

Technologies Internet :

Application réservation



Tian-Sen Ping Anthony (16333) ; Tas Emine (16362)

Haute Ecole Léonard De Vinci - ECAM

25/12/2016

Table des matières

Introduction.....	2`
Application.....	3
Fonctionnement	3
1. <i>Destination</i>	3
2. <i>Inscription</i>	4
3. <i>Validation</i>	5
4. <i>Confirmation</i>	6
Input test	7
Manager	8`
Annexe.....	11
1. main.php.....	11
2. reservation.php	12
3. detailView.php.....	15
4. detailModel.php	16
5. validationView.php.....	18
6. validationModel.php	19
7. confirmationView.php.....	20
8. manager.php	22
9. controls.php.....	24
10. index.php.....	28
11. exception.php.....	29
12. message.php.....	30
13. people.php.....	31

Introduction

Dans le cadre du cours de technologies Web, nous avons développé une application de réservation de voyage en PHP. Cette application permet aux utilisateurs de réserver leurs voyages, choisir leurs destinations, le nombre de participants, etc.

Nous avons codé en HTML5 et utilisé CSS3 pour la mise en forme.

L'ensemble du code se trouve sur GitHub. Lien :

https://github.com/PingTSen/Tas-Ping_TechnologieWEB



Application

Fonctionnement

L'application de réservation est séparée en plusieurs fenêtres, qui permettent la réservation d'un voyage.

1. Destination

RESERVATION

Le prix de la place est de 10 euros jusqu'à 12 ans et ensuite de 15 euros.

Le prix de l'assurance annulation est de 20 euros quel que soit le nombre de voyageurs.

Destination : Paris

Nombre de places : 1

Assurance annulation ☐

Etape suivante Annulation

Sur cette première page, l'utilisateur peut choisir le nombre de personnes qui souhaitent partir en voyage et leurs destinations, par la même occasion il y a aussi la possibilité de prendre une assurance annulation.

2. Inscription

Entrez vos informations :

Nom :

Age :

[Etape suivante](#)

[Retour](#)

[Annuler](#)

Cette page, permet l'inscription des participants du voyage avec leurs noms et âges. L'utilisateur a aussi l'occasion de retourner en arrière pour changer sa destination ou le nombre de participants, et il peut aussi annuler sa réservation.

3. Validation



Sur cette troisième page, l'organisateur du voyage peut confirmer la destination du voyage et les participants ou modifier (avec un retour en arrière), ainsi qu'annuler la réservation.

4. Confirmation

Confirmations des réservations

Votre demande a bien été enregistrée.

Merci de bien vouloir verser la somme de :

12 €

sur le compte :

000-000000-00

[Retour Accueil](#)

Après la réservation et confirmation, il y a un récapitulatif de la réservation avec le montant à payer sur le compte.

Input test

Les boîtes de dialogue des différentes fenêtres possèdent un test de validation de valeur entré par l'utilisateur.

RESERVATION

Verify that the number of places field is a number and greater than 0 and below 10
The destination field must not be empty.
The specific character is not accepted.

Le prix de la place est de 10 euros jusqu'à 12 ans et ensuite de 15 euros.

Le prix de l'assurance annulation est de 20 euros quel que soit le nombre de voyageurs.

Destination : Paris
Nombre de places : 0
Assurance annulation ☐

Etape suivante

Annulation

Entrez vos informations :

Verify that the age is a number and greater than 0 and below 120
The name must not be empty.
The specific character is not accepted.

Nom :
Age : 1

Etape suivante

Retour

Annuler

L'utilisateur doit entrer des valeurs logiques. Si les valeurs entrées ne sont pas correctes un message d'erreur sera affiché.

Sur la première figure du haut, vous pouvez voir le message d'erreur signalant que le nombre de places ne peut pas être une valeur vide, donc qu'elle doit être supérieure à 0 et inférieure à 10. Le champ de destination ne peut pas rester vide et que les caractères spécifiques ne sont pas acceptés.

Sur la deuxième figure, les erreurs d'entrées sont gérées de la même manière.

Manager

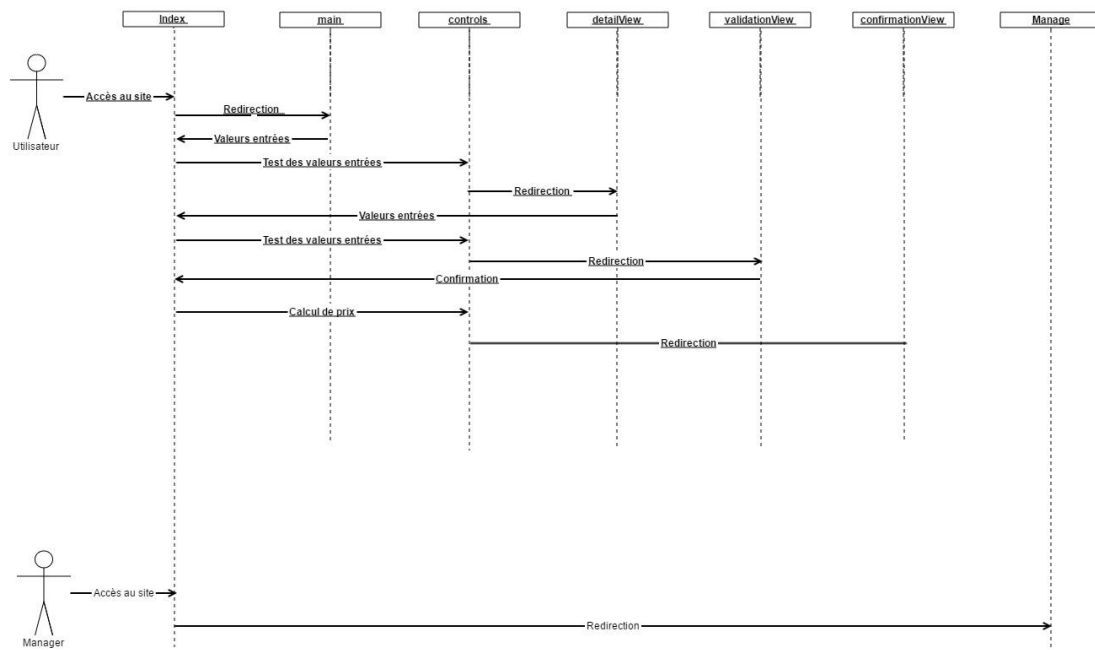
L'application possède aussi une fenêtre réservée spécialement pour le manager. Cette fenêtre reprend tous les réservations qui ont été faite.

Le manager peut faire différentes opérations sur les réservations.

Ces opérations peuvent être des modifications, suppression ou ajout de réservation.



Diagramme de séquence



Voici un diagramme qui reprend le fonctionnement de notre site.

Pour accéder au site, l'utilisateur est redirigé par l'index sur la page principale (page de réservation). Une fois arrivé sur la page principale, l'utilisateur va pouvoir entrer ses données qui vont devoir être confirmées. Si les données entrées sont correctes, la page suivante sera chargée (page de détail).

Sur cette page, l'utilisateur pourra inscrire les différents participants du voyage. Les données concernant les participants du voyage sont aussi vérifiées. Si les données entrées sont correctes alors la page suivante sera chargée (page de validation).

Cette page de validation va reprendre toutes les informations concernant le voyage. L'utilisateur confirmera sa réservation, une fois confirmée la dernière page sera chargée (page de confirmation).

La page de confirmation affiche la somme à payer pour le voyage.

Annexe

1. main.php

```
<!-- Author : Ping Tian-Sen Anthony & Tas Emine -->
```

```
<!--View of the reservation-->
```

```
<!DOCTYPE html>
```

```
<meta charset="UTF-8"/>
<head>
  <title> RESERVATION </title>
  <h1><center> RESERVATION </center></h1>
  <link rel="stylesheet" href='styles.css' />
  <span class="errorMessage">
    <?php if($reservation->getIsError()){
      echo $reservation->getErrorMessage();
    }
  ?>
  </span>
</head>

<body>
  <section>

    <p> Le prix de la place est de 10 euros jusqu'à 12 ans et
    ensuite de 15 euros.</p>
    <p> Le prix de l'assurance annulation est de 20 euros quel
    que soit le nombre de voyageurs.</p><br>
    <form name="rsv" method="post" action="index.php">
      <div id="container">
        <input type="hidden" name="page" value="controls" />
        <input type="hidden" name="check" value="ok" />
        <label for="destination"> Destination:</label>
        <input type="text" name="destination"
        value="<?php echo htmlspecialchars(
          $reservation->getDestination(); ?>" /><br>
        <label for="number_Places"> Nombre de places:</label>
        <input type="text" name="number_of_places"
        value="<?php echo htmlspecialchars(
          $reservation->getNumberOfPlaces(); ?>" /> <br>
        <label for="insurance"> Assurance annulation /label>
        <input type="checkbox" name="insurance" /> <br>
      </div><br>
      <center>
        <a href="#" class="button">
          <label>
            Etape suivante
            <input type="submit" name = 'nextR'
```

```

                                style="display:none" />
        </label>
    </a>
    <a href="#" class="button">
        <label>
            Annulation
            <input type="submit" name = 'cancel'
                                style="display:none" />
        </label>
    </a>
</center><br>
</form>
</section>
</body>
</html>

```

2. reservation.php

```
<!-- Author : Ping Tian-Sen Anthony & Tas Emine -->
```

```
<?php
```

```
include 'message_.php';
```

```

/**
 *Reservation model class
 *This class manage the reservation
 */

```

```
class reservation{
```

```

    private $html;
    private $numberOfPlaces;
    private $destination;
    private $isError;
    private $errorMessage;
    private $isAssured;

```

```

/**
 *Constructor of the class
 */

```

```
public function __construct()
{
```

```

    $this->numberOfPlaces = 0;
    $this->destination="Paris";
    $this->isAssured=false;
    $err = "Verify that the number of places field is a number and greater than 0 and below 10<br>
    The destination field must not be empty. <br>
    The specific character is not accepted.";

```



```
$this->errorMessage=new message_($err);
}

/**
 *Getter of error
 *@return True if a error occur else False
 */

public function getIsError()
{
    return $this->isError;
}

/**
 *Setter of error
 *@param error True if a error occur else False
 */

public function setIsError($error)
{
    $this->isError = $error;
}

/**
 *Getter of the errorMessage
 *@return the message
 */

public function getErrorMessage(){
    return $this->errorMessage->getMessage();
}

/**
 *Getter of the number of participant for a trip
 *@return the number
 */
public function getNumberOfPlaces()
{
    return $this->numberOfPlaces;
}

/**
 *Getter of the destination for a trip
 *@return the destination
 */
public function getDestination()
{
    return $this->destination;
}

/**
 *Setter of the number of participant for a trip
 *@param number the new number of participant
 */
```

```
public function setNumberOfPlaces($number)
{
    $this->numberOfPlaces = $number;
}

/**
 *Setter of destination
 *@dest the new destination
 */
public function setDestination($dest)
{
    $this->destination=$dest;
}

/**
 *Getter of the cancellation insurance
 *@return True if the insurance is eneable else False
 */
public function getIsInsured(){
    return $this->isAssured;
}

/**
 *Setter of the cancellation insurance
 *@pram b the new value, True if the insurance is eneable else False
 */
public function setIsInsured($b){
    $this->isAssured=$b;
}

/**
 *Destructor of the class
 */
public function __destruct ()
{
    $this->numberOfPlaces=1;
    $this->destination="Paris";
    $this->isAssured = false;
}

}

?>
```

3. detailView.php

<!-- Author : Ping Tian-Sen Anthony & Tas Emine -->

<!-- Inscription for each passenger-->

<!DOCTYPE html>

<meta charset="UTF-8"/>

<head>

<title> DETAILS DES RESERVATIONS </title>

<h2><center> Entrez vos informations : </center></h2>

<link rel="stylesheet" href='styles.css' />

<?php if(\$detail->getIsError()){

echo \$detail->getErrorMessage();

}

?>

</head>

<body>

<section>

<form method="post" action="index.php">

<div id="container">

<input type="hidden" name="page" value="controls" />

<input type="hidden" name="check" value="ok" />

<?php

\$listPeople=\$detail->getListPeople();

\$name = " ";

\$age=1;

for(\$i=0; \$i<\$reservation->getNumberOfPlaces(); \$i++)

{

if(\$listPeople != null){

\$name=\$listPeople[\$i]->getName();

\$age=\$listPeople[\$i]->getAge();

}

\$html = '<label for="Name"> Nom :</label>

<input type="text" name="nom[]" value=".\$name."

';

\$html.= '<label for="Age"> Age :</label>

<input type="text" name="age[]" value=".\$age.">

 ';

echo \$html;

}

?>

</div>

<center>

<label>

Etape suivante

<input type="submit" name = 'nextD'

style="display:none" />

```

        /label>
    </a>
    <a href="#" class="button">
        <label >
            Retour
            <input type="submit" name = 'previousD'
                        style="display:none" />
        </label>
    </a>
    <a href="#" class="button">
        <label >
            Annuler
            <input type="submit" name = 'cancel'
                        style="display:none" />
        </label>
    </a>
</center>
</form>
</section>
</body>
</html>

```

4. detailModel.php

```
<!-- Author : Ping Tian-Sen Anthony & Tas Emine -->
```

```
<?php
```

```
include 'people.php';
```

```
/* Information model class.
```

```
   This class get all the data about the people who are part of the reservation. */
```

```
class detail{
```

```
    private $numberPeople;
```

```
    private $name;
```

```
    private $age;
```

```
    private $people;
```

```
    private $isError;
```

```
    private $errorMessage;
```

```
    public function __construct($nb)
```

```
    {
```

```
        $this->numberPeople=$nb;
```

```
        $this->name = null;
```

```
        $this->age = 1;
```

```
        $this->isOk=false;
```

```
        $err = "Verify that the age is a number and greater than 0 and below 120 <br>
```

```
            The name must not be empty. <br>
```

```
            The specific caracer is not accepted.";
```

```

        $this->errorMessage=new message_($err);
    }

    /**
     * This function creat a list of people
     * @param listName, a list of name of each passenger for a reservation
     * @param listAge, a list of age of each passenger for a reservation
     */
    public function createPeople($listName,$listAge){
        for($i=0;$i<$this->numberPeople;$i++){
            $this->people[$i] = new people($listAge[$i],$listName[$i]);
        }
    }

    /**
     *Getter of a list of people
     * @return a list of people
     */
    public function getListPeople()
    {
        return $this->people;
    }

    /**
     *Getter of error
     * @return a boolean True if a error occure else False
     */
    public function getIsError()
    {
        return $this->isError;
    }

    /**
     *Setter of error
     * @param error, a boolean value that indicate if a error occure
     */
    public function setIsError($error)
    {
        $this->isError = $error;
    }

    /**
     *Getter of error Message
     * @return error message
     */
    public function getErrorMessage(){
        return $this->errorMessage->getMessage();
    }

}

?>

```



```

        <input type="submit" name = 'validate' style="display:none" />
    </label>
</a>
<a href="#" class="button">
    <label>
        Retour
        <input type="submit" name = 'previousV' style="display:none" />
    </label>
</a>
<a href="#" class="button">
    <label >
        Annulation
        <input type="submit" name = 'cancel' style="display:none" />
    </label>
</a>
</center>
</form>
</section>
</body>
</html>

```

6. validationModel.php

```
<!-- Author : Ping Tian-Sen Anthony & Tas Emine -->
```

```
<?php
```

```

/**
The scope of this class is to calculate the price for each trip.
*/

```

```
class validate {
```

```
    private $price ;
```

```

    public function __construct (){
        $this->price=0;
    }

```

```

/**
 *This function calculate the price for a trip
 *@param listPeople, participant of a trip
 */

```

```

    public function calculate ($listPeople){
        foreach($listPeople as $people){
            if ($this->isKid($people->getAge())){
                $this->price+=12;
            }
        }
    }

```

```

    }else {
        $this->price+=15;
    }
}
}

/**
 *Function that check if the cancellation insurance is eneable
 */

public function isInsured($insured){
    if($insured){
        $this->price+=20;
    }
}

/**
 *Function that check if a participant is a kid
 *@param age of a participant
 */

private function isKid($age){
    return ($age<=12);
}

/**
 *Getter of the price for a trip
 *@return the price
 */

public function getPrices(){
    return $this->price;
}

}

?>

```

7. confirmationView.php

```

<!-- Author : Ping Tian-Sen Anthony & Tas Emine -->
<!-- View for the confirmation of the reservation -->

```

```

<[DOCTYPE html]

<meta charset="UTF-8"/>
<head>

```

```
<title> CONFIRMATION DES RESERVATIONS </title>
<h2><center> Confirmations des réservations </center></h2>
<link rel="stylesheet" href='styles.css' />
</head>
<body>
  <div id="container">

    <p> Votre demande à bien été enregistrée. <br></p>
    <p> Merci de bien vouloir verser la somme de : </p>

    <center>
      <span class="style">
        <?php
          $price = unserialize($_SESSION['price']);
          echo $price->getPrices();echo ' €';
        ?>
      </span>
    </center>

    <p> sur le compte : <center><span class="style">
      000-000000-00 <br>
    </span></center>
  </div>

  <section>
    <center>
      <form method="post" action="index.php">
        <input type="hidden" name="page" value="controls" />
        <a href="#" class="button">
          <label>
            Retour Accueil
            <input type="submit" name = 'confirme' style="display:none" />
          </label>
        </a>
      </form>
    </center>
  </section>

</body>
</html>
```

8. manager.php

```
<!-- Author : Ping Tian-Sen Anthony & Tas Emine -->
```

```
<!--View for the manager -->
```

```
<!DOCTYPE html>
```

```
<meta charset="UTF-8"/>
```

```
<head>
```

```
<title> LISTE DES RESERVATIONS </title>
```

```
<h2><center> Liste des réservations</br> <i><font size=6>[For Manager]</font></i>
```

```
</center></h2>
```

```
<link rel="stylesheet" href='styles.css' />
```

```
</head>
```

```
<body>
```

```
<section>
```

```
<form method="post" action="index.php">
```

```
<center>
```

```
<a href="#" class="button">
```

```
<label>
```

```
Add Reservation
```

```
<input type="submit" name = 'ajoutReserv'  
style="display:none" />
```

```
</label>
```

```
</a>
```

```
</center>
```

```
</form>
```

```
<br>
```

```
</section>
```

```
<!-- Table creation -->
```

```
<center>
```

```
<table>
```

```
<tr>
```

```
<!-- Columns -->
```

```
<th>Id</th>
```

```
<th>Destination</th>
```

```
<th>Assurance</th>
```

```
<th>Total</th>
```

```
<th>Nom - Age</th>
```

```
<th>Editer</th>
```

```
<th>Supprimer</th>
```

```
</tr>
```

```
<!-- Table content -->
```

```
<?php
```

```
$mysqli = new mysqli("localhost", "root",  
"", "reservation") or die("Could not select database");
```

```
if ($mysqli->connect_errno) {
```

```
echo "Echec lors de la connexion à MySQL :(".
```

```
$mysqli->connect_errno . ")".$mysqli->connect_error;
```

```
}else{
```



```
$query = "SELECT * FROM reservation";
$result = $mysqli->query($query) or die("Query failed ");
```

```
while ($line = $result->fetch_array(MYSQLI_ASSOC))
```

```
{
    $id=$line['Id'];
    echo "</tr>\n";
    echo "<td><center>".$line['Id'].</center></td>";
    echo "<td><center>".
        $line['Destination'].</center></td>";
    echo "<td><center>".$line['Assurance'].</center></td>";
    echo "<td><center>".$line['Total'].</center></td>";
    echo "<td><center>".$line['NomAge'].</center></td>";
    echo "<td>
<form method='post' action='index.php'>
<center>
<a href='#' class='button'>
<label>
    Editer
    <input type='submit' name = 'edition'
        style='display:none 'value=$id />
    <input type='hidden' name='page'
        value='controls' />
</label>
</a>
</center>
</form></td>";
    echo "<td>
<form method='post' action='index.php'>
<center>
<a href=# class='button'>
<label>
    Supprimer
    <input type='submit' name = 'delete'
        style='display:none 'value=$id />
    <input type='hidden' name='page' value='controls' />
</label>
</a>
</center>
</form></td>";
    echo "</tr>\n";
}
}
```

```
?>

</table>
</center>
</body>
</html>
```

9. controls.php

```
<!-- Author : Ping Tian-Sen Anthony & Tas Emine -->
```

```
<?php
```

```
/**
```

```
The scope of this class is to manage each event that happened on every webpage.
```

```
*/
```

```
include 'reservation.php';
include 'exception_.php';
include 'detailModel.php';
include 'validateModel.php';
```

```
//Attributs
```

```
$exception = new exception_;
```

```
$step=1;
```

```
$detail;
```

```
/**
```

```
* Check session status.
```

```
*/
```

```
if (!isset($_SESSION["reservation"])) {
    $reservation = new reservation;
    $_SESSION['reservation'] = serialize($reservation);
} else {
    $reservation = unserialize($_SESSION['reservation']);
    //$detail = unserialize($_SESSION['detail']);
}
```

```
/**
```

```
* Manage bouton event
```

```
*/
```

```
if (isset($_POST['nextR'])) {
```

```
    $number = $_POST['number_of_places'];
```

```
    $destination = $_POST['destination'];
```

```
    if (($exception->isANumber($number)) && (!$exception->isEmpty($destination))){
```

```
        $reservation->setNumberOfPlaces($number);
```

```
        $reservation->setDestination($destination);
```

```
        if(isset($_POST['insurance'])){
```

```
            $reservation->setIsInsured(true);
```

```
        }else {
```

```
            $reservation->setIsInsured(false);
```

```
        }
```

```

    $_SESSION['reservation'] = serialize($reservation);

    $detail = new detail($reservation->getNumberOfPlaces());
    $_SESSION['detail'] = serialize($detail);

    $step = 2;

} else {

    $reservation->setNumberOfPlaces(0);
    $reservation->setDestination("Paris");
    $reservation->setIsError(true);

}

} elseif (isset($_POST['cancel'])) {
    $reservation->setNumberOfPlaces(0);
    $reservation->setDestination("Paris");
    session_destroy();
    $step=1;

} elseif (isset($_POST['nextD'])) {

    $listName = $_POST['nom'];
    $listAge = $_POST ['age'];
    $reservation = unserialize($_SESSION['reservation']);
    $detail = unserialize($_SESSION['detail']);
    $number=$reservation->getNumberOfPlaces();

    if ($exception->isNameOk($listName,$number) && $exception->isAValidAge($listAge,$number)){

        $detail->createPeople($listName,$listAge);
        $_SESSION['reservation'] = serialize($reservation);
        $_SESSION['detail'] = serialize($detail);
        $step=3;

    } else {

        $detail->setIsError(true);
        $step=2;

    }

} elseif (isset($_POST['previousD'])) {

    $reservation = unserialize($_SESSION['reservation']);
    $step=1;

} elseif (isset($_POST['validate'])) {

    $step=4;

} elseif (isset($_POST['previousV'])) {

```

```

    $detail=unserialize($_SESSION['detail']);
    $step=2;

}elseif (isset ($_POST['edition'])) {

    $_SESSION['id'] = serialize($_POST['edition']);

}elseif(isset($_POST['delete'])) {

    if (!empty($_POST['delete'])) {
        $id= $_POST['delete'];
    }

    if (!isset($_SESSION["sql"])) {
        $mysqli = new mysqli("localhost", "root", "", "reservation") or die
            ("Could not select database");
        if($mysqli->connect_errno){
            echo 'Connection failed : ('.$mysqli->connect_errno .')'.
                $mysqli->connect_errno;
        }
    }

    if($stmt = $mysqli->prepare ("Delete FROM `reservation` Where Id = ?")){
        $stmt->bind_param("i", $id);
        if (!$stmt->execute()){

            echo $stmt->error;

        }
    }else {

        echo $mysqli->error;

    }

    header ('Location:index.php?page=manager');
}

elseif (isset($_POST['addReserv'])) {

    $step=1;

}

elseif (isset($_POST['confirme'])) {

    if (!isset($_SESSION["sql"])) {
        $mysqli = new mysqli("localhost", "root", "", "reservation") or die ("Could not select database");
        if($mysqli->connect_errno){
            echo 'Connection failed : ('.$mysqli->connect_errno .')'. $mysqli->connect_errno;
        }
    }
}

```

```

$peopleString='';
$isInsured='FALSE';
$dest = $reservation->getDestination();
$insur= $reservation->getIsInsured();
$v= unserialize($_SESSION['price']);
$detail = unserialize($_SESSION['detail']);
$tot=$v->getPrices();
$p = $detail->getListPeople()[0];

for($i = 0; $i<$reservation->getNumberOfPlaces(); $i++){
    $people = $detail->getListPeople()[$i];
    $peopleString.= $people->getName();
    $peopleString.= ' - '.$people->getAge();
    $peopleString.="\n<br>";
}

if($insur){
    $isInsured='TRUE';
}

if(isset($_SESSION["id"])){
    $id=unserialize($_SESSION['id']);
    $query="UPDATE `reservation` SET Destination = ?, Assurance=?, Total=?, NomAge=? WHERE Id=?";
    if($stmt = $mysqli->prepare ("UPDATE reservation SET Destination = ?, Assurance=?, Total=?,
NomAge=? WHERE Id=?")){
        $stmt -> bind_param("ssisi", $dest, $isInsured, $tot, $peopleString,$id);
        if (!$stmt -> execute()){
            echo $stmt->error;
        }
        unset($_SESSION['id']);
    }else {
        echo $mysqli->error;
    }
}

}else{
    $query="INSERT INTO `reservation` (`Destination`, `Assurance`, `Total`, `NomAge`) VALUES (?, ?, ?, ?)";
    if($stmt = $mysqli->prepare ($query)){
        $stmt -> bind_param("ssisi", $dest, $isInsured, $tot, $peopleString);
        if (!$stmt -> execute()){
            echo $stmt->error;
        }
    }else {
        echo $mysqli->error;
    }
}

$_SESSION['step'] = serialize($step);

switch ($step){

```



```

case 1 :
    include 'main.php';
    break;

case 2 :
    include 'detailView.php';
    ;break;

case 3 :
    $reservation = unserialize($_SESSION['reservation']);
    $detail = unserialize($_SESSION['detail']);
    include 'validationView.php';
    ;break;

case 4:
    $val = new validate;
    $detail = unserialize($_SESSION['detail']);
    $reservation = unserialize($_SESSION['reservation']);
    $val->calculate($detail->getListPeople());
    $val->isInsured($reservation->getIsInsured());
    $val->getPrices();
    $_SESSION['price'] = serialize($val);
    include 'confirmationView.php';
    ;break;

}

?>

```

10.index.php

```

<!-- Author : Ping Tian-Sen Anthony & Tas Emine -->
<!-- Routing function-->

<?php

    session_start();

    if (!empty($_GET['page']) && is_file($_GET['page'].'.php')){
        include $_GET['page'].'.php';
    }else {
        include 'controls.php';
    }

?>

```

11. exception.php

```
<!-- Author : Ping Tian-Sen Anthony & Tas Emine -->
```

```
<?php
```

```
/* Exception class that will manage all error
*/
```

```
class exception_ {
```

```
    /**
     *Function that check if the param is a number
     *@param number, number to test
     *@return True if is a number else False
     */
    public function isANumber($number)
    {
        return (is_numeric($number) && ($number>0 && $number <10));
    }

    /**
     *Function that check if the param is empty
     *@param string, value to test
     *@return True if is empty else False
     */
    public function isEmpty($string){
        return empty($string);
    }

    /**
     *Function that check if the age is logic
     *@param listAge, age of each participant of a trip
     *@param numberPeople, number of participant
     *@return True if the age is correct else False
     */
    public function isAValidAge($listAge,$numberPeople){
        $isOk=true;
        $i=0;
        while($i<$numberPeople && ($isOk) ){
            $number=$listAge[$i];
            $isOk=(is_numeric($number) && ($number > 0 && $number < 120));
            $i++;
        }
        return $isOk;
    }

    /**
     *Function that check if the name is correct
     *@param listName, name of each participant
     *@param numberPeople, number of participant
     *@return True if the name is correct else False
     */
```

```

*/
public function isNameOk($listName,$numberPeople){
    $isOk=true;
    $i=0;
    while($i<$numberPeople && ($isOk) ){
        $isOk=!($this->isEmpty($listName[$i]));
        $i++;
    }
    return $isOk;
}

}

?>

```

12. message.php

<!-- Author : Ping Tian-Sen Anthony & Tas Emine -->

<?php

/* Message class.
This class create and modify message. */

class message_{

private \$message;

/**
*Constructor
*@param mess, the message
*/
public function __construct(\$mess){
 \$this->message=\$mess;
}

/**
*Getter of the message
*@return the message
*/

public function getMessage(){
 return \$this->message;
}

/**
*Setter of the message
*@param mess, the new message
*/

public function setMessage(\$mess){

```
$this->message=$mess;  
}  
  
}  
  
?>
```

13. people.php

<!-- Author : Ping Tian-Sen Anthony & Tas Emine -->

<?php

/* Class for a people.
This class has all the data about one person. */

class people{

private \$age=0;
private \$name=" ";

/*
*Constructor for a people
*@param age
*@param name
*/
public function __construct(\$age,\$name)
{
 \$this->age = \$age;
 \$this->name=\$name;
}

/**
*Getter for a name
*return a name
*/
public function getName()
{
 return \$this->name;
}

/**
*Getter for a age
*return a age
*/
public function getAge()
{
 return \$this->age;
}

/**
*Setter for a name

```

    *@param name the new name
    */
    public function setName($name)
    {
        $this->name=$name;
    }

```

```

        /**
        *Setter for a age
        *@param age, the new age
        */
        public function setAge($age)
        {
            $this->age=$age;
        }

```

```

    }

```

```

?>

```

styles.css

```

/* Author : Ping Tian-Sen Anthony & Tas Emine */
/* style sheet */

@font-face { /* Definition of a new font LearningCurveProRegular */
    font-family: 'Sketch_Block';
    src: url('Sketch_Block.eot');
    src: url('Sketch_Block.eot?#iefix') format('embedded-opentype'),
        url('Sketch_Block.woff') format('woff'),
        url('Sketch_Block.ttf') format('truetype'),
        url('Sketch_Block.svg#LearningCurveProRegular') format('svg');
}

.errorMessage
{
    color: #FE0000;
    width: 250px;                                /*Text size*/
    margin: auto;
    text-align: justify;
    margin-bottom: 20px;
    margin-top : 90px;
    font-family: 'Sketch_Block', Arial, serif;
    font-size: large;
    text-shadow: 2px 2px 4px #318277;            /*Text color blurring*/
}

```

```
h1, h2
{
    color: #B52135;
    font-family: Impact;
    font-family: 'Sketch_Block', Arial, serif;
    font-size: 60px;
    text-shadow: 2px 2px 4px #000000;
}

a.button{
    background:url (picture/go.gif) ;
    color:#006A5C;
    line-height:60px;                                /*Interline spacing*/
    font-size :90%;                                   /*Passage with mouse*/
    display: inline-block;
    text-align: center;
}

a:hover.button{
    color:#CA243B;
    font-size: 100%;
}

.style
{
    color:red;
    font-size: 1.5em;
}

/*The beige box on the center*/
#container
{
    padding: 10px;/*Entoure bien le cadre*/
    background: #ffffBf;
    border-radius: 15px;
    font-family: 'Sketch_Block', Arial, serif;
    font-size: small;                                /*text size*/
    box-shadow: 6px 6px 0px #006A5C;                 /*box contour*/
    text-shadow: 2px 2px 4px #006A5C;               /*text contour*/
    opacity: 0.7;
    width: 25em;                                       /*size of box*/
    margin-left: auto;
    margin-right: auto;
}

/*For align the text box*/
label
{
    display: block;
    width: 150px;
    float: left;
}
```

