Programmeercode

Versie 1.0.0

Inhoudsopgave

Reservation.php

```php
<?php

    /* Require classes */
    require_once "Query.php";
    require_once "Database.php";


    $hostname = 'localhost';
    $username = 'root';
    $password = '';
    $dbname   = 'taste';
    $charset  = 'utf8';


    // Database connection
    $Database = new Database($hostname, $username, $password, $dbname,
$charset);



    if ( isset($_GET['action']) )
    {
        switch ($_GET['action']) {
            case 'save':

                if ( isset($_GET['type']) && $_GET['type'] == 'reservation' )
                {
                    if ( !empty($_POST['reservation']) &&
!empty($_POST['customer']) )
                    {
                        $_POST['reservation']['Klant_ID'] = $Database-
>Insert("klanten", $_POST['customer']);
                        $Database->Insert("reserveringen",
$_POST['reservation']);
                    }
                }

            break;
        }

        header("location: reserveringen.php");
        exit();
    }

echo '
    <head>
```

```
        <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css
" integrity="sha384-
Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh"
crossorigin="anonymous">
    </head>
';

echo '
    <div class="container mt-5">
    <h1>Maak een nieuwe reservering</h1>
    <form action="?action=save&type=reservation" method="post">

        <input type="hidden" name="reservation[Datum_toegevoegd]" value="'.
date("Y-m-d H:i:s") .'">
        <input type="hidden" name="reservation[Status]" value="1">

        <input class="form-control mb-3" type="text" name="customer[Naam]"
placeholder="Naam" required>
        <input class="form-control mb-3" type="phone"
name="customer[Telefoon]" placeholder="Telefoon" required>
        <input class="form-control mb-3" type="email" name="customer[Email]"
placeholder="Email" required>

        <input class="form-control mb-3" type="number"
name="reservation[Tafel]" placeholder="Tafelnummer" required>
        <input class="form-control mb-3" type="date" name="reservation[Datum]"
placeholder="Datum" required>
        <input class="form-control mb-3" type="time" name="reservation[Tijd]"
placeholder="Tijd" required>
        <input class="form-control mb-3" type="number"
name="reservation[Aantal]" placeholder="Aantal" required>

        <a class="btn btn-primary" href="reserveringen.php">Terug</a>
        <input class="btn btn-primary" type="submit" value="Reservering
plaatsen">
    </form>

    </div>
';
```

Query.php

```php
<?php

    /**
     * @link https://gist.github.com/junyi-
xie/6e6430d2ba69827cbab33408c8b1e745
    */

    class Query
    {
        /**
         * Select function.
         *
         * @param string $sql The SQL string to select.
         * @param array $data The data which will be used as parameters for
the SQL.
         * @param int $mode The PDO mode used while returning the query.
         * @param bool $row Use if amount of rows needs to be returned.
         * @param bool $fetch Set the fetch mode to be single or multiple.
         *
         * @return int|array
         */
        public function Select(string $sql, array $data = [], int $mode =
\PDO::FETCH_OBJ, bool $row = false, bool $fetch = false)
        {
            $statement = $this->pdo->prepare($sql);

            if (!empty($data)) {
                foreach ($data as $key => &$value) {
                    $statement->bindValue("$key", $value);
                }
            }

            if (!$statement->execute()) {
                throw new \Exception("Error: " . implode(',', $this->pdo-
>errorInfo()));
            } else if ($row) {
                return $statement->rowCount();
            } else {
                return !$fetch ? $statement->fetchAll($mode) : $statement-
>fetch($mode);
            }
        }


        /**
         * Insert function.
         *
         * @param string $table The table specified to insert data in.
         * @param array $data The data used to insert into the table.
         *
         * @return int
         */
        public function Insert(string $table, array $data)
        {
            ksort($data);

            $keys = implode('`, `', array_keys($data));
            $values = ':' . implode(', :', array_keys($data));

            $statement = $this->pdo->prepare("INSERT INTO $table (`$keys`)
VALUES ($values)");

            foreach ($data as $key => $value) {
                $statement->bindValue(":$key", $value);
            }

            if (!$statement->execute()) {
                throw new Exception("Error: " . implode(',', $this->pdo-
>errorInfo()));
            }

            return $this->pdo->lastInsertId();
        }


        /**
         * Update function.
         *
         * @param string $table The table specified to update from.
         * @param array $data The data used to update the previous rows.
         * @param string $where The condition for which rows should be
updated.
         * @param int $limit The maximum rows to get updated.
         *
         * @return int|null
         */
        public function Update(string $table, array $data, string $where, int
$limit = 1)
        {
            ksort($data);

            $fields = null;

            foreach ($data as $key => $value) {
                $fields .= "`$key`= :$key,";
            }
```

```php
            $fields = rtrim($fields, ',');

            $statement = $this->pdo->prepare("UPDATE $table SET $fields WHERE
$where LIMIT $limit");

            foreach ($data as $key => $value) {
                $statement->bindValue(":$key", $value);
            }

            return $statement->execute();
        }


        /**
         * Delete function.
         *
         * @param string $table The table specified to get their items deleted
from.
         * @param string $where The condition for which rows should be
deleted.
         * @param int $limit The maximum rows to get deletd.
         *
         * @return bool
         */
        public function Delete(string $table, string $where, int $limit = 1)
        {
            return $this->pdo->prepare("DELETE FROM $table WHERE $where LIMIT
$limit")->execute();
        }
    }
```

Database.php

```php
<?php

    /**
     * @link https://gist.github.com/junyi-
xie/dba57c01a9b8b7cc075de05fa2b68d33
    */

    class Database extends Query
    {
        /**
         * The hostname.
         *
         * @var string
         */
        private $hostname;


        /**
         * The username.
         *
         * @var string
         */
        private $username;


        /**
         * The password.
         *
         * @var string
         */
        private $password;


        /**
         * The database.
         *
         * @var string
         */
        private $dbname;


        /**
         * The charset.
         *
         * @var string
         */
        private $charset;
        /**
         * PHP Data Object.
         *
         * @var object
         */
        protected $pdo;


        /**
         * Datbase Class Constructor.
         *
         * @param string $host The hostname.
         * @param string $user The username.
         * @param string $pass The password.
         * @param string $db The database.
         * @param string $charset The charset.
         *
         * @return void
         */
        public function __construct(string $host, string $user, string $pass,
string $db, string $charset)
        {
            $this->hostname = $host;
            $this->username = $user;
            $this->password = $pass;
            $this->dbname   = $db;
            $this->charset  = $charset;
            $this->connect();
        }


        /**
         * Connect to database.
         *
         * @return void
         */
        public function connect()
        {
            $dsn = "mysql:host=$this->hostname;dbname=$this-
>dbname;charset=$this->charset";
            $options = [
                PDO::ATTR_ERRMODE            => PDO::ERRMODE_EXCEPTION,
                PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC,
                PDO::ATTR_EMULATE_PREPARES   => false,
            ];
            try {
                $this->pdo = new \PDO($dsn, $this->username, $this->password,
$options);
            } catch (PDOException $e) {
                throw new PDOException($e->getMessage(), (int) $e->getCode());
            }
        }


        /**
         * Disconnect from database.
         *
         * @return void
         */
        public function disconnect()
        {
            $this->pdo = null;
        }


        /**
         * Database Class Destructor.
         *
         * @return void
         */
        public function __destruct()
        {
            $this->disconnect();
        }
    }
```

Overige broncode

Input fields validations v1

- Changed input type text to input type date
- Changed input type text to input type time

OLD:

<label>Reserveringsdatum (dd-mm-jjjj)</label>

<input type='text' name='reserveringsdatum' id='reserveringsdatum' value='" . $row['Datum'] . "' />

<label>Reserveringstijd (uu:mm)</label>

<input type='text' name='reserveringstijd' id='reserveringstijd' value='" . $row['Tijd'] . "' />

NEW:

<label>Reserveringsdatum (dd-mm-jjjj)</label>

<input type='date' name='reserveringsdatum' id='reserveringsdatum' value='" . $row['Datum'] . "' />

<label>Reserveringstijd (uu:mm)</label>

<input type='time' name='reserveringstijd' id='reserveringstijd' value='" . $row['Tijd'] . "' />

Adding price v1

- Added price for each product
- Added a totalprice that adds up all the products together

CODE:

$totaalprijs = 0.00;

Defined before the foreach so it adds up.

$prijs = $row['Prijs'] * $row['Aantal'];

$totaalprijs += $prijs;

<td>&euro;" . number_format((float)$prijs, 2) . "</td>

<tr><td>Totaal prijs:</td><td colspan='5'>&euro;". number_format((float)$totaalprijs, 2) ."</td></tr>

For each indivudial menuitem:

$rowmenuitem['Naam'] . " - &euro;". number_format((float)$rowmenuitem['Prijs'], 2)

Versiebeheer

Versie	**Door en datum**	**Opmerking**
1.0.0	Jun Yi Xie – 16 februari 2022	Document ingevuld