

FoodieGo

MD.Mehedi hassan
Version

Table of Contents

Table of contents
Data Structure Index

Data Structures

Here are the data structures with brief descriptions:

Database (A singleton class to manage a database connection instance)	4
FoodController	6
FoodModel	7

File Index

File List

Here is a list of all files with brief descriptions:

database-config.php	8
food_menu_view.php	10
food_search_view.php	12
FoodController.php (Handles food-related business logic and communication between models and views)	14
foodModel.php (Contains functions for food-related database operations)	16
index.php	18
order-controller.php	20
order-model.php	22
order-view.php	24

Data Structure Documentation

Database Class Reference

A singleton class to manage a database connection instance.

Public Member Functions

- **getConnection ()**
Returns the active database connection.

Static Public Member Functions

- static **getInstance ()**
*Returns the single instance of the **Database** class.*

Detailed Description

A singleton class to manage a database connection instance.

This class follows the Singleton design pattern to ensure that only one instance of the database connection exists throughout the application's lifecycle. It provides methods to retrieve and use the database connection.

Definition at line **36** of file **database-config.php**.

Member Function Documentation

getConnection ()

Returns the active database connection.

This method allows other parts of the application to interact with the database by returning the active connection resource.

Returns

resource The active database connection.

<Returns the connection resource

Definition at line **99** of file **database-config.php**.

static getInstance () [static]

Returns the single instance of the **Database** class.

This method implements the Singleton pattern. If the instance does not already exist, it is created. Once created, the same instance is returned for subsequent calls.

Returns

Database The single instance of the **Database** class.

<Creates the instance if it doesn't exist

<Returns the existing instance

Definition at line **84** of file **database-config.php**.

The documentation for this class was generated from the following file:

database-config.php

FoodController Class Reference

Public Member Functions

- `__construct ()`
 - `searchFoods ($search)`
 - `showFoodMenu ()`
-

Detailed Description

Definition at line **9** of file **FoodController.php**.

Constructor & Destructor Documentation

`__construct ()`

Definition at line **12** of file **FoodController.php**.

Member Function Documentation

`searchFoods ($search)`

Render the search view based on the search keyword.

Definition at line **19** of file **FoodController.php**.

`showFoodMenu ()`

Render the food menu view with all active foods.

Definition at line **27** of file **FoodController.php**.

The documentation for this class was generated from the following file:

FoodController.php

FoodModel Class Reference

Public Member Functions

- **getAllActiveFoods** ()
- **searchFoods** (\$search)

Detailed Description

Definition at line **9** of file **foodModel.php**.

Member Function Documentation

getAllActiveFoods ()

Fetch all active foods.

Returns

array Result set containing food records.

Definition at line **14** of file **foodModel.php**.

searchFoods (\$search)

Search for foods based on a keyword.

Parameters

string	<i>\$search</i>	Keyword to search.
--------	-----------------	--------------------

Returns

array Result set of matching foods.

Definition at line **25** of file **foodModel.php**.

The documentation for this class was generated from the following file:
foodModel.php

File Documentation

database-config.php File Reference

Data Structures

class **DatabaseA** *singleton class to manage a database connection instance.*

Variables

- **\$db** ['default']
Defines the default database connection parameters.

Variable Documentation

\$db ['default']

Initial value:

```
= array(  
    'hostname' => 'localhost',  
    'username' => 'root',  
    'password' => '',  
    'database' => 'foodiego',  
    'dbdriver' => 'mysqli'  
)
```

Defines the default database connection parameters.

This array contains the configuration details required to connect to the database. The parameters include the hostname, username, password, database name, and the driver to use.

Note

Modify the parameters according to your database settings.

Definition at line **21** of file **database-config.php**.

database-config.php

Go to the documentation of this file.

```
00001
00002 <?php
00003 defined('BASEPATH') OR exit('No direct script access allowed'); // Prevent direct
script access
00004
00021 $db['default'] = array(
00022     'hostname' => 'localhost',
00023     'username' => 'root',
00024     'password' => '',
00025     'database' => 'foodiego',
00026     'dbdriver' => 'mysqli'
00027 );
00028
00036 class Database {
00043     private static $instance = null;
00044
00051     private $connection;
00052
00059     private function __construct() {
00060         global $db;
00061
00062         // Establishing the MySQL connection using mysqli
00063         $this->connection = mysqli_connect(
00064             $db['default']['hostname'],
00065             $db['default']['username'],
00066             $db['default']['password'],
00067             $db['default']['database']
00068         );
00069
00070         // Check if the connection is successful, otherwise terminate the script
00071         if (mysqli_connect_errno()) {
00072             die("Database connection failed: " . mysqli_connect_error());
00073         }
00074     }
00075
00084     public static function getInstance() {
00085         if (self::$instance === null) {
00086             self::$instance = new self();
00087         }
00088         return self::$instance;
00089     }
00090
00099     public function getConnection() {
00100         return $this->connection;
00101     }
00102 }
00103 ?>
```

food_menu_view.php File Reference

Variables

- **if**(mysqli_num_rows(\$result) > 0) (**\$row=mysqli_fetch_assoc**(\$result)) (\$row['image_name']=="")
 - **endwhile**
 - **else** __pad0__
-

Variable Documentation

if(mysqli_num_rows(\$result) > 0)
(**\$row=mysqli_fetch_assoc**(\$result))(**\$row**['image_name']=="") (mysqli_num_rows(
\$result) , 0)

Definition at line **7** of file **food_menu_view.php**.

else __pad0__

Definition at line **25** of file **food_menu_view.php**.

endwhile

Definition at line **24** of file **food_menu_view.php**.

food_menu_view.php

Go to the documentation of this file.

```
00001 <?php include('partials-front/menu.php'); ?>
00002
00003 <section class="food-menu">
00004     <div class="container">
00005         <h2 class="text-center">Food Menu</h2>
00006
00007         <?php if (mysqli_num_rows($result) > 0): ?>
00008             <?php while ($row = mysqli_fetch_assoc($result)): ?>
00009                 <div class="food-menu-box">
00010                     <div class="food-menu-img">
00011                         <?php if ($row['image_name'] == ""): ?>
00012                             <div class='error'>Image not available.</div>
00013                         <?php else: ?>
00014                             
00016                         <?php endif; ?>
00017                     </div>
00018                     <div class="food-menu-desc">
00019                         <h4><?php echo htmlspecialchars($row['title']); ?></h4>
00020                         <p class="food-price">Tk. <?php echo
00021 htmlspecialchars($row['price']); ?></p>
00022                         <p class="food-detail"><?php echo
00023 htmlspecialchars($row['description']); ?></p>
00024                         <a href="<?php echo SITEURL; ?>order.php?food_id=<?php
00025 echo $row['id']; ?>" class="btn btn-primary">Order Now</a>
00026                     </div>
00027                 </div>
00028             <?php endwhile; ?>
00029         <?php else: ?>
00030             <div class='error'>Food not found.</div>
00031         <?php endif; ?>
00032     </div>
00033 </section>
00034
00035 <?php include('partials-front/footer.php'); ?>
```

food_search_view.php File Reference

food_search_view.php

Go to the documentation of this file.

```
00001 <?php include('partials-front/menu.php'); ?>
00002
00003 <section class="food-search text-center">
00004     <div class="container">
00005         <h2>Foods on Your Search <a href="#" class="text-white">"<?php echo
htmlspecialchars($search); ?>"</a></h2>
00006     </div>
00007 </section>
00008
00009 <?php include('views/food_menu_view.php'); ?>
```

FoodController.php File Reference

Handles food-related business logic and communication between models and views.

Data Structures

class FoodControllerVariables

- `$controller = new FoodController()`
- `if($_SERVER else ["REQUEST_METHOD"]=="POST" &&isset($_POST['search']))`

Detailed Description

Handles food-related business logic and communication between models and views.

Definition in file **FoodController.php**.

Variable Documentation

\$controller = new FoodController()

Definition at line **34** of file **FoodController.php**.

if (\$_SERVER else["REQUEST_METHOD"]=="POST" &&isset(\$_POST['search']))

Initial value:

```
{  
    $controller->showFoodMenu()  
}
```

Definition at line **37** of file **FoodController.php**.

FoodController.php

Go to the documentation of this file.

```
00001 <?php
00007 require_once 'models/FoodModel.php';
00008
00009 class FoodController {
00010     private $foodModel;
00011
00012     public function construct() {
00013         $this->foodModel = new FoodModel();
00014     }
00015
00019     public function searchFoods($search) {
00020         $result = $this->foodModel->searchFoods($search);
00021         include 'views/food_search_view.php';
00022     }
00023
00027     public function showFoodMenu() {
00028         $result = $this->foodModel->getAllActiveFoods();
00029         include 'views/food_menu_view.php';
00030     }
00031 }
00032
00033 // Route based on request type
00034 $controller = new FoodController();
00035 if ($_SERVER["REQUEST_METHOD"] == "POST" && isset($_POST['search'])) {
00036     $controller->searchFoods($_POST['search']);
00037 } else {
00038     $controller->showFoodMenu();
00039 }
00040 ?>
```

foodModel.php File Reference

Contains functions for food-related database operations.

Data Structures

class **FoodModel**

Detailed Description

Contains functions for food-related database operations.

Definition in file **foodModel.php**.

foodModel.php

Go to the documentation of this file.

```
00001 <?php
00007 require_once 'config/database-config.php';
00008
00009 class FoodModel {
00014     public function getAllActiveFoods() {
00015         global $conn;
00016         $sql = "SELECT * FROM food WHERE active = 'Yes'";
00017         return mysqli_query($conn, $sql);
00018     }
00019
00025     public function searchFoods($search) {
00026         global $conn;
00027         $search = mysqli_real_escape_string($conn, $search);
00028         $sql = "SELECT * FROM food WHERE title LIKE '%$search%' OR description LIKE
00029         '%$search%'";
00029         return mysqli_query($conn, $sql);
00030     }
00031 }
00032 ?>
```

index.php File Reference

Variables

- `const BASEPATH true`
 - `$request = $_SERVER['REQUEST_URI']`
 - `$base_path = '/FoodieGo/'`
-

Variable Documentation

`$base_path = '/FoodieGo/'`

Definition at line **11** of file **index.php**.

`$request = $_SERVER['REQUEST_URI']`

Definition at line **10** of file **index.php**.

`const BASEPATH true`

Definition at line **3** of file **index.php**.

index.php

Go to the documentation of this file.

```
00001 <?php
00002 // Define BASEPATH to allow direct script access
00003 define('BASEPATH', true);
00004
00005 session_start();
00006 require_once 'application/config/config.php';
00007 require once 'system/database/DB config.php';
00008
00009 // Get the requested URL
00010 $request = $_SERVER['REQUEST_URI'];
00011 $base_path = '/FoodieGo/';
00012 $request = str_replace($base_path, '', $request);
00013
00014 // Basic routing
00015 switch ($request) {
00016     case '':
00017     case '/':
00018         require 'application/controllers/food.php';
00019         $controller = new Food();
00020         $controller->menu();
00021         break;
00022
00023     case 'login':
00024         require 'application/controllers/auth.php';
00025         $controller = new Auth();
00026         $controller->login();
00027         break;
00028
00029     case 'logout':
00030         require 'application/controllers/auth.php';
00031         $controller = new Auth();
00032         $controller->logout();
00033         break;
00034
00035     default:
00036         http_response_code(404);
00037         require 'application/views/404.php';
00038         break;
00039 }
```

order-controller.php File Reference

order-controller.php

Go to the documentation of this file.

```
00001 <?php
00007 // Include necessary files
00008 require_once('../models/order-model.php');
00009 require_once('../vendor/autoload.php');
00010
00011 // Check if form is submitted
00012 if ($ SERVER['REQUEST METHOD'] == 'POST') {
00013     // Retrieve and sanitize form data
00014     $full_name = mysqli_real_escape_string($conn, $_POST['full-name']);
00015     $contact = mysqli_real_escape_string($conn, $_POST['contact']);
00016     $email = mysqli_real_escape_string($conn, $_POST['email']);
00017     $address = mysqli_real_escape_string($conn, $_POST['address']);
00018     $food_name = mysqli_real_escape_string($conn, $_POST['food']);
00019     $price = (float)$_POST['price'];
00020     $quantity = (int)$_POST['qty'];
00021
00022     // Process Stripe payment
00023     \Stripe\Stripe::setApiKey('sk_test_your_api_key');
00024     try {
00025         $charge = \Stripe\Charge::create([
00026             'amount' => $price * 100,
00027             'currency' => 'usd',
00028             'source' => $_POST['stripeToken'],
00029             'description' => 'Payment for Food Order',
00030         ]);
00031
00032         if ($charge->status === 'succeeded') {
00033             // Insert order into the database
00034             if (insert_order($full_name, $contact, $email, $address, $food_name,
00035 $price, $quantity)) {
00036                 echo "Payment successful. Your order has been placed!";
00037             } else {
00038                 echo "Payment successful, but there was an issue storing the order.";
00039             }
00040         } catch (\Stripe\Exception\CardException $e) {
00041             echo "Payment failed: " . $e->getError()->message;
00042         }
00043     }
00044     ?>
```

order-model.php File Reference

Functions

- **insert_order** (\$full_name, \$contact, \$email, \$address, \$food_name, \$price, \$quantity)
-

Function Documentation

insert_order (\$full_name, \$contact, \$email, \$address, \$food_name, \$price, \$quantity)

Order Model Handles database operations for orders. Insert a new order into the database.

Parameters

string	<i>\$full_name</i>	Customer's full name.
string	<i>\$contact</i>	Customer's contact number.
string	<i>\$email</i>	Customer's email address.
string	<i>\$address</i>	Customer's delivery address.
string	<i>\$food_name</i>	Ordered food name.
float	<i>\$price</i>	Price of the food item.
int	<i>\$quantity</i>	Quantity ordered.

Returns

bool True if the insertion is successful, false otherwise.

Definition at line **22** of file **order-model.php**.

order-model.php

Go to the documentation of this file.

```
00001 <?php
00007 // Database connection
00008 require_once('config/database-config.php');
00009
00022 function insert_order($full_name, $contact, $email, $address, $food_name, $price,
$quantity) {
00023     global $conn;
00024
00025     $sql = "INSERT INTO orders (customer_name, customer_contact, customer_email,
customer_address, food, price, qty)
00026         VALUES (?, ?, ?, ?, ?, ?, ?)";
00027
00028     $stmt = mysqli_prepare($conn, $sql);
00029     mysqli_stmt_bind_param($stmt, 'sssssd', $full_name, $contact, $email,
$address, $food_name, $price, $quantity);
00030
00031     return mysqli_stmt_execute($stmt);
00032 }
00033 ?>
00034
```

order-view.php File Reference

Variables

- `if ($image_name=="")`
-

Variable Documentation

`if($image_name=="") ($image_name = = "")`

Order View Displays the food order form.

Definition at line **18** of file **order-view.php**.

order-view.php

Go to the documentation of this file.

```
00001 <?php
00007 include('partials-front/menu.php');
00008 ?>
00009
00010 <section class="food-search">
00011     <div class="container">
00012         <h2 class="text-center text-white">Fill this form to confirm your
order.</h2>
00013         <form action="../controllers/order-controller.php" method="POST"
class="order">
00014             <!-- Food Details -->
00015             <fieldset>
00016                 <legend>Selected Food</legend>
00017                 <div class="food-menu-img">
00018                     <?php if ($image_name == ""): ?>
00019                         <div class='error'>Image not available.</div>
00020                     <?php else: ?>
00021                         " class="img-responsive img-curve">
00022                     <?php endif; ?>
00023                 </div>
00024                 <div class="food-menu-desc">
00025                     <h3><?php echo $title; ?></h3>
00026                     <input type="hidden" name="food" value="<?php echo $title; ?>">
00027                     <p class="food-price">Tk.<?php echo $price; ?></p>
00028                     <input type="hidden" name="price" value="<?php echo $price;
?>">
00029                     <div class="order-label">Quantity</div>
00030                     <input type="number" name="qty" class="input-responsive"
value="1" required>
00031                 </div>
00032             </fieldset>
00033
00034             <!-- Customer Details -->
00035             <fieldset>
00036                 <legend>Delivery Details</legend>
00037                 <input type="text" name="full-name" placeholder="E.g. Dhrubo,
Homaira" class="input-responsive" required>
00038                 <input type="tel" name="contact" placeholder="E.g. 9843xxxxxx"
class="input-responsive" required>
00039                 <input type="email" name="email" placeholder="E.g.
email@domain.com" class="input-responsive" required>
00040                 <textarea name="address" rows="10" placeholder="E.g. Street, City,
Country" class="input-responsive" required></textarea>
00041
00042             <!-- Stripe Payment Button -->
00043             <script src="https://checkout.stripe.com/checkout.js"
class="stripe-button"
00044                 data-key="pk_test_your_public_key"
00045                 data-amount="<?php echo $price * 100; ?>"
00046                 data-name="Food Order"
00047                 data-description="Payment for Food Order"
00048                 data-image="https://example.com/logo.png"
00049                 data-locale="auto"
00050                 data-currency="usd">
00051             </script>
00052         </fieldset>
00053     </form>
00054 </div>
00055 </section>
00056
00057 <?php include('partials-front/footer.php'); ?>
```

Index

INDEX