">Select your hotel: </label>

<select id="select-reservation-hotel" name="select-reservation-hotel">

<option value="default">Select hotel: </option>

`;

for(hotel of destination.hotels){

html+=`<option value="${hotel.id}"}>${hotel.name} ${hotel.stars}\* - ${hotel.price\_per\_night}€</option>`;

}

html+=`

</select>

</div>

`;

return html;

}

function getReservationSelectTravelDate(destinations, destinationId){

let destination;

for (let dest of destinations){

if(dest.id==destinationId) {

destination=dest;

}

}

let html=`

<div class="reservation-input-holder">

<label for="select-reservation-travel-date">Select your travel date: </label>

<select id="select-reservation-travel-date" name="select-reservation-travel-date">

<option value="default">Select travel date: </option>

`;

for(date of destination.travel\_dates){

html+=`<option value="${date.id}"}>${date.travel\_beginning} until ${date.travel\_ending}</option>`;

}

html+=`

</select>

</div>

`;

return html;

}

function getSubmitReservationRequestButton(destinations){

let travelDateId =$("#select-reservation-travel-date").val();

let hotelId = $("#select-reservation-hotel").val();

let destinationId = $("#select-reservation-destination").val();

let hotel;

let travelDate;

if(travelDateId!="default" && hotelId!="default"){

let destination;

for (let dest of destinations){

if(dest.id==destinationId) {

destination=dest;

}

}

for (let selectedHotel of destination.hotels){

if(hotelId==selectedHotel.id){

hotel = selectedHotel;

}

}

for (let selectedDate of destination.travel\_dates){

if(travelDateId==selectedDate.id){

travelDate = selectedDate;

}

}

let travelDays = (Date.parse(travelDate.travel\_ending)-Date.parse(travelDate.travel\_beginning))/(1000\*60\*60\*24);

let travelPrice = travelDays\*hotel.price\_per\_night+destination.travel\_price.price;

localStorage.setItem("travelPrice", travelPrice);

let html = `

<label for="reservation-submit-request">Your total price is: <span id="trip-price">${travelPrice}€</span></label>

<input type="button" id="reservation-submit-request" value="Make a reservation now"/>

<span id="reservation-submit-request-span"></span>

`;

$("#submit-reservation-holder").html(html);

addReservationSubmitRequestListener();

} else{

let html=`

<label>You must select both a hotel and a travel date before making a reservation. </label>

`;

$("#submit-reservation-holder").html(html);

}

}

function getReservationSelectDestinationCarousel(destinations, destinationId){

let destination;

let i=0;

for (let dest of destinations){

if(dest.id==destinationId) {

destination=dest;

}

}

let html=`

<div id="reservation-destination-carousel">

<div class="carousel slide" data-bs-ride="carousel">

<div class="carousel-inner">

`;

for(let image of destination.information\_images){

html+=`

<div class="carousel-item ${i++==0?'active':''}">

<img src="../../assets/images/Destinations/${destination.name}/${image.src}.jpg" class="d-block w-100" alt="${image.alt}">

</div>

`;

}

html+=`

</div>

</div>

</div>

`;

return html;

}

function writeReservationDestinationInput(destinationId, destinations){

let html=getReservationSelectDestinationInputHtml(destinations, destinationId);

html+="<hr/>";

html+=getReservationSelectHotel(destinations, destinationId);

html+="<hr/>";

html+=getReservationSelectTravelDate(destinations, destinationId);

$("#reservation-inputs-holder").html(html);

html=getReservationSelectDestinationCarousel(destinations, destinationId);

$("#reservation-select-region-holder").html($("#reservation-select-region-holder").html()+html);

addReservationSelectDestinationListener();

$("#reservation-destination-carousel .carousel").carousel({

interval: 3000

});

addReservationSelectHotelDateListener();

html=`

<label>You must select both a hotel and a travel date before making a reservation. </label>

`;

$("#submit-reservation-holder").html(html);

}

//endregion

//END Make A Reservation

//Show Reservations

if(getUrlPath()=="showReservations.php"){

writeTable("showReservations");

}

//END Show Reservations

//SHOW STATISTICS

if(getUrlPath()=="admin.php"){

$.ajax({

"url":"../../models/showPagesActivity.php",

"method":"get",

success: function(response){

let table=response.table;

//WRITE TABLE

let text = `<table><thead><tr>`;

if (table != false) {

columns = Object.keys(table[0]);

for ( let column of columns) {

text += `<th>` + column + `</th>`;

}

for (let row of table) {

text += "<tr>"

for (let column of columns) {

text += `<td>` + row[column] + `</td>`;

}

text += `</tr>`

}

text += `</tbody></table>`;

$("#statistics").html(text);

} else {

text = "There are no items to show.";

$("#statistics").html(text)

}

},

error: function(xhr){

}

})

}

//END SHOW STATISTICS

});

## config

### .env

DBNAME=berix\_agency

SERVER=localhost

USERNAME=root

PASSWORD=

### config.php

<?php

define("ABSOLUTE\_PATH", $\_SERVER["DOCUMENT\_ROOT"]."/Berix - Agency Praktikum PHP");

define("ENV\_FAJL", ABSOLUTE\_PATH."/config/.env");

define("LOG\_FAJL", ABSOLUTE\_PATH."/data/log.txt");

define("SERVER", env("SERVER"));

define("DATABASE", env("DBNAME"));

define("USERNAME", env("USERNAME"));

define("PASSWORD", env("PASSWORD"));

function env($name){

$data = file(ENV\_FAJL);

foreach($data as $row){

$values = explode("=", $row);

if($name==$values[0]) return (trim($values[1]));

}

}

### connection.php

<?php

require\_once ("config.php");

require\_once (ABSOLUTE\_PATH."/models/functions.php");

try {

$conn = new PDO("mysql:host=".SERVER.";dbname=".DATABASE.";charset=utf8", USERNAME, PASSWORD);

$conn->setAttribute(PDO::ATTR\_DEFAULT\_FETCH\_MODE, PDO::FETCH\_OBJ);

$conn->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

}

catch(PDOException $ex){

echo $ex->getMessage();

}

function executeQuery($query){

global $conn;

return $conn->query($query)->fetchAll();

}

function executeQueryOneRow($query){

global $conn;

return $conn->query($query)->fetch();

}

?>

## data (sadrzi i bazu podataka kao backup, za svaki slucaj)

### logAccessedPages.txt (napisacu samo par linija)

admin.php admin@admin.com 1660658447

index.php admin@admin.com 1660658448

admin.php admin@admin.com 1660658449

author.php admin@admin.com 1660658450

admin.php admin@admin.com 1660658451

showReservations.php admin@admin.com 1660658455

admin.php admin@admin.com 1660658456

admin.php admin@admin.com 1660658517

admin.php anonymous 1660658774

index.php anonymous 1660658774

### logSuccessfullLogin.php (napisacu samo par linija)

14 ::1 1660658809

14 ::1 1660659992

### logUnsuccessfullLogin (napisacu samo par linija)

14 ::1 1660640059

14 ::1 1660640076

14 ::1 1660640077

14 ::1 1660658782

## models

### contact.php

<?php

session\_start();

include("../config/connection.php");

$errors = [];

if(!isset($\_GET["message"])||empty($\_GET["message"])){

array\_push($errors, "Message can't be empty.");

exitWithError(422, $errors);

}

$message= $\_GET["message"];

$isLoggedIn = false;

if(isset(($\_SESSION["user"]))&&!empty($\_SESSION["user"])) $isLoggedIn = true;

if($isLoggedIn && $\_GET["email"]==""){

$email = $\_SESSION["user"]->email;

} elseif($\_GET["email"]==""){

$email = 'anonymous';

}

else{

$email=$\_GET["email"];

$reEmail = "/(?:[a-z0-9!#$%&'\*+\/=?^\_`{|}~-]+(?:\.[a-z0-9!#$%&'\*+\/=?^\_`{|}~-]+)\*|\"(?:[\x01-\x08\x0b\x0c\x0e-\x1f\x21\x23-\x5b\x5d-\x7f]|\\[\x01-\x09\x0b\x0c\x0e-\x7f])\*\")@(?:(?:[a-z0-9](?:[a-z0-9-]\*[a-z0-9])?\.)+[a-z0-9](?:[a-z0-9-]\*[a-z0-9])?|\[(?:(?:(2(5[0-5]|[0-4][0-9])|1[0-9][0-9]|[1-9]?[0-9]))\.){3}(?:(2(5[0-5]|[0-4][0-9])|1[0-9][0-9]|[1-9]?[0-9])|[a-z0-9-]\*[a-z0-9]:(?:[\x01-\x08\x0b\x0c\x0e-\x1f\x21-\x5a\x53-\x7f]|\\[\x01-\x09\x0b\x0c\x0e-\x7f])+)\])/";

if(!preg\_match($reEmail, $email)) array\_push($errors, "Email format is not valid");

}

$reMessage = "/^[\w .,!?:;]{2,200}$/";

if(!preg\_match($reMessage, $message)) array\_push($errors, "Your message is too short or too long, or you have used forbidden characters.");

if(count($errors)==0){

if(!insertContactMessage($email, $message)){

array\_push($errors, "Could not insert into database. Contact support");

exitWithError(422, $errors);

}

}else{

exitWithError(422, $errors);

}

### deleteRow.php

<?php

include("../config/connection.php");

$table = $\_POST["table"];

if($table=="showReservations"){

$table="reservations";

}

$id = $\_POST["id"];

$bool = deleteRow($table, $id);

if($bool){

header("Content-Type: application/json");

http\_response\_code(200);

echo json\_encode(["isSuccessfull"=>true]);

} else {

header("Content-Type: application/json");

http\_response\_code(422);

echo json\_encode(["isSuccessfull"=>false]);

}

### functions.php

<?php

function exitWithError($status, $errors){

http\_response\_code($status);

echo json\_encode(["errors"=>$errors]);

die();

}

function getNavMenuItems(){

$query = "SELECT \* FROM nav\_menu\_items";

$navMenuItems = executeQuery($query);

return $navMenuItems;

}

function getFooterLinks(){

$query = "SELECT \* FROM footer\_links";

$footerLinks = executeQuery($query);

return $footerLinks;

}

function userExists($email){

global $conn;

$query = "SELECT \* FROM users WHERE email = :email";

$prepare = $conn->prepare($query);

$prepare->bindParam(":email", $email);

$prepare->execute();

return $prepare->fetch();

}

function insertUser($email, $password, $firstName, $lastName, $phoneNumber){

global $conn;

$query = "INSERT INTO users(first\_name, last\_name, email, password, phone\_number, role\_id) VALUES (:firstName, :lastName, :email, :password, :phoneNumber, 2);";

$prepare = $conn->prepare($query);

$prepare->bindParam(":email", $email);

$prepare->bindParam(":lastName", $lastName);

$prepare->bindParam(":firstName", $firstName);

$prepare->bindParam(":phoneNumber", $phoneNumber);

$prepare->bindParam(":password", md5($password));

return $prepare->execute();

}

function insertContactMessage($email, $message){

global $conn;

$query = "INSERT INTO contact\_messages(email, message) VALUES (:email, :message);";

$prepare = $conn->prepare($query);

$prepare->bindParam(":email", $email);

$prepare->bindParam(":message", $message);

return $prepare->execute();

}

function getDestinations(){

global $conn;

$query = "SELECT \* FROM destinations";

$destinations = executeQuery($query);

foreach($destinations as $destination){

$destination->hotels=getDestinationHotels($destination->id);

$destination->travel\_dates=getDestinationTravelDates($destination->id);

$destination->activities=getDestinationActivities($destination->id);

foreach($destination->activities as $activity){

$activity->images = getDestinationActivityImages($activity->id);

}

$destination->travel\_price = getDestinationTravelPrice($destination->id);

$destination->information = getDestinationInformation($destination->id);

$destination->information\_images = getDestinationInformationImages($destination->information->id);

$travelDays = [];

$hotelPrices = [];

foreach($destination->travel\_dates as $travelDate){

array\_push($travelDays, (strtotime($travelDate->travel\_ending)-strtotime($travelDate->travel\_beginning))/(60 \* 60 \* 24));

}

$travelDays=min($travelDays);

foreach ($destination->hotels as $hotel){

array\_push($hotelPrices, $hotel->price\_per\_night);

}

$hotelPrices=min($hotelPrices);

$cheapestTripPrice=$hotelPrices\*$travelDays+$destination->travel\_price->price;

$destination->cheapest\_trip\_price=$cheapestTripPrice;

}

return $destinations;

}

function getDestinationHotels($destinationId){

$query = "SELECT h.id, h.name, h.price\_per\_night, h.stars FROM destination\_hotels d INNER JOIN hotels h ON d.hotel\_id = h.id WHERE d.destination\_id = ".$destinationId.";";

$hotels = executeQuery($query);

return $hotels;

}

function getDestinationTravelDates($destinationId){

$query = "SELECT d.id, d.travel\_beginning, d.travel\_ending FROM travel\_dates d INNER JOIN destination\_travel\_dates t ON d.id = t.travel\_date\_id WHERE t.destination\_id = ".$destinationId.";";

$travelDates = executeQuery($query);

return $travelDates;

}

function getDestinationActivities($destinationId){

$query = "SELECT a.id, a.name, a.description FROM activities a INNER JOIN destination\_activities d ON d.activity\_id = a.id WHERE d.destination\_id = ".$destinationId.";";

$activities = executeQuery($query);

return $activities;

}

function getDestinationActivityImages($activityId){

$query = "SELECT i.id, i.src, i.alt FROM activities a INNER JOIN activity\_images i ON i.activity\_id = a.id WHERE a.id = ".$activityId.";";

$images = executeQuery($query);

return $images;

}

function getDestinationTravelPrice($destinationId){

$query = "SELECT t.id, t.price FROM travel\_prices t INNER JOIN destinations d ON t.destination\_id = d.id WHERE d.id = ".$destinationId.";";

$travelPrice = executeQueryOneRow($query);

return $travelPrice;

}

function getDestinationInformation($destinationId){

$query = "SELECT i.id, i.description FROM destination\_informations i INNER JOIN destinations d ON d.destination\_information\_id = i.id WHERE d.id = ".$destinationId.";";

$info = executeQueryOneRow($query);

return $info;

}

function getDestinationInformationAll(){

$query = "SELECT id FROM destination\_informations;";

$info = executeQuery($query);

return $info;

}

function getDestinationInformationImages($infoId){

$query = "SELECT images.id, images.src, images.alt FROM destination\_informations i INNER JOIN destination\_information\_images images ON i.id = images.destination\_informations\_id WHERE i.id = ".$infoId.";";

$images = executeQuery($query);

return $images;

}

function getTable($table){

$query = "SELECT \* FROM ".$table." ORDER BY id";

return executeQuery($query);

}

function getReservations($userId){

$query = "SELECT r.id, d.name AS 'Destination:', h.name AS 'Hotel:', t.travel\_beginning AS 'Travel start:', t.travel\_ending AS 'Travel end:', r.price AS 'Total Price:'

FROM reservations r

INNER JOIN destinations d ON r.destination\_id=d.id

INNER JOIN travel\_dates t ON r.travel\_dates\_id=t.id

INNER JOIN hotels h ON r.hotel\_id=h.id

WHERE r.user\_id=".$userId.";";

return executeQuery($query);

}

function getTableWithSearch($table, $column, $term){

global $conn;

$query = 'SELECT \* FROM '.$table.' WHERE '.$column.' LIKE "%'.$term.'%"';

$table = executeQuery($query);

return $table;

}

function deleteRow($table, $id){

global $conn;

$query = "DELETE FROM ".$table." WHERE id=:id;";

$prepare = $conn->prepare($query);

$prepare->bindParam(":id", $id);

try{

return $prepare->execute();

}

catch(PDOException $ex){

return false;

}

}

function getActivities(){

$query = "SELECT id, name FROM activities";

return executeQuery($query);

}

function insertHotel($hotelName, $pricePerNight, $stars){

global $conn;

$query = "INSERT INTO hotels(name, price\_per\_night, stars) VALUES (:hotelName, :pricePerNight, :stars);";

$prepare = $conn->prepare($query);

$prepare->bindParam(":hotelName", $hotelName);

$prepare->bindParam(":pricePerNight", $pricePerNight);

$prepare->bindParam(":stars", $stars);

return $prepare->execute();

}

function connectHotelDestination($destinationId){

//Select last inserted hotel

$query = "SELECT \* FROM hotels ORDER BY id DESC LIMIT 1";

$hotel= executeQueryOneRow($query);

$hotelId = $hotel->id;

//Connect hotel to destinaton

global $conn;

$query = "INSERT INTO destination\_hotels(destination\_id, hotel\_id) VALUES(:destinationId, :hotelId)";

$prepare = $conn->prepare($query);

$prepare->bindParam(":destinationId", $destinationId);

$prepare->bindParam(":hotelId", $hotelId);

return $prepare->execute();

}

function connectActivityDestination($destinationId){

//Select last inserted activity

$query = "SELECT \* FROM activities ORDER BY id DESC LIMIT 1";

$activity= executeQueryOneRow($query);

$activityId = $activity->id;

//Connect hotel to destinaton

global $conn;

$query = "INSERT INTO destination\_activities(destination\_id, activity\_id) VALUES(:destinationId, :activityId)";

$prepare = $conn->prepare($query);

$prepare->bindParam(":destinationId", $destinationId);

$prepare->bindParam(":activityId", $activityId);

return $prepare->execute();

}

function insertActivity($activityName, $activityDescription){

global $conn;

$query = "INSERT INTO activities(name, description) VALUES (:activityName, :activityDescription);";

$prepare = $conn->prepare($query);

$prepare->bindParam(":activityName", $activityName);

$prepare->bindParam(":activityDescription", $activityDescription);

return $prepare->execute();

}

function uploadActivityImage($src, $alt, $activityId){

global $conn;

$query = "INSERT INTO activity\_images(src, alt, activity\_id) VALUES (:src, :alt, :activity\_id);";

$prepare = $conn->prepare($query);

$prepare->bindParam(":src", $src);

$prepare->bindParam(":alt", $alt);

$prepare->bindParam(":activity\_id", $activityId);

return $prepare->execute();

}

function uploadDestinationImage($src, $alt, $destinationId){

//Get Destination Information

$destinationInformationId = getDestinationInformation($destinationId);

$destinationInformationId = $destinationInformationId->id;

global $conn;

$query = "INSERT INTO destination\_information\_images(src, alt, destination\_informations\_id) VALUES (:src, :alt, :destinationInformationId);";

$prepare = $conn->prepare($query);

$prepare->bindParam(":src", $src);

$prepare->bindParam(":alt", $alt);

$prepare->bindParam(":destinationInformationId", $destinationInformationId);

return $prepare->execute();

}

function insertDestination($destinationName, $destinationDescription){

//Insert destination description

global $conn;

$query="INSERT INTO destination\_informations(description) VALUES (:destinationDescription)";

$prepare = $conn->prepare($query);

$prepare->bindParam(":destinationDescription", $destinationDescription);

$prepare->execute();

//Get the inserted destination id

$query="SELECT \* FROM destination\_informations ORDER BY id DESC LIMIT 1";

$destinationInformation = executeQueryOneRow($query);

$destinationInformationId = $destinationInformation->id;

//Insert destination

$query="INSERT INTO destinations(name, destination\_information\_id) VALUES (:destinationName, :destinationInformationId);";

$prepare = $conn->prepare($query);

$destinationName = ucfirst($destinationName);

$prepare->bindParam(":destinationName", $destinationName);

$prepare->bindParam(":destinationInformationId", $destinationInformationId);

return $prepare->execute();

}

function checkIfDestinationAlreadyHasAssociatedPrice($destinationId){

global $conn;

$query = "SELECT \* FROM travel\_prices WHERE destination\_id = :destinationId";

$prepare = $conn->prepare($query);

$prepare->bindParam(":destinationId", $destinationId);

$prepare->execute();

$travel\_price = $prepare->fetch();

return $travel\_price;

}

function getDestinationsIdName(){

$query="SELECT id, name FROM destinations";

return executeQuery($query);

}

function insertTravelPrice($travelPrice, $destinationId){

global $conn;

$query = "INSERT INTO travel\_prices(price, destination\_id) VALUES (:travelPrice, :destinationId);";

$prepare = $conn->prepare($query);

$prepare->bindParam(":travelPrice", $travelPrice);

$prepare->bindParam(":destinationId", $destinationId);

return $prepare->execute();

}

function insertTravelDate($travelDateBeginning, $travelDateEnding, $destinationId){

//Insert travel date

global $conn;

$query="INSERT INTO travel\_dates(travel\_beginning, travel\_ending) VALUES (:travelDateBeginning, :travelDateEnding);";

$prepare=$conn->prepare($query);

$prepare->bindParam(":travelDateBeginning", $travelDateBeginning);

$prepare->bindParam(":travelDateEnding", $travelDateEnding);

$isSuccessfull = $prepare->execute();

//If destination is sent connect date to destination

if($destinationId!="default" && $isSuccessfull){

//Get the id of previously inserted date

$query = "SELECT \* FROM travel\_dates ORDER BY id DESC LIMIT 1";

$travelDate = executeQueryOneRow($query);

$travelDateId = $travelDate->id;

//Connect date to destination

$query = "INSERT INTO destination\_travel\_dates(travel\_date\_id, destination\_id) VALUES (:travelDateId, :destinationId);";

$prepare=$conn->prepare($query);

$prepare->bindParam(":travelDateId", $travelDateId);

$prepare->bindParam(":destinationId", $destinationId);

return $prepare->execute();

}

return true;

}

function insertReservation($destinationId, $hotelId, $userId, $travelDateId, $travelPrice){

global $conn;

$query = "INSERT INTO reservations (destination\_id, hotel\_id, travel\_dates\_id, user\_id, price) VALUES (:destinationId, :hotelId, :travelDateId, :userId, :travelPrice)";

$prepare = $conn->prepare($query);

$prepare->bindParam(":destinationId", $destinationId);

$prepare->bindParam(":hotelId", $hotelId);

$prepare->bindParam(":travelDateId", $travelDateId);

$prepare->bindParam(":userId", $userId);

$prepare->bindParam(":travelPrice", $travelPrice);

try{

return $prepare->execute();

}

catch(PDOException $ex){

return false;

}

}

function writeUnsuccessfullLoginAttempt($userId, $hostAddress){

$file =fopen("../data/logUnsuccessfullLogin.txt", "a");

$stringToWrite = $userId."\t".$hostAddress."\t".time()."\n";

fwrite($file, $stringToWrite);

fclose($file);

}

function writeSuccessfullLoginAttempt($userId, $hostAddress){

$file =fopen("../data/logSuccessfullLogin.txt", "a");

$stringToWrite = $userId."\t".$hostAddress."\t".time()."\n";

fwrite($file, $stringToWrite);

fclose($file);

}

function getUniqueLoggedInUsers(){

$file =file("../../data/logSuccessfullLogin.txt");

$users = [];

$numberOfUsers = 0;

$currentTime = time();

foreach($file as $row){

$newRow = explode("\t", $row);

if(($currentTime - (int)$newRow[2])<(24\*60\*60)){

if(!in\_array($newRow[0], $users)){

array\_push($users, $newRow[0]);

$numberOfUsers++;

}

}

}

return $numberOfUsers;

}

function checkIfTooManyLoginTries($userId, $timeSpan, $numberOfTimesToLockAccount){

$file =file("../data/logUnsuccessfullLogin.txt");

$encounters = 0;

$currentTime = time();

foreach($file as $row){

$newRow = explode("\t", $row);

if(($currentTime-$newRow[2])<$timeSpan){

if($newRow[0]==$userId){

$encounters++;

}

}

}

if($encounters>($numberOfTimesToLockAccount-1)){

return true;

}

return false;

}

function updateLockedStatus($userId){

$query="UPDATE users SET locked=true WHERE id=".$userId.";";

executeQuery($query);

}

function writeAccessedPage($page, $userEmail){

$file = fopen("../../data/logAccessedPages.txt", "a");

$stringToWrite = $page."\t".$userEmail."\t".time()."\n";

fwrite($file, $stringToWrite);

fclose($file);

}

function getAccessedPages(){

$file = file("../data/logAccessedPages.txt");

$pages = [];

$occurences = [];

$percentages = [];

$data = [];

//get unique page names

foreach($file as $row){

$pageTitle = explode("\t", $row)[0];

if(!in\_array($pageTitle, $pages)) array\_push($pages, $pageTitle);

}

//initialise occurences so they can be incremented

$numberOfPages = count($pages);

for($i=0;$i<$numberOfPages;$i++){

$occurences[$i]=0;

}

//increment occurences

for($i=0;$i<$numberOfPages;$i++){

foreach($file as $row){

$pageTitle = explode("\t", $row)[0];

if($pages[$i]==$pageTitle){

$occurences[$i]++;

}

}

}

//get percentages

$total = 0;

for($i=0;$i<$numberOfPages;$i++){

$total+=$occurences[$i];

}

for($i=0;$i<$numberOfPages;$i++){

$percentages[$i]=round($occurences[$i]/$total\*100, 0);

}

for($i=0;$i<$numberOfPages;$i++){

array\_push($data, [ "Pages"=>$pages[$i], "Times Visited"=>$occurences[$i], "Times Visited In %"=>$percentages[$i] ]);

}

return $data;

}

### getActivities.php

<?php

include("../config/connection.php");

$activities = getActivities();

header("Content-Type: application/json");

echo json\_encode(["activities"=>$activities]);

### getDestinationInformations.php

<?php

include("../config/connection.php");

$info = getDestinationInformationAll();

header("Content-Type: appication/json");

echo json\_encode(["info"=>$info]);

### getDestinations.php

<?php

include("../config/connection.php");

$destinations = getDestinations();

header("Content-Type: appication/json");

echo json\_encode(["destinations"=>$destinations]);

### getDestinationsIdName.php

<?php

include("../config/connection.php");

$destinations = getDestinationsIdName();

header("Content-Type: appication/json");

echo json\_encode(["destinations"=>$destinations]);

### getSpecificDestination.php

<?php

include("../config/connection.php");

$destinations = getDestinations();

$idToSearch = $\_GET["id"];

foreach($destinations as $destination){

if ($destination->id == $idToSearch) {

header("Content-Type: application/json");

echo json\_encode(["destination"=>$destination]);

}

}

### getTable.php

<?php

session\_start();

include("../config/connection.php");

$tableName = $\_GET["table"];

if($tableName=="showReservations"){

$table=getReservations($\_SESSION["user"]->id);

} else{

$table=getTable($tableName);

}

header("Content-Type: application/json");

echo json\_encode(["table"=>$table]);

### insertActivity.php

<?php

include("../config/connection.php");

$errors = [];

if(!isset($\_POST["activityName"])||empty($\_POST["activityName"])){

array\_push($errors, "Activity name is empty.");

}

if(!isset($\_POST["activityDescription"])||empty($\_POST["activityDescription"])){

array\_push($errors, "Activity description is empty.");

}

if(count($errors)>0){

exitWithError(422, $errors);

}

$reActivityName = "/^([a-zA-Z ]){2,50}$/";

$reActivityDescription = "/^(.){10,}$/";

$activityName = $\_POST["activityName"];

$activityDescription = $\_POST["activityDescription"];

$destinationId = $\_POST["destinationId"];

if(!preg\_match($reActivityName, $activityName)){

array\_push($errors, "Activity name must be between 2 and 50 characters long with only letters.");

}

if(!preg\_match($reActivityDescription, $activityDescription)){

array\_push($errors, "Description can't be empty, or shorter then 10 characters.");

}

if(count($errors)>0){

exitWithError(422, $errors);

}

if(insertActivity($activityName, $activityDescription)){

if($destinationId!="default"){

if(connectActivityDestination($destinationId)){

http\_response\_code(201);

echo("Success");

die();

}else{

http\_response\_code(500);

echo("Error");

die();

};

}

http\_response\_code(201);

echo("Success");

}else{

http\_response\_code(500);

echo("Error");

};

### insertDestination.php

<?php

include("../config/connection.php");

$errors = [];

if(!isset($\_POST["destinationName"])||empty($\_POST["destinationName"])){

array\_push($errors, "Destination name is empty.");

}

if(!isset($\_POST["destinationDescription"])||empty($\_POST["destinationDescription"])){

array\_push($errors, "Destination description is empty.");

}

if(count($errors)>0){

exitWithError(422, $errors);

}

$reDestinationName = "/^([a-zA-Z ]){2,50}$/";

$reDestinationDescription = "/^(.){10,}$/";

$destinationName = $\_POST["destinationName"];

$destinationDescription = $\_POST["destinationDescription"];

if(!preg\_match($reDestinationName, $destinationName)){

array\_push($errors, "Destination name must be between 2 and 50 characters long with only letters.");

}

if(!preg\_match($reDestinationDescription, $destinationDescription)){

array\_push($errors, "Description can't be empty, or shorter then 10 characters.");

}

if(count($errors)>0){

exitWithError(422, $errors);

}

if(insertDestination($destinationName, $destinationDescription)){

http\_response\_code(201);

echo("Success");

}else{

http\_response\_code(500);

echo("Error");

};

### insertHotel.php

<?php

include("../config/connection.php");

$errors = [];

if(!isset($\_POST["hotelName"])||empty($\_POST["hotelName"])){

array\_push($errors, "Hotel name is empty.");

}

if(!isset($\_POST["pricePerNight"])||empty($\_POST["pricePerNight"])){

array\_push($errors, "Hotel price is empty.");

}

if(!isset($\_POST["stars"])||empty($\_POST["stars"])){

array\_push($errors, "The number of hotel stars is empty.");

}

if(count($errors)>0){

exitWithError(422, $errors);

}

$rePricePerNight = "/^[1-9][0-9]\*(\.[0-9]([1-9]?))?$/";

$reHotelName = "/^([a-zA-Z ]){2,50}$/";

$reStars = "/^[1-5]$/";

$hotelName = $\_POST["hotelName"];

$pricePerNight = $\_POST["pricePerNight"];

$stars = $\_POST["stars"];

$destinationId = $\_POST["destinationId"];

if(!preg\_match($rePricePerNight, $pricePerNight)){

array\_push($errors, "Hotel price must be in XY[.Z[V]] format.");

}

if(!preg\_match($reHotelName, $hotelName)){

array\_push($errors, "Hotel name must be between 2 and 50 characters long with only letters.");

}

if(!preg\_match($reStars, $stars)){

array\_push($errors, "Number of stars must be between 1 and 5.");

}

if(count($errors)>0){

exitWithError(422, $errors);

}

if(insertHotel($hotelName, $pricePerNight, $stars)){

if($destinationId!="default"){

if(connectHotelDestination($destinationId)){

http\_response\_code(201);

echo("Success");

die();

}else{

http\_response\_code(500);

echo("Error");

die();

};

}

http\_response\_code(201);

echo("Success");

}else{

http\_response\_code(500);

echo("Error");

};

### insertReservation.php

<?php

session\_start();

include("../config/connection.php");

if(!isset($\_SESSION["user"])||empty($\_SESSION["user"])){

exitWithError(403, ["You must be logged in in order to make a reservation."]);

}

$errors = [];

if(!isset($\_POST["destinationId"]) || empty($\_POST["destinationId"])){

array\_push($errors, "Destination must be selected.");

}

if(!isset($\_POST["hotelId"]) || empty($\_POST["hotelId"])){

array\_push($errors, "Hotel must be selected.");

}

if(!isset($\_POST["travelPrice"]) || empty($\_POST["travelPrice"])){

array\_push($errors, "There has been a problem in sending trip price. Contact support.");

}

if(!isset($\_POST["travelDateId"]) || empty($\_POST["travelDateId"])){

array\_push($errors, "Travel date must be selected.");

}

if(count($errors)>0){

exitWithError(422, $errors);

}

$destinationId =$\_POST["destinationId"];

$hotelId =$\_POST["hotelId"];

$travelPrice =$\_POST["travelPrice"];

$travelDateId =$\_POST["travelDateId"];

$userId = $\_SESSION["user"]->id;

if(insertReservation($destinationId, $hotelId, $userId, $travelDateId, $travelPrice )){

exitWithError(200, ["You have successfully made a reservation."]);

} else{

exitWithError(422, ["There has been an internal server error with processing of your request. Please contact support."]);

}

### insertTravelDate.php

<?php

include("../config/connection.php");

$errors = [];

if(!isset($\_POST["travelDateBeginning"])||empty($\_POST["travelDateBeginning"])){

array\_push($errors, "Beginning travel date can't be empty.");

}

if(!isset($\_POST["travelDateEnding"])||empty($\_POST["travelDateEnding"])){

array\_push($errors, "Ending travel date can't be empty.");

}

if(!isset($\_POST["destinationId"])||empty($\_POST["destinationId"])){

array\_push($errors, "Some destination must be selected.");

}

if(count($errors)>0){

exitWithError(422, $errors);

}

$travelDateBeginning = $\_POST["travelDateBeginning"];

$travelDateEnding = $\_POST["travelDateEnding"];

$destinationId = $\_POST["destinationId"];

$reTravelDate = "/^\d{4}\-(0[1-9]|1[012])\-(0[1-9]|[12][0-9]|3[01])$/";

if(!preg\_match($reTravelDate, $travelDateBeginning)){

array\_push($errors, "Choose correct beginning date.");

}

if(!preg\_match($reTravelDate, $travelDateEnding)){

array\_push($errors, "Choose correct ending date.");

}

if(count($errors)>0){

exitWithError(422, $errors);

}

if(insertTravelDate($travelDateBeginning, $travelDateEnding, $destinationId)){

exitWithError(200, ["Successfully inserted travel date."]);

} else{

exitWithError(502, ["There has been an internal server error. Contact administrator."]);

}

### insertTravelPrice.php

<?php

include("../config/connection.php");

$errors = [];

if(!isset($\_POST["travelPrice"])||empty($\_POST["travelPrice"])){

array\_push($errors, "Travel price can't be empty.");

}

if(!isset($\_POST["destinationId"])||empty($\_POST["destinationId"])){

array\_push($errors, "Some destination must be selected.");

}

if(count($errors)>0){

exitWithError(422, $errors);

}

$travelPrice = $\_POST["travelPrice"];

$destinationId = $\_POST["destinationId"];

$reTravelPrice = "/^[1-9][0-9]\*(\.[0-9]([1-9]?))?$/";

if(!preg\_match($reTravelPrice, $travelPrice)){

array\_push($errors, "Price must be in XY[.Z[V]] format");

}

if($destinationId=="default"){

array\_push($errors, "Some destination must be selected.");

}

if(count($errors)>0){

exitWithError(422, $errors);

}

if(checkIfDestinationAlreadyHasAssociatedPrice($destinationId)!=false){

array\_push($errors, "That destination already has an associated travel price. Delete it first then try again.");

}

if(count($errors)>0){

exitWithError(422, $errors);

}

if(insertTravelPrice($travelPrice, $destinationId)){

exitWithError(200, ["Successfully inserted travel price for your destination."]);

} else{

exitWithError(500, ["There has been an internal server error. Contact administrator."]);

}

### login.php

<?php

session\_start();

include("../config/connection.php");

$errors = [];

if(isset($\_SESSION["user"])&&(!empty($\_SESSION["user"]))){

array\_push($errors, "You are already logged in.");

exitWithError(403, $errors);

}

if(!isset($\_POST["email"])||empty($\_POST["email"])){

array\_push($errors, "Please fill in all the fields.");

exitWithError(422, $errors);

}

if(!isset($\_POST["password"])||empty($\_POST["password"])){

array\_push($errors, "Please fill in all the fields.");

exitWithError(422, $errors);

}

$email = $\_POST["email"];

$password = $\_POST["password"];

$user = userExists($email);

if(!$user){

array\_push($errors, "User with that email does not exist.");

exitWithError(422, $errors);

}

if($user->locked==1){

array\_push($errors, "This account is locked. Contact support.");

exitWithError(422, $errors);

}

$reEmail = "/(?:[a-z0-9!#$%&'\*+\/=?^\_`{|}~-]+(?:\.[a-z0-9!#$%&'\*+\/=?^\_`{|}~-]+)\*|\"(?:[\x01-\x08\x0b\x0c\x0e-\x1f\x21\x23-\x5b\x5d-\x7f]|\\[\x01-\x09\x0b\x0c\x0e-\x7f])\*\")@(?:(?:[a-z0-9](?:[a-z0-9-]\*[a-z0-9])?\.)+[a-z0-9](?:[a-z0-9-]\*[a-z0-9])?|\[(?:(?:(2(5[0-5]|[0-4][0-9])|1[0-9][0-9]|[1-9]?[0-9]))\.){3}(?:(2(5[0-5]|[0-4][0-9])|1[0-9][0-9]|[1-9]?[0-9])|[a-z0-9-]\*[a-z0-9]:(?:[\x01-\x08\x0b\x0c\x0e-\x1f\x21-\x5a\x53-\x7f]|\\[\x01-\x09\x0b\x0c\x0e-\x7f])+)\])/";

$rePassword = "/^(?=.\*[a-z])(?=.\*[A-Z])(?=.\*\d)(?=.\*[@$!%\*?&])[A-Za-z\d@$!%\*?&]{8,}$/";

if(!preg\_match($rePassword, $password)) array\_push($errors, "Password must be at least 8 characters long, with at least 1 number, 1 upper, 1 lower letter and 1 special character.");

if(!preg\_match($reEmail, $email)) array\_push($errors, "Email format is not valid");

if(count($errors)>0){

exitWithError(422, $errors);

}

if(md5($password)!=$user->password) {

writeUnsuccessfullLoginAttempt($user->id, $\_SERVER['REMOTE\_ADDR']);

$numberOfTimesToLockAccount = 3;

$timeSpan = 300;

$numberOfTries = checkIfTooManyLoginTries($user->id, $timeSpan, $numberOfTimesToLockAccount);

if($numberOfTries==true){

updateLockedStatus($user->id);

mail($user->email, "Your account has been locked", "Contact support to recover your account.\r\n");

array\_push($errors, "This account is locked. Contact support.");

exitWithError(422, $errors);

}

array\_push($errors, "Password is not correct.");

}

if(count($errors)==0){

writeSuccessfullLoginAttempt($user->id, $\_SERVER['REMOTE\_ADDR']);

$\_SESSION["user"]=$user;

} else{

exitWithError(422, $errors);

}

### logout.php

<?php

session\_start();

include("../config/connection.php");

$errors = [];

if(!isset($\_SESSION["user"])&&(empty($\_SESSION["user"]))){

array\_push($errors, "You are not logged in.");

exitWithError(403, $errors);

}

session\_destroy();

### register.php

<?php

session\_start();

include("../config/connection.php");

$errors = [];

if(isset($\_SESSION["user"])&&(!empty($\_SESSION["user"]))){

array\_push($errors, "You are already logged in.");

exitWithError(403, $errors);

}

if(!isset($\_POST["email"])||empty($\_POST["email"])){

array\_push($errors, "Please fill in all the fields.");

exitWithError(422, $errors);

}

if(!isset($\_POST["password"])||empty($\_POST["password"])){

array\_push($errors, "Please fill in all the fields.");

exitWithError(422, $errors);

}

if(!isset($\_POST["phoneNumber"])||empty($\_POST["phoneNumber"])){

array\_push($errors, "Please fill in all the fields.");

exitWithError(422, $errors);

}

if(!isset($\_POST["firstName"])||empty($\_POST["firstName"])){

array\_push($errors, "Please fill in all the fields.");

exitWithError(422, $errors);

}

if(!isset($\_POST["lastName"])||empty($\_POST["lastName"])){

array\_push($errors, "Please fill in all the fields.");

exitWithError(422, $errors);

}

$email = $\_POST["email"];

$password = $\_POST["password"];

$lastName = $\_POST["lastName"];

$firstName = $\_POST["firstName"];

$phoneNumber = $\_POST["phoneNumber"];

$reFirstName = "/^[a-z ,.'-]{2,30}$/";

$reLastName = "/^[a-z ,.'-]{2,30}$/";

$rePassword = "/^(?=.\*[a-z])(?=.\*[A-Z])(?=.\*\d)(?=.\*[@$!%\*?&])[A-Za-z\d@$!#%\*?&]{8,}$/";

$rePhoneNumber = "/(^\+[\d]{10,13})|(^\+[\d]{3,5}(\s\d{2,4}){1,4})/";

$reEmail = "/(?:[a-z0-9!#$%&'\*+\/=?^\_`{|}~-]+(?:\.[a-z0-9!#$%&'\*+\/=?^\_`{|}~-]+)\*|\"(?:[\x01-\x08\x0b\x0c\x0e-\x1f\x21\x23-\x5b\x5d-\x7f]|\\[\x01-\x09\x0b\x0c\x0e-\x7f])\*\")@(?:(?:[a-z0-9](?:[a-z0-9-]\*[a-z0-9])?\.)+[a-z0-9](?:[a-z0-9-]\*[a-z0-9])?|\[(?:(?:(2(5[0-5]|[0-4][0-9])|1[0-9][0-9]|[1-9]?[0-9]))\.){3}(?:(2(5[0-5]|[0-4][0-9])|1[0-9][0-9]|[1-9]?[0-9])|[a-z0-9-]\*[a-z0-9]:(?:[\x01-\x08\x0b\x0c\x0e-\x1f\x21-\x5a\x53-\x7f]|\\[\x01-\x09\x0b\x0c\x0e-\x7f])+)\])/";

if(!preg\_match($reFirstName, $firstName)) array\_push($errors, "First name must be between 2 and 30 letters long.");

if(!preg\_match($reLastName, $lastName)) array\_push($errors, "Last name must be between 2 and 30 letters long.");

if(!preg\_match($rePassword, $password)) array\_push($errors, "Password must be at least 8 characters long, with at least 1 number, 1 upper, 1 lower letter and 1 special character.");

if(!preg\_match($rePhoneNumber, $phoneNumber)) array\_push($errors, "Number must be in +<country code> format.");

if(!preg\_match($reEmail, $email)) array\_push($errors, "Email format is not valid");

if(count($errors)>0){

exitWithError(422, $errors);

}

if(userExists($email)){

array\_push($errors, "User with that email already exists.");

exitWithError(422, $errors);

}

if(!insertUser($email, $password, $firstName, $lastName, $phoneNumber)){

array\_push($errors, "Could not insert into database. Contact support");

exitWithError(422, $errors);

}

### searchAdminPanel.php

<?php

include("../config/connection.php");

$table = $\_GET["table"];

$column = $\_GET["column"];

$term = $\_GET["term"];

$newTable = getTableWithSearch($table, $column, $term);

header("Content-Type: application/json");

echo json\_encode(["table"=>$newTable]);

### showPagesActivity.php

<?php

include("../config/connection.php");

$data = getAccessedPages();

if(count($data)==0) $data=false;

header("Content-Type: application/json");

http\_response\_code(200);

echo json\_encode(["table"=>$data]);

### uploadActivityPicture.php

<?php

include("../config/connection.php");

$errors=[];

$uploadDir = "assets/images";

$imageRoot = "";

if(isset($\_FILES["activity-image"]["name"])&&!empty($\_FILES["activity-image"]["name"])){

$uploadDir .= "/Activities/";

$tmpName = $\_FILES['activity-image']['tmp\_name'];

}

if($uploadDir == "assets/images"){

header('Location: ' . $\_SERVER['HTTP\_REFERER']);

http\_response\_code(422);

die();

}

if(!isset($\_POST["activity-image-src"])||empty($\_POST["activity-image-src"])){

array\_push($errors, "Src field is not filled.");

}

if(!isset($\_POST["activity-image-alt"])||empty($\_POST["activity-image-alt"])){

array\_push($errors, "Alt field is not filled.");

}

if(!isset($\_POST["activity-select"])||empty($\_POST["activity-select"])){

array\_push($errors, ctype\_upper("activity is not selected."));

}

if(count($errors)>0){

header('Location: ' . $\_SERVER['HTTP\_REFERER']);

http\_response\_code(422);

die();

}

$src=$\_POST["activity-image-src"];

$alt=$\_POST["activity-image-alt"];

$activityId=$\_POST["activity-select"];

$src=$\_POST["activity-image-src"];

$uploadDir.=$\_POST["activity-image-alt"];

$uploadDir.="/".$\_POST["activity-image-src"].".jpg";

move\_uploaded\_file($tmpName, ABSOLUTE\_PATH."/".$uploadDir );

uploadActivityImage($src, $alt, $activityId);

header('Location: ' . $\_SERVER['HTTP\_REFERER']);

### uploadDestinationPicture.php

<?php

include("../config/connection.php");

$errors=[];

$uploadDir = "assets/images";

if(isset($\_FILES["destination-image"]["name"])&&!empty($\_FILES["destination-image"]["name"])){

$uploadDir .= "/Destinations/";

$tmpName = $\_FILES['destination-image']['tmp\_name'];

}

if($uploadDir == "assets/images"){

header('Location: ' . $\_SERVER['HTTP\_REFERER']);

http\_response\_code(422);

die();

}

if(!isset($\_POST["destination-image-src"])||empty($\_POST["destination-image-src"])){

array\_push($errors, "Src field is not filled.");

}

if(!isset($\_POST["destination-image-alt"])||empty($\_POST["destination-image-alt"])){

array\_push($errors, "Alt field is not filled.");

}

if(!isset($\_POST["destination-select"])||empty($\_POST["destination-select"])){

array\_push($errors, ctype\_upper("destination is not selected."));

}

if(count($errors)>0){

header('Location: ' . $\_SERVER['HTTP\_REFERER']);

http\_response\_code(422);

die();

}

$src=$\_POST["destination-image-src"];

$alt=$\_POST["destination-image-alt"];

$destinationId=$\_POST["destination-select"];

$uploadDir.=$\_POST["destination-image-alt"];

$uploadDir.="/".$\_POST["destination-image-src"].".jpg";

move\_uploaded\_file($tmpName, ABSOLUTE\_PATH."/".$uploadDir );

echo($uploadDir);

uploadDestinationImage($src, $alt, $destinationId);

header('Location: ' . $\_SERVER['HTTP\_REFERER']);

## views

### fixed

### registration

#### login.php

<div class="popup-whole-screen" id="login-form">

<div class="popup-wrapper">

<h3>Login</h3>

<hr/>

<?php

$inputId = ['login-email', 'login-password'];

$inputLabelText = ['Enter your email: \*', 'Enter your password: \*'];

$inputPlaceholderText = ['', ''];

$inputSpanId = ['login-email-span', 'login-password-span'];

$inputSpanText = ['', ''];

$i=count($inputId);

?>

<form action="" method="Post" id="login-form-field" name="login-form-field">

<?php include("../partials/formInputElement.php") ?>

<hr/>

<div class="button-group">

<input type="button" id="login" value="Login"/>

<input type="button" class="register" value="Sign Up"/>

<input type="button" class="return" value="Return"/>

<span id="login-span">

</span>

</div>

</form>

</div>

</div>

#### register.php

<div class="popup-whole-screen" id="register-form">

<div class="popup-wrapper">

<h3>Register</h3>

<hr/>

<?php

$inputId = ['register-email', 'register-first-name', 'register-last-name', 'register-phone-number', 'register-password'];

$inputLabelText = ['Email: \*', 'First name: \*', 'Last name: \*', 'Phone number: \*', 'Password: \*'];

$inputPlaceholderText = ['', 'Must be between 2-30 letters', 'Must be between 2-30 letters', 'Please use +<country code> format' , ''];

$inputSpanId = ['register-email-span', 'register-first-name-span', 'register-last-name-span', 'register-phone-number-span', 'register-password-span'];

$inputSpanText = ['', '', '', '', ''];

$i=count($inputId);

?>

<form action="" method="Post" id="registration-form" name="registration-form" >

<?php include("../partials/formInputElement.php") ?>

<hr/>

<div class="button-group">

<input type="button" class="register" id="register" value="Register"/>

<span id="register-span">

<input type="button" class="return" value="Return"/>

</span>

</div>

</form>

</div>

</div>

### serverCommunication

#### contact.php

<div class="popup-whole-screen" id="contact-form">

<div class="popup-wrapper">

<h3>Message</h3>

<hr/>

<?php

$inputId = ['contact-email', 'contact-message'];

$inputLabelText = ['Enter your email: (optional)', 'Enter your message: \*'];

$inputPlaceholderText = ['', 'Maximum 200 letters.'];

$inputSpanId = ['contact-email-span', 'contact-message-span'];

$inputSpanText = ['', ''];

$i=count($inputId);

?>

<form action="" method="get" id="contact-form" name="contact-form">

<?php include("../partials/formInputElement.php") ?>

<hr/>

<div class="button-group">

<input type="button" id="contact" value="Send message"/>

<input type="button" class="return" value="Return"/>

<span id="contact-span">

</span>

</div>

</form>

</div>

</div>

#### insertItem.php

<div class="popup-whole-screen" id="insert-item-form">

<div class="popup-wrapper">

<h3>Insert:</h3>

<hr/>

<form action="" method="Post" enctype="multipart/form-data" id="insert-form" name="insert-form">

</form>

</div>

</div>

### footer.php

<footer>

<div class="holder" id="footer-wrapper">

<div id="footer-copyright">

<p>&copy; 2022 Berix - Agency, Berisavac Aleksa 32/18 IT </p><hr/>

</div>

<div id="footer-links" class="holder">

<?php

$footerLinks = getFooterLinks();

foreach($footerLinks as $footerLink):

?>

<a href="../../assets/<?= $footerLink->src ?>">

<i class="<?= $footerLink->fa\_fa ?>"></i>

</a>

<?php endforeach ?>

</div>

</div>

</footer>

</body>

</html>

### head.php

<?php

session\_start();

include("../../config/connection.php");

?>

<!DOCTYPE html>

<html lang="en">

<head>

<link rel="icon" href="../../assets/images/logoIcon.ico" type="image/gif" sizes="16x16">

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<meta name="description" content="<?php echo $description; ?>">

<title><?php echo $title; ?></title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous">

<link href="../../assets/css/style.css" rel="stylesheet">

<script src="https://kit.fontawesome.com/9d68c2c5c8.js" crossorigin="anonymous"></script>

</head>

<body>

### header.php

<?php

include("../fixed/registration/login.php");

include("../fixed/registration/register.php");

include("../fixed/serverCommunication/contact.php");

include("../fixed/serverCommunication/insertItem.php");

?>

<div id="scroll-to-top">

<i class="fa-solid fa-arrow-up"></i>

</div>

<header class="holder">

<a href="index.php">

<div id="header-logo">

<img src="../../assets/images/navLogo.png" alt="Berix Logo"/>

</div>

</a>

<?php include("../fixed/navigation.php"); ?>

<div id="header-text">

<h2> <?php

echo ("$headerText");

if(isset($\_SESSION["user"]->first\_name)&&!empty($\_SESSION["user"]) && explode("/", $\_SERVER['REQUEST\_URI'])[count(explode("/", $\_SERVER['REQUEST\_URI']))-1]=="index.php" )

{

$userName= ucfirst($\_SESSION["user"]->first\_name);

echo (", $userName");

}

?> </h2>

</div>

</header>

### navigation.php

<nav>

<button id="btnMenu"><i class="fa fa-bars"></i> Menu</button>

<ul>

<?php

$navMenuItems = getNavMenuItems();

foreach($navMenuItems as $navMenuItem):

if(isset($\_SESSION["user"]->first\_name)&&!empty($\_SESSION["user"])&&$navMenuItem->link\_name=='Login') continue;

if(!isset($\_SESSION["user"]->first\_name)&&empty($\_SESSION["user"])&&$navMenuItem->link\_name=='Reservations') continue;

if(!isset($\_SESSION["user"]->first\_name)&&empty($\_SESSION["user"])&&$navMenuItem->link\_name=='Logout') continue;

if($navMenuItem->link\_name=='Admin') continue;

?>

<a href="<?= $navMenuItem->href ?>">

<li>

<?= $navMenuItem->link\_name ?>

</li>

</a>

<?php endforeach ?>

<?php

$navMenuItems = getNavMenuItems();

foreach($navMenuItems as $navMenuItem):

if($navMenuItem->link\_name!='Admin') continue;

if($navMenuItem->link\_name=='Admin'&&!isset($\_SESSION["user"]->first\_name)&&empty($\_SESSION["user"])||($\_SESSION["user"]->role\_id!=1)) continue;

?>

<a href="<?= $navMenuItem->href ?>">

<li>

<?= $navMenuItem->link\_name ?>

</li>

</a>

<?php endforeach ?>

</ul>

</nav>

### scripts.php

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMhuP+IlRH9sENBO0LRn5q+8nbTov4+1p" crossorigin="anonymous"></script>

<script type="text/javascript" src="../../assets/js/jquery-3.6.0.js"></script>

<script type="text/javascript" src="../../assets/js/main.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/crypto-js/3.1.2/rollups/md5.js"></script>

## pages

### admin.php

<?php

$description = "Admin Panel";

$title = "Admin - Panel";

$headerText = "Admin Panel";

include("../fixed/head.php");

if(!isset($\_SESSION["user"])||empty($\_SESSION["user"])||$\_SESSION["user"]->role\_id!=1){

$url = "http://$\_SERVER[HTTP\_HOST]$\_SERVER[REQUEST\_URI]";

$url = explode("/", $url);

$url[count($url)-1]="index.php";

$url = implode("/", $url);

header("Location: ".$url);

}

include("../fixed/header.php");

//content

?>

<main class="holder" id="main-admin-panel">

<div id="error-message"></div>

<div id="tables-wrapper">

<div id="select-table-to-show">

<select id="table-select">

<option value="default">Select a table to show:</option>

<?php

$tables = [

["Contact Messages", "contact\_messages"],

["Footer Links", "footer\_links"],

["Navigation Links", "nav\_menu\_items"],

["Reservations", "reservations"],

["Users", "users"],

["User Roles", "user\_roles"],

["Hotels", "hotels"],

["Destinations", "destinations"],

["Destination - Information", "destination\_informations"],

["Destination - Images", "destination\_information\_images"],

["Destination - Hotels", "destination\_hotels"],

["Destination - Travel Dates", "destination\_travel\_dates"],

["Destination - Activities", "destination\_activities"],

["Activities", "activities"],

["Activity - Images", "activity\_images"],

["Travel Dates", "travel\_dates"],

["Bus Price", "travel\_prices"]

];

foreach($tables as $table):

?>

<option value="<?= $table[1]?>"><?= ucfirst($table[0])?></option>

<?php endforeach ?>

</select>

<div id="admin-search">

<select id="select-search">

</select>

<input id="search-term" name="search-term" type="text" placeholder="Search term"/>

<input id="search-button" name="search-button" type="button" value="Search"/>

<input id="search-reset" name="search-reset" type="button" value="Reset Table"/>

<span id="invalid-search-term">

</span>

</div>

</div>

<div id="table-information-wrapper">

</div>

<div id="select-item-to-insert">

<select id="table-select-to-insert">

<option value="default">Select a table to add an item into:</option>

<?php

$tables = [

["Destinations", "destinations"],

["Destination Images", "destination\_information\_images"],

["Activities", "activities"],

["Activity Images", "activity\_images"],

["Hotels", "hotels"],

["Travel Dates", "travel\_dates"],

["Travel Price", "travel\_prices"]

];

foreach($tables as $table):

?>

<option value="<?= $table[1]?>"><?= ucfirst($table[0])?></option>

<?php endforeach ?>

</select>

</div>

<div id="users-logged-in">

<p> The number of unique users logged in in the last 24h is:

<?php

echo getUniqueLoggedInUsers();

?>

</p>

</div>

<div id="statistics">

</div>

</div>

</main>

<?php

//endContent

include("../fixed/scripts.php");

include("../fixed/footer.php");

if(isset($\_SESSION["user"])&&!empty($\_SESSION["user"])) $userEmail=$\_SESSION["user"]->email; else $userEmail="anonymous";

writeAccessedPage("admin.php", $userEmail);

?>

### author.php

<?php

$description = "Author information";

$title = "Author";

$headerText = "About Author";

include("../fixed/head.php");

include("../fixed/header.php");

//content

?>

<main class="holder" id="main-author">

<div>

<img src="../../assets/images/profile.jpg" alt="author" />

<p>My name is Aleksa Berisavac. I am currently attending the ICT Academy in Belgrade, orientation internet technologies. I am aspiring to be an accomplished web developer, and am working towards that goal. If you are wondering about my works, you can find them in the <a href="http://beri-portfolio-js-jquery.synergize.co/?i=2#about">Projects</a> section of my portfolio. </p>

</div>

</main>

<?php

//endContent

include("../fixed/scripts.php");

include("../fixed/footer.php");

if(isset($\_SESSION["user"])&&!empty($\_SESSION["user"])) $userEmail=$\_SESSION["user"]->email; else $userEmail="anonymous";

writeAccessedPage("author.php", $userEmail);

?>

### index.php

<?php

$requestedPath = explode("/",$\_SERVER['PHP\_SELF'])[count(explode("/",$\_SERVER['PHP\_SELF']))-1];

if(isset($\_GET["page"]) && !empty($\_GET["page"])) $requestedPath=$\_GET["page"];

if($requestedPath=="index.php"||$requestedPath=="index.html"||$requestedPath==""){

} else{

$requestedPath = explode("=", $requestedPath)[count(explode("=", $requestedPath))-1];

echo $requestedPath;

switch($requestedPath){

default:

header("Location: index.php");

die();

case "author.php":

header("Location: author.php");

die();

case "admin.php":

header("Location: admin.php");

die();

case "showReservations.php":

header("Location: showReservations.php");

die();

};

}

$description = "Welcome page for Berix Agency";

$title = "Home - Page";

$headerText = "Welcome To Berix";

include("../fixed/head.php");

include("../fixed/header.php");

//content

?>

<main class="holder" id="main-index">

<div id="destinations-holder" class="holder">

<?php include("../partials/destination.php");

foreach($destinations as $destination):

$i=0;

?>

<div class="destination-holder">

<div class="carousel slide" data-bs-ride="carousel" data-id="<?=$destination->id?>">

<div class="carousel-inner">

<?php foreach($destination->information\_images as $image): ?>

<div class="carousel-item <?= $i++==0?'active':''?>">

<img src="../../assets/images/Destinations/<?=$destination->name?>/<?=$image->src?>.jpg" class="d-block w-100" alt="<?=$image->alt?>">

</div>

<?php endforeach; ?>

</div>

</div>

<div class="destination-information">

<h3><?=$destination->name?></h3>

<p> Starting from <?=$destination->cheapest\_trip\_price?> euros</p>

</div>

</div>

<?php endforeach ?>

</div>

</main>

<?php

//endContent

include("../fixed/scripts.php");

include("../fixed/footer.php");

if(isset($\_SESSION["user"])&&!empty($\_SESSION["user"])) $userEmail=$\_SESSION["user"]->email; else $userEmail="anonymous";

writeAccessedPage("index.php", $userEmail);

?>

### makeAReservation.php

<?php

$description = "Getting a reservation.";

$title = "Reservation";

$headerText = "Make a reservation";

include("../fixed/head.php");

include("../fixed/header.php");

//content

?>

<main class="holder" id="main-make-a-reservation">

<div id="reservation-holder" class="holder">

<?php include("../partials/destination.php"); ?>

<div class="reservation-region-holder">

<h4>Ready to make a trip? Select your destination, hotel, and time when you want to travel below</h4>

</div>

<div class="reservation-region-holder">

<div id="reservation-select-region-holder">

<div class="reservation-select-holder">

<div id="reservation-inputs-holder">

<div class="reservation-input-holder">

<label for="select-reservation-destination">Select your destination: </label>

<select id="select-reservation-destination" name="select-reservation-destination">

<option value="default">Select destination: </option>

<?php foreach($destinations as $destination): ?>

<option value="<?= $destination->id ?>"><?= $destination->name ?></option>

<?php endforeach ?>

</select>

</div>

</div>

</div>

</div>

<div class="reservation-select-holder">

<div class="reservation-destination-holder">

</div>

</div>

</div>

<div class="reservation-region-holder">

<div id="submit-reservation-holder">

</div>

</div>

</div>

</main>

<?php

//endContent

include("../fixed/scripts.php");

include("../fixed/footer.php");

if(isset($\_SESSION["user"])&&!empty($\_SESSION["user"])) $userEmail=$\_SESSION["user"]->email; else $userEmail="anonymous";

writeAccessedPage("makeAReservation.php", $userEmail);

?>

### showReservations.php

<?php

$description = "Show made reservations.";

$title = "Reservations";

$headerText = "See your reservations";

include("../fixed/head.php");

if(!isset($\_SESSION["user"])||empty($\_SESSION["user"])||$\_SESSION["user"]->role\_id!=1){

$url = "http://$\_SERVER[HTTP\_HOST]$\_SERVER[REQUEST\_URI]";

$url = explode("/", $url);

$url[count($url)-1]="index.php";

$url = implode("/", $url);

header("Location: ".$url);

}

include("../fixed/header.php");

//content

?>

<main class="holder" id="main-show-reservations">

<a href="makeAReservation.php"><div id="go-to-reservations">

<i class="fa fa-plane" aria-hidden="true">&nbsp;&nbsp;Make a reservation</i>

</div></a>

<div id="table-information-wrapper"></div>

<div id="error-message"></div>

</main>

<?php

//endContent

include("../fixed/scripts.php");

include("../fixed/footer.php");

if(isset($\_SESSION["user"])&&!empty($\_SESSION["user"])) $userEmail=$\_SESSION["user"]->email; else $userEmail="anonymous";

writeAccessedPage("showReservations.php", $userEmail);

?>

## partials

### destination.php

<?php

$destinations = getDestinations();

### formInputElement.php

<?php

for( $counter=0; $counter<$i ;$counter++):?>

<div class="form-input-element">

<label for="<?= $inputId[$counter] ?>"><?= $inputLabelText[$counter] ?></label>

<input type="text" placeholder="<?= $inputPlaceholderText[$counter] ?>" id="<?= $inputId[$counter] ?>" name="<?= $inputId[$counter] ?>"/>

<span id="<?= $inputSpanId[$counter] ?>"><?= $inputSpanText[$counter] ?></span>

</div>

<?php endfor; ?>

## index.php

<?php

header("Location: views/pages/index.php");

die();

?>