<?php

namespace Framework;

class DatabaseTable {

    private $table;

    private $pdo;

    private $primaryKey;

    public function \_\_construct($pdo, $table, $primaryKey) {

        $this->pdo = $pdo;

        $this->table = $table;

        $this->primaryKey = $primaryKey;

    }

public function find($field, $value)

{

    $stmt = $this->pdo->prepare('SELECT \* FROM ' . $this->table . ' WHERE ' . $field . ' = :value');

    $values = [

        'value' => $value

    ];

    $stmt->execute($values);

    return $stmt->fetch();

}

public function findMutiple($field, $value)

{

    $stmt = $this->pdo->prepare('SELECT \* FROM ' . $this->table . ' WHERE ' . $field . ' = :value');

    $values = [

        'value' => $value

    ];

    $stmt->execute($values);

    return $stmt->fetchall();

}

public function findAll()

{

    $stmt = $this->pdo->prepare('SELECT \* FROM ' . $this->table);

    $stmt->execute();

    return $stmt->fetchall();

}

public function findMostRecent($value, $number)

{

    $stmt = $this->pdo->prepare('SELECT \* FROM ' . $this->table . ' ORDER BY ' . $value . ' DESC LIMIT ' . $number);

    $stmt->execute();

    return $stmt->fetchall();

}

public function delete($field, $value)

{

    $stmt = $this->pdo->prepare('DELETE FROM ' . $this->table . ' WHERE ' . $field . ' = :value');

    $values = [

        'value' => $value

    ];

    $stmt->execute($values);

}

public function deleteAll()

{

    $stmt = $this->pdo->prepare('DELETE FROM ' . $this->table . '');

    $stmt->execute();

}

public function countAll() {

    $stmt = $this->pdo->prepare('SELECT COUNT(\*) FROM ' . $this->table);

    $stmt->execute();

    return $stmt->fetchColumn();

}

public function count($field, $value) {

    $stmt = $this->pdo->prepare('SELECT COUNT(\*) FROM ' . $this->table. ' WHERE ' . $field . ' = :value');

    $values = [

        'value' => $value

    ];

    $stmt->execute($values);

    return $stmt->fetchColumn();

}

public function insert($record)

{

    $keys = array\_keys($record);

    $values = implode(',', $keys);

    $valuesWithColon = implode(', :', $keys);

    $query = 'INSERT INTO ' . $this->table . ' (' . $values . ') VALUES (:' . $valuesWithColon . ')';

    $stmt = $this->pdo->prepare($query);

    $stmt->execute($record);

}

public function update($record) {

    $query = 'UPDATE ' . $this->table . ' SET ';

    $parameters = [];

    foreach($record as $key => $value) {

        $parameters[] = $key . ' = :' .$key;

    }

    $query .= implode(', ', $parameters);

    $query .= ' WHERE ' . $this->primaryKey . ' = :primaryKey';

    $record['primaryKey'] = $record[$this->primaryKey];

    $stmt = $this->pdo->prepare($query);

    $stmt->execute($record);

}

public function save($record) {

    if (empty($record[$this->primaryKey])) {

        unset($record[$this->primaryKey]);

    }

    try {

        $this->insert($record);

    }

    catch (\Exception $e) {

        $this->update($record);

    }

}

}

<?php

namespace Framework;

class EntryPoint

{

    private $routes;

    public function \_\_construct(\Framework\Routes $routes)

    {

        $this->routes = $routes;

    }

    public function run($contents)

    {

        $route = ltrim(explode('?', $\_SERVER['REQUEST\_URI'])[0], '/');

        $page = $this->routes->getPage($route);

        $output = $this->loadTemplate('../templates/' . $page['template'] . '.html.php', $page['variables']);

        $title = $page['title'];

        require '../templates/layout.html.php';

    }

    public function loadTemplate($fileName, $templateVars)

    {

        extract($templateVars);

        ob\_start();

        require $fileName;

        $contents = ob\_get\_clean();

        return $contents;

    }

    public function load\_File($file\_name)

    {

        ob\_start();

        require $file\_name;

        $contents = ob\_get\_clean();

        return $contents;

    }

    public function load\_header($file\_name)

    {

        header('location: ' . $file\_name);

    }

}

<?php

namespace Framework;

interface Routes {

    public function getPage($route);

}

@import url('https://stockfont.org/?2a2fa737785a618407ef19d9fc9611a6db7131bb753f53f1b6d96e112be43c32');

\* {margin: 0; padding: 0;}

body {background-color: #DEDEDE; font-family: 'Oxygen-Regular';

    display: grid;

  grid-template-columns: 10% 70% 20%;

    grid-template-rows: auto;

    grid-template-areas: "header header header"

    "img img aside"

    ". main aside"

    "footer footer footer";

}

h1 { font-family: 'Audiowide', cursive; margin: 0.4em;}

aside h1 a {color: white;}

header h1:before {content: "\1F5B3"; font-family: 'hack'; padding-right: 0.4em;}

header h1 {color: #92B2BF; text-shadow: 2px 2px 2px darkgreen;}

body > section {grid-area: img; height: 20vh; background-position: center center;

    background-image: url(../banners/1.jpg);

    max-height: 300px;

    background-position: center center;}

header a { color: white; text-decoration: none; }

header address {float: right; color: white; font-style: normal; padding: 1em;}

main {min-height: 20vh; grid-area: main; padding: 5vw; background-color: #E8E8E8;}

header {background-color: #1C2524; grid-area: header;}

main h1, main a h2 {color: #92B2BF;}

main a {text-decoration: none;}

main a h3 {color: #000000; text-decoration: none; font-size: 1.2em;}

aside {background-color: #2A3F52; min-height: 20px; grid-area: aside; padding: 1vw; color: white;}

aside ul {list-style-type: none; margin-bottom: 1em;}

aside li {padding-left: 20px;}

aside > ul > li {font-size: 2em;}

aside > ul > li > ul {font-size: 0.5em;}

aside a {color: #333; text-decoration: none}

footer {color: white; padding: 1em; background-color: black; font-size: 0.7em; grid-area: footer;}

header { display: grid;  grid-template-columns: 10% 80% 10%;}

header ul {align-self: center; display: grid; grid-template-columns: auto auto auto; grid-template-rows: auto; color: white; list-style-type: none;}

header > ul > li {font-size: 2em; cursor: pointer; text-align: center;}

header ul li ul { margin-left: 10vw; text-align: left; display: none; position: absolute; background-color: #333; padding: 20px;}

header ul li ul li {padding: 1vw;}

header ul li:hover ul {display: block;}

h1 {font-weight: bold; margin-bottom: 1em;}

p {margin-top: 0.5em; margin-bottom: 0.5em;}

img { width: 6em; height: 5em; object-fit: contain; margin: 0.5em 1em 1em 2.5em; padding: 1.5em; float:right; border-radius: 2em; background-color: #222222;}

h2 {margin-top: 0.5em;  margin-bottom: 0.5em;}

form label, form input, form textarea { box-sizing: border-box;  display: block; width: 42%; padding: 2%; margin-left: 1em;}

form textarea {height: 200px; float: left;}

form {overflow: auto;}

input[type="submit"] {width: 60%; display: inline-block; border: 0.1em solid #ccc; border-radius: 1%; box-sizing: border-box; margin-top: 1.2em; margin-bottom: 2em;}

input[type="text"], input[type="password"] { width: 60%; display: inline-block; border: 0.1em solid #ccc; border-radius: 1%; box-sizing: border-box;}

select { width: 40%; border: 0.1em solid #ccc; border-radius: 1%; box-sizing: border-box; margin-top: 1em; margin-bottom: 1em; margin-left: 1em; padding: 1em}

input[type="radio"] { margin-right: 3em; float: right; width: 10%; margin-top: 2em;}

.error {color: red}

.success {color: green}

.products, .reviews {list-style-type: none;}

.products li, .reviews li {padding: 2em;}

.products li:nth-child(odd), .reviews li:nth-child(odd) {background-color: #fff;}

.products li:nth-child(even), .products li:nth-child(even) {background-color: #ddd;}

.price { text-align: right; color: darkgreen; font-family: 'Audiowide'; font-weight: bold; font-size: 2em;}

.reviews .details {text-align: right;}

h4 {margin-top: 1em; margin-bottom: 1em;}

<?php

session\_start();

session\_regenerate\_id(true);

require '../autoload.php';

$routes = new \techstore\Routes();

$contents = new \techstore\Content();

$entryPoint = new \Framework\EntryPoint($routes);

$contents = $contents->fetchData();

$entryPoint->run($contents);

<?php

namespace techstore\Controllers;

class Admin

{

    private $functions;

    public function \_\_construct($functions)

    {

        $this->functions = $functions;

    }

    public function login()

    {

        $this->functions->AlreadyLoggedIn();

        $error = $this->functions->signinUser('admin');

        return [

            'title' => 'Login Portal',

            'template' => 'login',

            'variables' => [

                'loginErr' => $error,

                'signin' => 'admin'

            ]

        ];

    }

    public function create()

    {

        $this->functions->adminOnly();

        $error = $this->functions->createUser('admin');

        return [

            'title' => 'Manage Admins',

            'template' => 'register',

            'variables' => [

                'error' => $error,

                'type' => 'admin'

            ]

        ];

    }

    public function admincreated() {

        $this->functions->adminOnly();

        return [

            'title' => 'Admin Account Created',

            'template' => 'message',

            'variables' => [

                'title' => 'Admin Account Successfully Created',

                'message' => '<a href="/category/home" >Return to Homepage</a>'

            ]

        ];

    }

}

<?php

namespace techstore\Controllers;

class Category

{

    private $functions;

    public function \_\_construct($functions)

    {

        $this->functions = $functions;

    }

    public function category()

    {

        $this->functions->notFound('/category/nocategoryfound');

        $this->functions->productTable->findMutiple('category\_id', $\_GET['id']);

        if (isset($\_GET['id'])) {

           $products = $this->functions->productTable->findMutiple('category\_id', $\_GET['id']);

           $products\_count = $this->functions->productTable->count('category\_id', $\_GET['id']);

           $category = $this->functions->categoryTable->find('category\_id', $\_GET['id']);

        }

        $this->functions->valueEmpty($products, '/category/nocategoryfound');

        return [

            'title' => 'Products',

            'template' => 'home',

            'variables' => [

                'products' => $products,

                'products\_count' => $products\_count,

                'category' => $category,

                'page' => 'products'

            ]

        ];

    }

    public function managecategory()

    {

        $this->functions->adminOnly();

        $error = $msg = "";

        $categories = $this->functions->categoryTable->findAll();

        if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

            $error = $this->functions->notEmptyCheck($\_POST["name"], "A field was left blank!");

        }

        if ($error == "" && (isset($\_POST['submit']))) {

            $values = [

                "name" => $\_POST['name'],

            ];

            $this->functions->categoryTable->insert($values);

            header('location: /category/managecategory');

            $msg = 'The Category was successfully created!';

        }

        if (isset($\_GET['remove'])) {

            $product = $this->functions->productTable->find('product\_id', $\_GET['remove']);

            $category = $this->functions->categoryTable->find('category\_id', $product['category\_id']);

            $this->functions->valueEmpty($category, '/category/managecategory');

            $msg = 'The Product ' . $product['name'] . ' was successfully removed from the ' . $category['name'] . ' category';

            $values = [

                'product\_id' => $\_GET['remove'],

                'category\_id' => null

            ];

            $this->functions->productTable->update($values);

        }

        return [

            'title' => 'Manage Categories',

            'template' => 'managecategory',

            'variables' => [

                'error' => $error,

                'msg' => $msg,

                'productTable' => $this->functions->productTable,

                'categories' => $categories

            ]

        ];

    }

    public function home()

    {

        return [

            'title' => 'Tech Store Products',

            'template' => 'home',

            'variables' => [

                'products' => $this->functions->productTable->findMostRecent('product\_id', '10')

            ]

        ];

    }

    public function delete()

    {

        $this->functions->adminOnly();

        $category = $this->functions->categoryTable->find('category\_id', $\_GET['id']);

        $products = $this->functions->productTable->findMutiple('category\_id', $category['category\_id']);

        if (isset($\_GET['id']) && !(empty($category))) {

            foreach ($products as $product) {

                $values = [

                    "product\_id" => $product['product\_id'],

                    "category\_id" => null

                ];

                $this->functions->productTable->update($values);

            }

            $this->functions->categoryTable->delete('category\_id', $\_GET['id']);

            header("location: /category/deleted");

        }

        return [

            'title' => 'Delete Category',

            'template' => 'message',

            'variables' => []

        ];

    }

    public function deleted()

    {

        return [

            'title' => 'Category Deleted',

            'template' => 'message',

            'variables' => [

                'title' => 'Category Successfully Deleted',

                'message' => '<a href="/category/managecategory" >Return to Manage Category Page</a>'

            ]

        ];

    }

    public function NoCategoryFound()

    {

        return [

            'title' => 'No Category Found',

            'template' => 'message',

            'variables' => [

                'title' => 'Products Category not found or either contains no products',

                'message' => '<a href="/category/home" >Return to Homepage</a>'

            ]

        ];

    }

}

<?php

namespace techstore\Controllers;

class Customer

{

    private $functions;

    public function \_\_construct($functions)

    {

        $this->functions = $functions;

    }

    public function login()

    {

        $this->functions->AlreadyLoggedIn();

        $error = $this->functions->signinUser('customer');

        return [

            'title' => 'Login Portal',

            'template' => 'login',

            'variables' => [

                'loginErr' => $error,

                'signin' => 'customer'

            ]

        ];

    }

    public function register()

    {

        $this->functions->AlreadyLoggedIn();

       $error = $this->functions->createUser('customer');

        return [

            'title' => 'register',

            'template' => 'register',

            'variables' => [

                'error' => $error,

                'type' => 'customer'

            ]

        ];

    }

    public function signout()

    {

        $this->functions->signOut();

        return [

            'title' => 'Signed Out',

            'template' => 'message',

            'variables' => [

                'title' => 'Signed Out',

                'message' => '<a href="/category/home" >Return to HomePage</a>'

            ]

        ];

    }

    public function accessdenied()

    {

        return [

            'title' => 'Access Denied',

            'template' => 'message',

            'variables' => [

                'title' => 'Access Denied',

                'message' => '<a href="/category/home" >Return to Homepage</a>'

            ]

        ];

    }

    public function created()

    {

        return [

            'title' => 'Account Created',

            'template' => 'message',

            'variables' => [

                'title' => 'Account Successfully Created',

                'message' => '<a href="/customer/login" >Sign in into your Account</a>'

            ]

        ];

    }

}

<?php

namespace techstore\Controllers;

class Product

{

    private $functions;

    public function \_\_construct($functions)

    {

        $this->functions = $functions;

    }

    public function product()

    {

        $error = $msg = '';

        $this->functions->notFound('/product/noproductfound');

        if (isset($\_GET['id'])) {

           $product = $this->functions->productTable->find('product\_id', $\_GET['id']);

           $questions = $this->functions->questionTable->findMutiple('product\_id', $\_GET['id']);

        }

        $this->functions->valueEmpty($product, '/product/noproductfound');

        if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

            $error = $this->functions->notEmptyCheck($\_POST["question"], "Question cannot be blank!");

        }

        if ($error == '' && (($\_POST['submit'])?? '')) {

            $customer = $this->functions->customerTable->find('email', $\_SESSION['email'] ?? '');

            $values = [

                'customer\_id' => $customer['customer\_id'],

                'product\_id' => $\_GET['id'],

                'question' => $\_POST['question']

            ];

            $this->functions->questionTable->insert($values);

            $msg = "Your question was successfully submitted!";

        }

        return [

            'title' => 'Product',

            'template' => 'productpage',

            'variables' => [

                'product' => $product,

                'questions' => $questions,

                'error' => $error,

                'msg' => $msg,

                'noquestions' => $this->functions->notEmptyCheck($questions, "This Product has no Questions"),

                'customerTable' => $this->functions->customerTable,

                'adminTable' => $this->functions->adminTable,

            ]

        ];

    }

    public function manageproduct()

    {

        $this->functions->adminOnly();

        $error = $msg = "";

        if (isset($\_POST['submit'])) {

            $values = [

                "product\_id" => $\_POST['productselection'],

                "category\_id" => $\_POST['category'],

            ];

            $this->functions->productTable->update($values);

            $category = $this->functions->categoryTable->find('category\_id', $\_POST['category']);

            $msg = 'The Product was successfully added to the category of ' . $category['name'];

        }

        if (isset($\_POST['add'])) {

            if (isset($\_FILES["image"])) {

                $target\_dir = "../public/uploaded-images/";

                $target\_file = $target\_dir . basename($\_FILES["image"]["name"]);

                $imageFileType = strtolower(pathinfo($target\_file, PATHINFO\_EXTENSION));

            }

            if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

                if (

                    (file\_exists($target\_file)) || ($\_FILES["image"]["size"] > 500000) ||

                    ($imageFileType != "jpg" && $imageFileType != "png" && $imageFileType != "jpeg")

                ) {

                    $error = "Product Image is either invalid, already exists or is too large!";

                }

                if (empty($\_POST["name"]) || empty($\_POST["manufacturer"]) || empty($\_POST["price"]) || empty($\_POST["description"])) {

                    $error = "One or more fields were left blank!";

                }

                if (!(preg\_match('/^\d+\.\d+$/', $\_POST["price"]))) {

                    $error = "Price must be an integer with a decimal!";

                }

            }

            if ($error == "" && (isset($\_POST['add']))) {

                $values = [

                    "name" => $\_POST['name'],

                    "manufacturer" => $\_POST['manufacturer'],

                    "price" => $\_POST['price'],

                    "description" => $\_POST['description'],

                    "product\_picture\_img" => '../uploaded-images/' . basename($\_FILES["image"]["name"])

                ];

                move\_uploaded\_file($\_FILES["image"]["tmp\_name"], $target\_file);

                $this->functions->productTable->insert($values);

                $msg = 'The Product was successfully created!';

            }

        }

        if (isset($\_GET['allproducts'])) {

            $products = $this->functions->productTable->findAll();

        } else {

            $products = $this->functions->productTable->findMostRecent('product\_id', '5');

        }

        return [

            'title' => 'Manage Products',

            'template' => 'manageproduct',

            'variables' => [

                'error' => $error,

                'msg' => $msg,

                'recent\_products' => $products,

                'categories' => $this->functions->categoryTable->findall()

            ]

        ];

    }

    public function delete()

    {

        $this->functions->adminOnly();

        $product = $this->functions->productTable->find('product\_id', $\_GET['id']);

        $questions = $this->functions->questionTable->findMutiple('product\_id', $\_GET['id']);

        if (isset($\_GET['id']) && !(empty($product))) {

            unlink('../public' . substr($product['product\_picture\_img'], 2));

            foreach ($questions as $question) {

                $this->functions->questionTable->delete('product\_id', $question['product\_id']);

            }

            $this->functions->productTable->delete('product\_id', $\_GET['id']);

            header("location: /product/deleted");

        }

        return [

            'title' => 'Delete Product',

            'template' => 'message',

            'variables' => []

        ];

    }

    public function deleted()

    {

        return [

            'title' => 'Deleted',

            'template' => 'message',

            'variables' => [

                'title' => 'Product Successfully Deleted',

                'message' => '<a href="/product/manageproduct" >Return to Manage Products Page</a>'

            ]

        ];

    }

    public function NoProductFound()

    {

        return [

            'title' => 'No Product Found',

            'template' => 'message',

            'variables' => [

                'title' => 'Product Not Found',

                'message' => '<a href="/category/home" >Return to Homepage</a>'

            ]

        ];

    }

}

<?

namespace techstore\Controllers;

class Question

{

    private $functions;

    public function \_\_construct($functions)

    {

        $this->functions = $functions;

    }

    public function customer()

    {

        $this->functions->customerOnly();

        $customer = $this->functions->customerTable->find('email', $\_SESSION['email']);

        $questions = $this->functions->questionTable->findMutiple('customer\_id', $customer['customer\_id']);

        return [

            'title' => 'Questions',

            'template' => 'questions',

            'variables' => [

                'questions' => $questions,

                'noquestions' => $this->functions->notEmptyCheck($questions, "You have no Questions"),

                'adminTable' => $this->functions->adminTable,

                'productTable' => $this->functions->productTable

            ]

        ];

    }

    public function admin()

    {

        $this->functions->adminOnly();

        $error = $msg = '';

        if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

            $error = $this->functions->notEmptyCheck($\_POST["answer"], "Answer cannot be blank!");

        }

        if ($error == '' && (($\_POST['submit']) ?? '')) {

            $admin = $this->functions->adminTable->find('email', $\_SESSION['email']);

            $values = [

                'question\_id' => $\_POST['question\_id'],

                'admin\_id' => $admin['admin\_id'],

                'answer' => $\_POST['answer'],

                'status' => 'verified',

                'visibility' => $\_POST['visibility']

            ];

            $this->functions->questionTable->update($values);

            $msg = "Your Answer was successfully submitted!";

        }

        if ($\_GET['unanswered'] ?? '') {

            $questions = $this->functions->questionTable->findMutiple('status', 'pending');

            $title = 'Unanswered Questions';

        } else {

            $questions = $this->functions->questionTable->findAll();

            $title = 'All Questions';

        }

        return [

            'title' => $title,

            'template' => 'questions',

            'variables' => [

                'questions' => $questions,

                'error' => $error,

                'msg' => $msg,

                'noquestions' => $this->functions->notEmptyCheck($questions, "No Questions were Found"),

                'adminTable' => $this->functions->adminTable,

                'productTable' => $this->functions->productTable,

                'customerTable' => $this->functions->customerTable

            ]

        ];

    }

}

<?php

namespace techstore;

class Content

{

    public function fetchData()

    {

        require '../database.php';

        $categoryTable = new \Framework\DatabaseTable($pdo, 'category', 'category\_id');

        $productTable = new \Framework\DatabaseTable($pdo, 'products', 'product\_id');

        $categories = $categoryTable->findAll();

        $products = $productTable->findMostRecent('product\_id', '3');

        $product\_list = [];

        $product\_desc\_list = [];

        $product\_id\_list = [];

        foreach ($products as $product) {

            $product\_list[] = $product['name'];

            $product\_id\_list[] = $product['product\_id'];

            $product\_desc\_list[] = $product['description'];

        }

        $contents = [

            'categories' => $categories,

            'P1' => $product\_list[0],

            'P1\_id' => $product\_id\_list[0],

            'D1' => $product\_desc\_list[0],

            'P2' => $product\_list[1],

            'P2\_id' => $product\_id\_list[1],

            'D2' => $product\_desc\_list[1]

        ];

        return $contents;

    }

}

<?php

namespace techstore;

trait TablesFunctions

{

    public $adminTable;

    public $productTable;

    public $customerTable;

    public $categoryTable;

    public $questionTable;

    public function \_\_construct($adminTable, $productTable, $customerTable, $categoryTable, $questionTable)

    {

        $this->adminTable = $adminTable;

        $this->productTable = $productTable;

        $this->customerTable = $customerTable;

        $this->categoryTable = $categoryTable;

        $this->questionTable = $questionTable;

    }

    public function adminOnly()

    {

        if (!($\_SESSION['loggedin'] ?? '') || !($\_SESSION['account\_type'] == 'admin')) {

            header('location: /customer/accessdenied');

        }

    }

    public function customerOnly()

    {

        if (!($\_SESSION['loggedin'] ?? '') || $\_SESSION['account\_type'] == 'admin') {

            header('location: /customer/accessdenied');

        }

    }

    public function AlreadyLoggedIn()

    {

        if ($\_SESSION['loggedin'] ?? '') {

            header('location: /customer/accessdenied');

        }

    }

    public function notEmptyCheck($field, $value)

    {

        if (empty($field)) {

            $error = $value;

        } else {

            $error = "";

        }

        return $error;

    }

    public function findAccountInfo($value, $field)

    {

        if (($value == 'admin')) {

            $user = $this->adminTable->find('email', $field);

        } else {

            $user = $this->customerTable->find('email', $field);

        }

        return $user;

    }

    public function signinUser($value)

    {

        $error = "";

        if (isset($\_POST['submit'])) {

            $user = $this->findAccountInfo($value, $\_POST['email']);

            if ($user && password\_verify($\_POST['password'], $user['password'])) {

                $\_SESSION['loggedin'] = true;

                $\_SESSION['name'] = $user['firstname'] . ' ' . $user['surname'];

                $\_SESSION['account\_type'] = $value;

                $\_SESSION['email'] = $\_POST['email'];

                header('location: /category/home');

            } else {

                $error = 'You did not enter the correct email and password';

            }

        }

        return $error;

    }

    public function createUser($value)

    {

        $error = "";

        $token = bin2hex(random\_bytes(24));

        if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

            $user = $this->findAccountInfo($value, $\_POST['email']);

            if (strlen($\_POST["password"]) < 8 || strlen($\_POST["firstname"]) < 3 || strlen($\_POST["surname"]) < 3) {

                $error = 'Password or Names are too short!';

            }

            if (

                !(preg\_match("/^[a-zA-Z- ']\*$/", $\_POST["firstname"])) || !(preg\_match("/^[a-zA-Z- ']\*$/", $\_POST["surname"])) ||

                !(filter\_var($\_POST["email"], FILTER\_VALIDATE\_EMAIL)) || strtolower($user['email'] ?? '') == strtolower($\_POST['email'] ?? '')

            ) {

                $error = 'Names contains invalid characters or either the email is invalid!';

            }

            if (empty($\_POST["firstname"]) || empty($\_POST["surname"]) || empty($\_POST["email"]) || empty($\_POST["password"])) {

                $error = 'One or more fields were left blank!';

            }

            $\_SESSION['token'] = $token;

        }

        if ($error == "" && isset($\_SESSION['token']) == $token) {

            $date = new \DateTime();

            $date = $date->format("Y-m-d");

            $hash = password\_hash($\_POST['password'], PASSWORD\_DEFAULT);

            $values = [

                "firstname" => $\_POST['firstname'],

                "surname" => $\_POST['surname'],

                "email" => $\_POST['email'],

                "password" => $hash,

                "date\_created" => $date

            ];

            unset($\_SESSION['token']);

            if ($value == 'admin') {

                $this->adminTable->insert($values);

                header('location: /admin/admincreated');

            } else {

                $this->customerTable->insert($values);

                header('location: /customer/created');

            }

        } else {

            unset($\_SESSION['token']);

        }

        return $error;

    }

    public function notEmptyCheckMutiple($fields, $value)

    {

        foreach ($fields as $field) {

            if (empty($field)) {

                $error = $value;

            } else {

                $error = "";

            }

        }

        return $error;

    }

    public function signOut()

    {

        unset($\_SESSION['loggedin']);

        unset($\_SESSION['name']);

        unset($\_SESSION['account\_type']);

        unset($\_SESSION['email']);

    }

    public function notFound($value)

    {

        if ((!(isset($\_GET['id'])) || $\_GET['id'] == null || intval($\_GET['id']) == 0)) {

            header('location: ' . $value . '');

        }

    }

    public function valueEmpty($field, $value)

    {

        if (empty($field) || $field == null) {

            header('location: ' . $value . '');

        }

    }

}

class Functions

{

    use TablesFunctions;

}

<?php

namespace techstore;

class Routes implements \Framework\Routes

{

    public function getPage($route)

    {

        require '../database.php';

        $productTable = new \Framework\DatabaseTable($pdo, 'products', 'product\_id');

        $categoryTable = new \Framework\DatabaseTable($pdo, 'category', 'category\_id');

        $customerTable = new \Framework\DatabaseTable($pdo, 'customers', 'customer\_id');

        $adminTable = new \Framework\DatabaseTable($pdo, 'admins', 'admin\_id');

        $questionTable = new \Framework\DatabaseTable($pdo, 'questions', 'question\_id');

        $functions = new \techstore\Functions($adminTable, $productTable, $customerTable, $categoryTable, $questionTable);

        $controllers = [];

        $controllers['product'] = new \techstore\Controllers\Product($functions);

        $controllers['category'] = new \techstore\Controllers\Category($functions);

        $controllers['customer'] = new \techstore\Controllers\Customer($functions);

        $controllers['admin'] = new \techstore\Controllers\Admin($functions);

        $controllers['question'] = new \techstore\Controllers\Question($functions);

        if ($route == '') {

            $page = $controllers['category']->home();

        } else {

            list($controllerName, $functionName) = explode('/', $route);

            $controller = $controllers[$controllerName];

            $page = $controller->$functionName();

        }

        return $page;

    }

}

<? if ($page ?? '' == 'products') { ?>

    <h1><?= $category['name'] ?></h1>

    <p>A Number of <?= $products\_count ?> products were found</p>

<? } else { ?>

    <h1>Welcome to Ed's Electronics</h1>

    <p>We stock a large variety of electrical goods including phones, tvs, computers and games. Everything comes with at least a one year guarantee and free next day delivery.</p>

<? } ?>

<hr />

    <br>

    <h2>Products list</h2>

<ul class="products">

    <? foreach ($products as $product) { ?>

        <li>

        <a href="/product/product?id=<?=$product['product\_id']?>">

            <img src="<?= $product['product\_picture\_img'] ?>" alt="product-img" />

           <h3><?= $product['name'] ?></h3></a>

            <p><?= $product['description'] ?></p>

            <div class="price">£<?= $product['price'] ?></div>

        </li><? } ?>

</ul>

<!doctype html>

<html>

<head>

    <title><?= $title ?></title>

    <meta charset="utf-8" />

    <link rel="stylesheet" href="../CSS-Files/electronics.css" />

</head>

<body>

    <header>

        <h1>Ed's Electronics</h1>

        <ul>

            <li><a href="/category/home">Home</a></li>

            <li>Products

                <ul>

                    <? foreach ($contents['categories'] as $category) { ?>

                        <li><a href="/category/category?id=<?= $category['category\_id'] ?>"><?= $category['name'] ?></a></li>

                    <? } ?>

                </ul>

            </li>

            <li>Options

                <ul>

                    <? if (isset($\_SESSION['loggedin'])) { ?>

                        <? if ($\_SESSION['account\_type'] == 'admin') { ?>

                            <li><a href="/question/admin">Questions</a></li>

                            <li><a href="/admin/create">Manage Admins</a></li>

                            <li><a href="/product/manageproduct">Manage Products</a></li>

                            <li><a href="/category/managecategory">Manage Categories</a></li>

                        <? } else { ?>

                            <li><a href="/question/customer">Questions</a></li>

                        <? } ?>

                        <li><a href="/customer/signout">Sign Out</a></li>

                    <? } else { ?>

                        <li><a href="/customer/login">Sign in</a></li>

                        <li><a href="/customer/register">Create Account</a></li>

                        <li><a href="/admin/login">Admin</a></li>

                    <? } ?>

                </ul>

            </li>

        </ul>

        <address>

            <p>We are open 9-5, 7 days a week. Call us on

                <strong>01604 11111</strong>

            </p>

        </address>

    </header>

    <section></section>

    <main>

        <?= $output ?>

    </main>

    <aside>

        <h1><a href="/product/product?id=<?=$contents['P1\_id']?>">Featured Product</a></h1>

        <p><strong><?=$contents['P1']?></strong></p>

        <p><?=$contents['D1']?></p>

        <br>

        <h1><a href="/product/product?id=<?=$contents['P2\_id']?>">Product of the Day</a></h1>

        <p><strong><?=$contents['P2']?></strong></p>

        <p><?=$contents['D2']?></p>

    </aside>

    <footer>

        &copy; Ed's Electronics 2023

    </footer>

</body>

</html>

<? if ($signin == 'customer') { ?>

    <h1>Sign In</h1>

<? } else { ?>

    <h1>Admin Sign In</h1>

<? } ?>

<hr />

<form action="" method="post">

    <label>Email</label> <input type="text" name="email" placeholder="Your Email..." />

    <label>Password</label> <input type="password" name="password" placeholder="Your Password..." />

    <input type="submit" name="submit" value="submit" />

    <p class="error"><?= $loginErr ?><p>

</form>

<? if ($signin == 'customer') { ?>

    <a href="/customer/register">

        <h2>Don't have an account? Create one now!</h2>

    </a>

<? } ?>

<h1>Manage Categories</h1>

<p class="success"><?= $msg ?? '' ?>

<p>

    <hr />

    <br>

<h2>All categories:</h2>

<form action="" method="post">

    <ul class="products">

        <? foreach ($categories as $category) { ?>

            <li>

            <a href="/category/product?id=<?=$category['category\_id']?>"><h3><?=$category['name']?></h3></a>

                <? $products = $productTable->findMutiple('category\_id', $category['category\_id']) ?>

                <br>

                <? foreach ($products as $product) { ?>

                    <a href="/product/product?id=<?=$product['product\_id']?>"><p><?=$product['name']?></a>

                    <a class="error" href="/category/managecategory?remove=<?= $product['product\_id'] ?>"> &nbsp;&nbsp; Remove</a>

                </p>

                <? } ?>

                <br>

                <a class="error" href="/category/delete?id=<?= $category['category\_id'] ?>">Delete</a>

            </li>

        <? } ?>

        <br>

</form>

<br>

<hr />

<br>

<h2>Add a category</h2>

<form action="" method="post">

    <label>Category Name</label> <input type="text" name="name" />

    <input type="submit" name="submit" value="Add Category" />

    <p class="error"><?= $error ?? '' ?>

    <p>

</form>

<h1>Manage Products</h1>

<? if (isset($\_GET['allproducts'])) {?>

<a href="/product/manageproduct">

            <p>Filter to see most recent products</p></a>

<p class="success"><?=$msg ?? ''?><p>

            <hr />

<br>

<h2>All Products:</h2>

<? } else {?>

    <a href="/product/manageproduct?allproducts=true">

            <p>Filter to see all Products</p></a>

<p class="success"><?=$msg ?? ''?><p>

            <hr />

<br>

<h2>Most Recent Products Added:</h2>

    <? } ?>

<form action="" method="post">

<ul class="products">

<?  foreach ($recent\_products as $recent\_product) {?>

                <li>

                    <input type="radio" name="productselection" value="<?=$recent\_product['product\_id']?>" checked="checked">

                    <a href="/product/product?id=<?=$recent\_product['product\_id']?>"><h3><?=$recent\_product['name']?></h3></a>

                    <? if ($recent\_product['category\_id'] == null) { ?>

                        <br>

                        <a class="error">Product is not in a Category!</a>

                        <br>

                   <? }?>

                    <br>

                    <a class="error" href="/product/delete?id=<?=$recent\_product['product\_id']?>">Delete</a>

                </li>

                <? }?>

                </form>

<br>

<h2>Assign Product to a Category</h2>

<select name="category" id="category">

    <? foreach ($categories as $category) {?>

                        <option selected="selected" value="<?=$category['category\_id']?>"><?=$category['name']?></option>

                <? }?>

                </select>

<input type="submit" name="submit" value="Add Product to Category" />

</form>

<br>

            <hr />

            <br>

            <h2>Add a Product</h2>

            <form action="" method="post" enctype="multipart/form-data">

                <label>Product Name</label> <input type="text" name="name" />

                <label>Product Manufacturer</label> <input type="text" name="manufacturer" />

                <label>Product Price (Must Contain a Decimal)</label> <input type="text" name="price" />

                <label for="image">Product Image</label><input type="file" id="image" name="image">

                <label>Product Description</label> <textarea name="description" ></textarea>

                <input type="submit" name="add" value="Add Product" />

                <p class="error"><?=$error ?? ''?><p>

            </form>

<h1><?= $title?></h1>

<hr />

<h2><?= $message?></h2>

<?php

?>

<h1><?= $product['name'] ?></h1>

<p>Brand: <?= $product['manufacturer'] ?></p>

<hr />

<br>

<ul class="products">

    <li>

        <img src="<?= $product['product\_picture\_img'] ?>" alt="product-img" />

        <p><?= $product['description'] ?></p>

        <br>

        <br>

    </li>

</ul>

<br>

<hr />

<br>

<? if (isset($\_SESSION['loggedin']) && $\_SESSION['account\_type'] == 'customer') { ?>

    <h2>Submit a question</h2>

    <form action="" method="post">

        <br>

        <input type="text" name="question" placeholder="Your Question..." />

        <input type="submit" name="submit" value="submit" />

        <p class="error"><?=$error ?? ''?><p>

        <p class="success"><?=$msg ?? ''?><p>

    </form>

    <br>

    <hr>

<? } ?>

<h4>Product Questions</h4>

<ul class="reviews">

    <?= $noquestions ?? ''?>

    <? foreach ($questions as $question) { ?>

        <? if ($question['status'] == 'verified' && $question['visibility'] == 'yes') { ?>

            <li>

                <h3><?= $question['question'] ?></h3>

                <p><?=$question['answer']?></p>

                <div class="details">

                    <? $customer = $customerTable->find('customer\_id', $question['customer\_id']) ?>

                    <? $admin = $adminTable->find('admin\_id', $question['admin\_id']) ?>

                    <em>Date Posted: <?=$question['date\_posted']?></em><br>

                    <em>Submitted By: <strong> <?=$customer['firstname'] . ' '.$customer['surname']?></strong></em><br>

                    <em>Answered By: <strong> <?=$admin['firstname'] . ' '. $admin['surname']?></strong></em>

                </div>

            </li>

   <? } }?>

</ul>

<h1>Questions</h1>

<? if ($\_SESSION['account\_type'] == 'admin') {

    if ($\_GET['unanswered'] ?? '') { ?>

        <h4>Unanswered Questions</h4>

        <a href="/question/admin">

            <p>See all questions</p>

            <p class="success"><?=$msg ?? ''?><p>

        </a>

    <? } else { ?>

        <h4>All Questions</h4>

        <a href="/question/admin?unanswered=true">

            <p>Filter to see all unanswered questions</p>

            <p class="success"><?=$msg ?? ''?><p>

        </a>

    <? } ?>

<? } else { ?>

    <h4>Your Product Questions</h4>

<? } ?>

<hr />

<br>

<ul class="reviews">

    <?= $noquestions ?? '' ?>

    <? foreach ($questions as $question) { ?>

        <? $product = $productTable->find('product\_id', $question['product\_id']) ?>

        <li>

            <h3><?= $question['question'] ?></h3>

            <p>Product: <?= $product['name'] ?></p><br>

            <? if ($\_SESSION['account\_type'] == 'admin' && $question['status'] == 'pending') { ?>

                <form action="" method="post">

                    <br>

                    <input type="hidden" name="question\_id" value="<?=$question['question\_id']?>" />

                    <select name="visibility" id="visibility">

                        <option selected="selected" value="yes">Yes, Show this question</option>

                        <option value="no">No, Don't show this question</option>

                </select>

                    <input type="text" name="answer" placeholder="Your Answer..." />

                    <input type="submit" name="submit" value="submit" />

                    <p class="error"><?= $error ?? '' ?>

                    <p>

                    <p>

                </form>

            <? } else { ?>

                <p>Response: <?= $question['answer'] ?></p>

            <? } ?>

            <div class="details">

                <em>Status: <?= $question['status'] ?></em><br>

                <? if ($\_SESSION['account\_type'] == 'admin') {

                    $customer = $customerTable->find('customer\_id', $question['customer\_id']) ?>

                    <em>Visibility: <?= $question['visibility']?></em><br>

                    <em>Submitted By: <?= $customer['firstname'] . ' ' . $customer['surname'] ?></em><br>

                <? } ?>

                <em>Date Submitted: <?= $question['date\_posted'] ?></em><br>

                <?

                if ($question['status'] == 'verified' && $question['admin\_id'] ?? '') {

                    $admin = $adminTable->find('admin\_id', $question['admin\_id']) ?>

                    <strong><em>Answered By: <?= $admin['firstname'] . ' ' . $admin['surname'] ?></em></strong>

                <? } ?>

            </div>

        </li>

    <?  } ?>

</ul>

<? if ($type == 'admin') { ?>

<h1>Create a new admin account</h1>

<? } else { ?>

<h1>Create a new account</h1>

<? } ?>

<hr />

<form action="" method="post">

    <label>First Name</label> <input type="text" name="firstname" placeholder="Your First Name..."/>

    <label>Surname</label> <input type="text" name="surname" placeholder="Your Surname..."/>

    <label>Email</label> <input type="text" name="email" placeholder="Your Email..."/>

    <label>Password</label> <input type="password" name="password" placeholder="Your Password..."/>

    <input type="submit" name="submit" value="submit" />

    <p class="error"><?=$error?><p>

</form>

<? if ($type == 'customer') { ?>

<a href="/customer/login"><h2>Already have an account? Sign in now!</h2></a>

<? } ?>

<?php

function autoload($className) {

    require str\_replace('\\', '/', $className) . '.php';

}

spl\_autoload\_register('autoload');

<?php

$pdo = new PDO('mysql:host=mysql;dbname=techstoredb;charset=utf8', 'student', 'student');

-- MariaDB dump 10.19  Distrib 10.5.19-MariaDB, for Linux (x86\_64)

--

-- Host: mysql    Database: techstoredb

-- ------------------------------------------------------

-- Server version 10.11.6-MariaDB-1:10.11.6+maria~ubu2204

/\*!40101 SET @OLD\_CHARACTER\_SET\_CLIENT=@@CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET @OLD\_CHARACTER\_SET\_RESULTS=@@CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET @OLD\_COLLATION\_CONNECTION=@@COLLATION\_CONNECTION \*/;

/\*!40101 SET NAMES utf8mb4 \*/;

/\*!40103 SET @OLD\_TIME\_ZONE=@@TIME\_ZONE \*/;

/\*!40103 SET TIME\_ZONE='+00:00' \*/;

/\*!40014 SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0 \*/;

/\*!40014 SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0 \*/;

/\*!40101 SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='NO\_AUTO\_VALUE\_ON\_ZERO' \*/;

/\*!40111 SET @OLD\_SQL\_NOTES=@@SQL\_NOTES, SQL\_NOTES=0 \*/;

--

-- Current Database: `techstoredb`

--

CREATE DATABASE /\*!32312 IF NOT EXISTS\*/ `techstoredb` /\*!40100 DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4\_general\_ci \*/;

USE `techstoredb`;

--

-- Table structure for table `admins`

--

DROP TABLE IF EXISTS `admins`;

/\*!40101 SET @saved\_cs\_client     = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `admins` (

  `admin\_id` int(11) NOT NULL AUTO\_INCREMENT,

  `firstname` varchar(50) NOT NULL,

  `surname` varchar(50) NOT NULL,

  `password` varchar(150) NOT NULL,

  `email` varchar(50) NOT NULL,

  `date\_created` date NOT NULL DEFAULT current\_timestamp(),

  PRIMARY KEY (`admin\_id`)

) ENGINE=InnoDB AUTO\_INCREMENT=4 DEFAULT CHARSET=latin1 COLLATE=latin1\_swedish\_ci;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `admins`

--

LOCK TABLES `admins` WRITE;

/\*!40000 ALTER TABLE `admins` DISABLE KEYS \*/;

INSERT INTO `admins` VALUES (1,'admin','1','$2y$10$nak43Xll.3TPEB9SA.acve2xfW8Tky0rlNcmG1HZqIu48LxTnXL6O','testing00@gmail.com','2024-01-22'),(2,'admin','2','$2y$10$nak43Xll.3TPEB9SA.acve2xfW8Tky0rlNcmG1HZqIu48LxTnXL6O','testing01@gmail.com','2024-01-22');

/\*!40000 ALTER TABLE `admins` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `category`

--

DROP TABLE IF EXISTS `category`;

/\*!40101 SET @saved\_cs\_client     = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `category` (

  `category\_id` int(11) NOT NULL AUTO\_INCREMENT,

  `name` varchar(100) NOT NULL,

  `date\_created` date NOT NULL DEFAULT current\_timestamp(),

  PRIMARY KEY (`category\_id`)

) ENGINE=InnoDB AUTO\_INCREMENT=7 DEFAULT CHARSET=latin1 COLLATE=latin1\_swedish\_ci;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `category`

--

LOCK TABLES `category` WRITE;

/\*!40000 ALTER TABLE `category` DISABLE KEYS \*/;

INSERT INTO `category` VALUES (1,'TVs','2024-01-22'),(2,'Phones','2024-01-22'),(6,'PC\'s','2024-01-25');

/\*!40000 ALTER TABLE `category` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `customers`

--

DROP TABLE IF EXISTS `customers`;

/\*!40101 SET @saved\_cs\_client     = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `customers` (

  `customer\_id` int(11) NOT NULL AUTO\_INCREMENT,

  `firstname` varchar(50) NOT NULL,

  `surname` varchar(50) NOT NULL,

  `password` varchar(150) NOT NULL,

  `email` varchar(50) NOT NULL,

  `date\_created` date NOT NULL DEFAULT current\_timestamp(),

  PRIMARY KEY (`customer\_id`)

) ENGINE=InnoDB AUTO\_INCREMENT=8 DEFAULT CHARSET=latin1 COLLATE=latin1\_swedish\_ci;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `customers`

--

LOCK TABLES `customers` WRITE;

/\*!40000 ALTER TABLE `customers` DISABLE KEYS \*/;

INSERT INTO `customers` VALUES (1,'Josh','Allen','testing1','JoshAllen@gmail.com','2024-01-22'),(2,'James','Stock','testing2','JamesStock@gmail.com','2024-01-22'),(3,'Elliot','Smith','testing3','ElliotSmith@yahoo.com','2024-01-22'),(4,'Lily','Rose','testing4','LilyRose@outlook.com','2024-01-22');

/\*!40000 ALTER TABLE `customers` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `products`

--

DROP TABLE IF EXISTS `products`;

/\*!40101 SET @saved\_cs\_client     = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `products` (

  `product\_id` int(11) NOT NULL AUTO\_INCREMENT,

  `category\_id` int(11) DEFAULT NULL,

  `name` varchar(200) NOT NULL,

  `description` varchar(1000) DEFAULT NULL,

  `manufacturer` varchar(100) DEFAULT NULL,

  `price` decimal(6,2) NOT NULL,

  `date\_posted` date NOT NULL DEFAULT current\_timestamp(),

  `product\_picture\_img` varchar(150) DEFAULT '../images/banners/1.jpg',

  PRIMARY KEY (`product\_id`),

  KEY `category\_id` (`category\_id`),

  CONSTRAINT `products\_ibfk\_1` FOREIGN KEY (`category\_id`) REFERENCES `category` (`category\_id`)

) ENGINE=InnoDB AUTO\_INCREMENT=7 DEFAULT CHARSET=latin1 COLLATE=latin1\_swedish\_ci;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `products`

--

LOCK TABLES `products` WRITE;

/\*!40000 ALTER TABLE `products` DISABLE KEYS \*/;

INSERT INTO `products` VALUES (1,1,'Samsung CF390 27\" 16:9 Curved LCD FHD 1920x1080 Curved Desktop Black Monitor','Setting up this Samsung monitor is simple as it comes equipped with both HDMI and VGA inputs. Once configured \nviewers can take advantage of the Full HD 1920 x 1080 resolution display complete with a 3,000:1 contrast ratio, a 250 cd/m² brightness rating, and a 4 ms response time for a clear picture wi\nth reduced ghosting and blur. This 27\" Curved Samsung monitor can also be mounted to the wall using a 75 x 75mm mount, \nand with 178° horizontal and vertical viewing angles the screen can be viewed from virtually any angle.','Samsung',99.99,'2023-01-05','../uploaded-images/samsung\_tv.png'),(2,1,'LG FHD 32-Inch Computer Monitor 32ML600M-B, IPS with HDR 10 Compatibility, Black','Enjoy incredible detail and image clarity with Full HD 1080p resolution. This versatile IPS computer monitor\n helps keep colors consistent at a wide viewing angle while reproducing 95 percent coverage of the DCI-P3 color gamut. Color Calibration helps maintain accurate color on the screen and prevents \n gradual changes. Stay in the game as you take fast, easy control of essential monitor settings including picture, audio and Screen Split with just a few clicks of your mouse with On-Screen Control','LG',129.99,'2023-06-05','../uploaded-images/LG\_tv.png'),(3,2,'Iphone 14 Pro Max','iPhone 14 Pro Max. Capture incredible detail with a 48MP Main camera. Experience iPhone in a whole new way with Dynami\nc Island and Always-On display. Crash Detection, a new safety feature, calls for help when you can’t.','Apple',729.99,'2023-08-05','../uploaded-images/iphone\_14\_pro\_max.png'),(4,6,'Gaming PC','NVIDIA® GeForce RTX® 40 Series GPUs are beyond fast for gamers and creators. Theyre powered by the ultra-efficient NVIDIA Ada Lovelace architecture which delivers \na quantum leap in both performance and AI-powered graphics. Experience lifelike virtual worlds with ray tracing and ultra-high FPS gaming with the lowest latency. Discover revolutionary \nnew ways to create and unprecedented workflow acceleration.','PC Specialist',929.99,'2023-09-05','../uploaded-images/gaming\_pc.png');

/\*!40000 ALTER TABLE `products` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `questions`

--

DROP TABLE IF EXISTS `questions`;

/\*!40101 SET @saved\_cs\_client     = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `questions` (

  `question\_id` int(11) NOT NULL AUTO\_INCREMENT,

  `customer\_id` int(11) NOT NULL,

  `product\_id` int(11) NOT NULL,

  `admin\_id` int(11) DEFAULT NULL,

  `question` varchar(225) DEFAULT NULL,

  `answer` varchar(225) DEFAULT NULL,

  `status` char(40) NOT NULL DEFAULT 'pending',

  `visibility` char(40) NOT NULL DEFAULT 'yes',

  `date\_posted` date NOT NULL DEFAULT current\_timestamp(),

  PRIMARY KEY (`question\_id`),

  KEY `product\_id` (`product\_id`),

  KEY `customer\_id` (`customer\_id`),

  KEY `admin\_id` (`admin\_id`),

  CONSTRAINT `questions\_ibfk\_1` FOREIGN KEY (`product\_id`) REFERENCES `products` (`product\_id`),

  CONSTRAINT `questions\_ibfk\_2` FOREIGN KEY (`customer\_id`) REFERENCES `customers` (`customer\_id`),

  CONSTRAINT `questions\_ibfk\_3` FOREIGN KEY (`admin\_id`) REFERENCES `admins` (`admin\_id`)

) ENGINE=InnoDB AUTO\_INCREMENT=11 DEFAULT CHARSET=latin1 COLLATE=latin1\_swedish\_ci;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `questions`

--

LOCK TABLES `questions` WRITE;

/\*!40000 ALTER TABLE `questions` DISABLE KEYS \*/;

INSERT INTO `questions` VALUES (1,1,1,1,'Whats the size of this Monitor?','The Size of this monitor is 27 inches.','verified','yes','2024-01-22'),(2,3,1,2,'Does this monitor support HDR 10?','No, this monitor does not support HDR 10.','verified','yes','2024-01-22'),(3,3,1,1,'What is the Resolution of this Monitor?','Medion MS-7713 AMD ANTHON 64 X4 640 CPU 4GB RAM Motherboard Heatsink and IOplate','verified','no','2024-01-22'),(4,3,1,1,'Is this Monitor Curved?','','pending','no','2024-01-22'),(5,2,2,1,'Does this monitor support HDR 10?','Yes, this monitor does support HDR 10.','verified','yes','2024-01-22');

/\*!40000 ALTER TABLE `questions` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Table structure for table `reviews`

--

DROP TABLE IF EXISTS `reviews`;

/\*!40101 SET @saved\_cs\_client     = @@character\_set\_client \*/;

/\*!40101 SET character\_set\_client = utf8 \*/;

CREATE TABLE `reviews` (

  `review\_id` int(11) NOT NULL AUTO\_INCREMENT,

  `customer\_id` int(11) NOT NULL,

  `product\_id` int(11) NOT NULL,

  `title` varchar(225) DEFAULT NULL,

  `review` varchar(225) DEFAULT NULL,

  `rating` int(11) DEFAULT NULL,

  `status` char(40) NOT NULL DEFAULT 'pending',

  `visibility` char(40) NOT NULL DEFAULT 'yes',

  `date\_posted` date NOT NULL DEFAULT current\_timestamp(),

  PRIMARY KEY (`review\_id`),

  KEY `product\_id` (`product\_id`),

  KEY `customer\_id` (`customer\_id`),

  CONSTRAINT `reviews\_ibfk\_1` FOREIGN KEY (`product\_id`) REFERENCES `products` (`product\_id`),

  CONSTRAINT `reviews\_ibfk\_2` FOREIGN KEY (`customer\_id`) REFERENCES `customers` (`customer\_id`)

) ENGINE=InnoDB AUTO\_INCREMENT=2 DEFAULT CHARSET=latin1 COLLATE=latin1\_swedish\_ci;

/\*!40101 SET character\_set\_client = @saved\_cs\_client \*/;

--

-- Dumping data for table `reviews`

--

LOCK TABLES `reviews` WRITE;

/\*!40000 ALTER TABLE `reviews` DISABLE KEYS \*/;

INSERT INTO `reviews` VALUES (1,1,1,'Good monitor. Works on PC and Mac','I needed an extra monitor to manage the windows on my work computer. It works well. I say good because I am not a \npro on the ends and outs of a good monitor. Works on PC and Mac. .',4,'verified','yes','2024-01-22');

/\*!40000 ALTER TABLE `reviews` ENABLE KEYS \*/;

UNLOCK TABLES;

--

-- Dumping routines for database 'techstoredb'

--

/\*!40103 SET TIME\_ZONE=@OLD\_TIME\_ZONE \*/;

/\*!40101 SET SQL\_MODE=@OLD\_SQL\_MODE \*/;

/\*!40014 SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS \*/;

/\*!40014 SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS \*/;

/\*!40101 SET CHARACTER\_SET\_CLIENT=@OLD\_CHARACTER\_SET\_CLIENT \*/;

/\*!40101 SET CHARACTER\_SET\_RESULTS=@OLD\_CHARACTER\_SET\_RESULTS \*/;

/\*!40101 SET COLLATION\_CONNECTION=@OLD\_COLLATION\_CONNECTION \*/;

/\*!40111 SET SQL\_NOTES=@OLD\_SQL\_NOTES \*/;

-- Dump completed on 2024-01-25 21:18:06