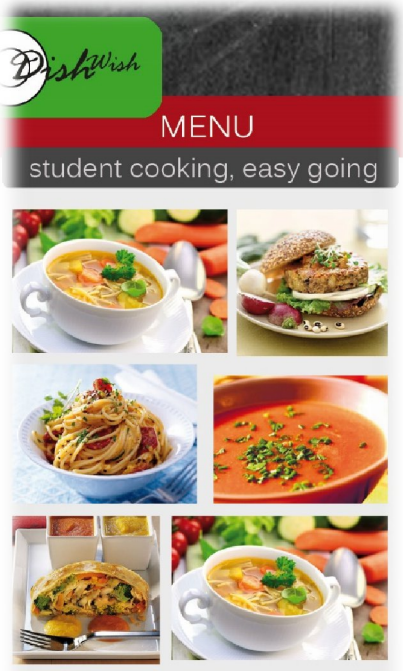
Dokumentation

1.Screendesign

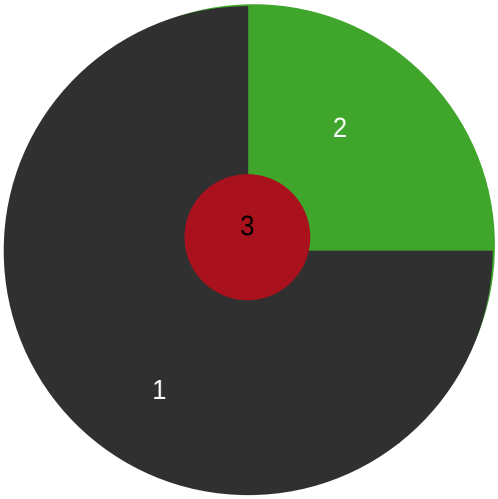
1.1. Fullscreen

1.2.Mobile Ansicht



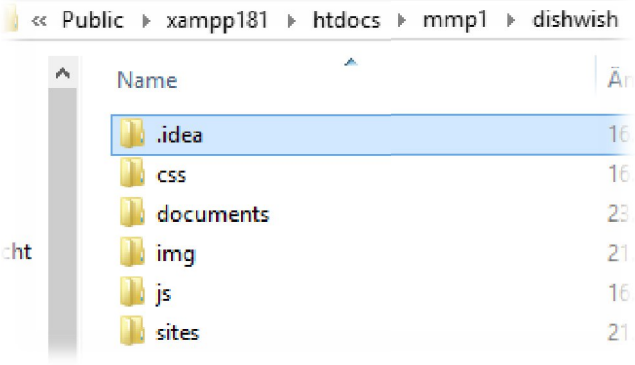
1.3.Farbschema

#30303	#43444	#ecebec	#3fa52b	#a9121	#9a989

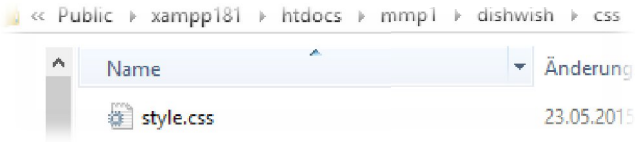


1. Subordinate, or base color
2. Dominant or main color
3. Accent orhighlightcolor

2.Ordnerstruktur

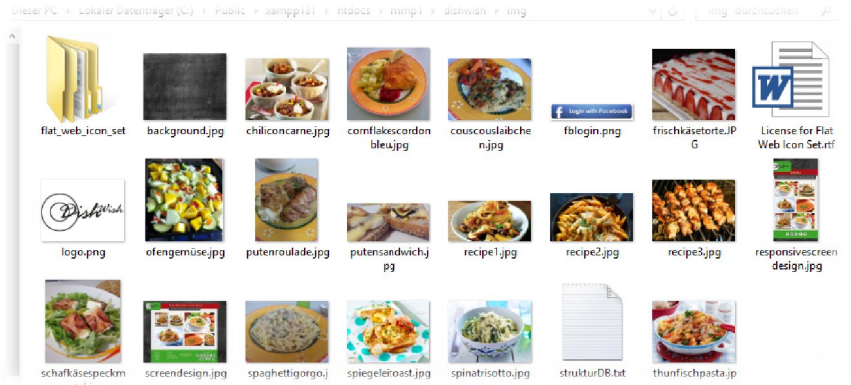


Im Ordner **css** befindet sich die Datei style.css, was die gesamte Design-Umsetzung für die verschiedenen Auflösungen enthält.

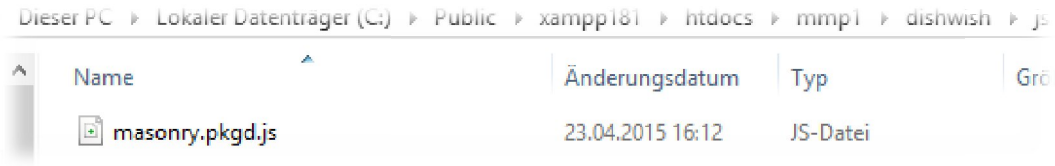


Im Ordner **documents**befinden sich documents.pdf und resüme.pdf**.**

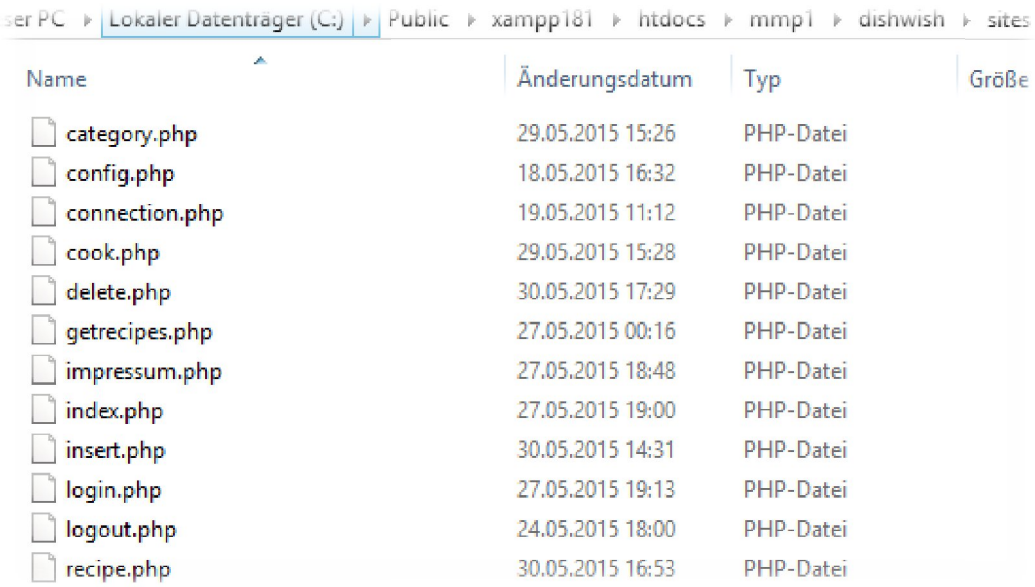
Der Ordner img enthält alle Bilder, die auf der Website dargestellt werden.



Der Ordner **js**enthält die Galerie, die auf der Webseite implementiert ist.



Und im Ordner **sites** befindet sich der komplette Aufbau der Webseite, daher alle Dokumente (php-Dateien) aus der die Webseite besteht.

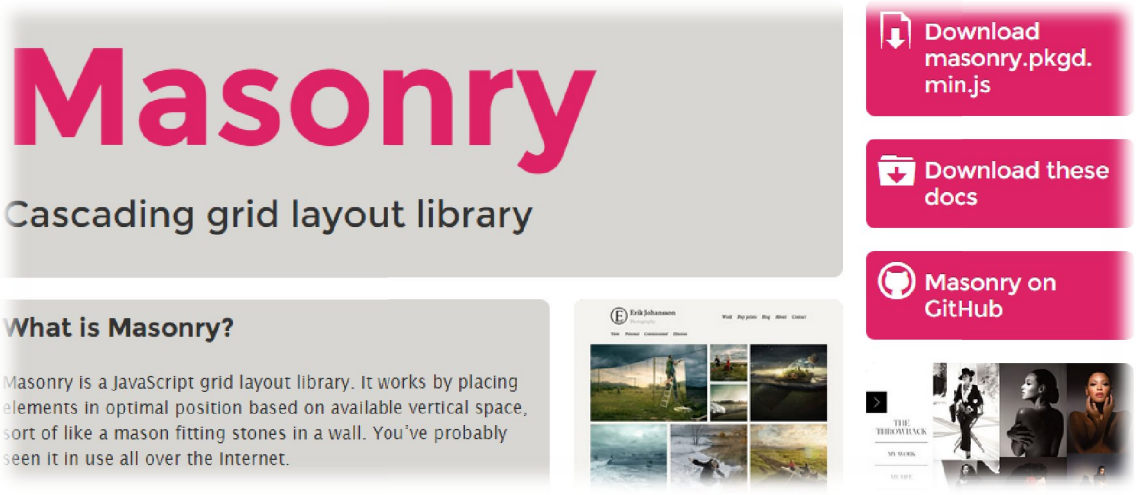


3.Implementierte Plugins

http://masonry.desandro.com/

Masonry isteineJavascript-Layout-Bibliothek. Sie platziert Elemente optimal, basierend auf dem vertikalen Abstand. Bei Veränderung der

Fenstergröße, passt sie sich genau an und verändert wenn nötig, die Platzierung der Elemente.



4.Aufbau der php-Seiten am Beispiel index.php

4.1.Head

```php
<?php
    include "connection.php";

?>
<!DOCTYPE html>
<html>
<head lang="en">
    <meta charset="UTF-8">
    <meta content="width=device-width, initial-scale=1, maximum-scale=1" name="viewport">
    <title>Dishwish</title>

    <!-- add Masonry - GridSystem -->
    <script src="js/masonry.pkgd.js/"></script>

    <!-- add JQuery from Google -->
    <script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>

    <!-- Latest compiled and minified CSS -->
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.4/css/bootstrap.min.css">

    <!-- Optional theme -->
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.4/css/bootstrap-theme.min.css">

    <!-- Latest compiled and minified JavaScript -->
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.4/js/bootstrap.min.js"></script>

    <link rel="stylesheet" href="../css/style.css" type="text/css" media="screen" />

    <script>
        $container.masonry('bindResize');/*masonry-Gallery anpassen an die Fenstergröße*/
     </script>

</head>
```

Der Aufbau jeder Seite ist gleich, vor dem HTML – Tag und dem Doctype ist die Datei „connection.php“ eingebunden, diese stellt die Verbindung zu Datenbank sicher, enthält die Login-Überprüfung und die

Datei „config.php“. In dieser sind der Datenbank-Name, der Host und die Zugangsdaten enthalten.

Im <head>-Bereich befinden sich der verwendete Zeichensatz, das Metatag, das für die Mediaqueries wichtig ist, das eingebundene Stylesheet und die benötigten JavaScript-Files und Funktionen:

- Masonry Gallery (und eine Funktion, für die Anpassung der Galerien an die Fenstergröße)
- jQuery
- Bootstrap

4.2.Body

```
<body>
    <div id="wrap">

        <section class="header"> <!--Header mit Navigationsmenü, das sich bei der Änderung der Fenstergröße anpasst-->

            <nav id="mainnav" class="clearfix toggled-on">
              <div class="container-fluid">
            <!--! Brand and toggle get grouped for better mobile display-->
                <div class="navbar-header">
                    <button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#bs-example-navbar-collapse-1">
                        <span class="sr-only">Toggle navigation</span>
                        <span class="icon-bar"></span>
                        <span class="icon-bar"></span>
                        <span class="icon-bar"></span>
                    </button>
                </div> <!--navbar header-->

            <div id="logo">
                <img src="../img/logo.png" alt="Logo DishWish"> <!--Logo der Seite-->
            </div>

                 <div class="collapse navbar-collapse in" id="bs-example-navbar-collapse-1" aria-expanded="true">
                        <ul class="nav-menu" class="nav navbar-nav">
                            <li><a href="index.php">Home</a></li>
                            <li><a href="category.php">Kategorien</a></li>
                            <li><a href="login.php">Login</a></li>
                        <?php
                            if(isset($_SESSION['USER'])) echo '<li><a href="logout.php">Logout</a></li>';
                            //Nach dem Einloggen erscheint ein Logout-Button im Menü
                        ?>
                            <li><a href="cook.php">Kühlschrank-Kochen</a></li>
                        </ul>
                 </div><!--collapse navbar-->
            </div><!--container fluid-->

            <div id="slogan"><h3>student cooking, easy going</h3></div><!--Slogan der Seite-->

        </nav>

        </section>
```

Der body ist aufgeteilt in einen Navi-, Header-, und Footer-Bereich.

Im **Header** befindet sich das Navigationsmenü, das sich automatisch an die Fenstergröße anpasst. In diesem Bereich gibt es eine php-Funktion, die dem Menü einen Logout-Button hinzufügt, sobald man sich eingeloggt hat. Außerdem sind das Logo und der Slogan im Header implementiert.

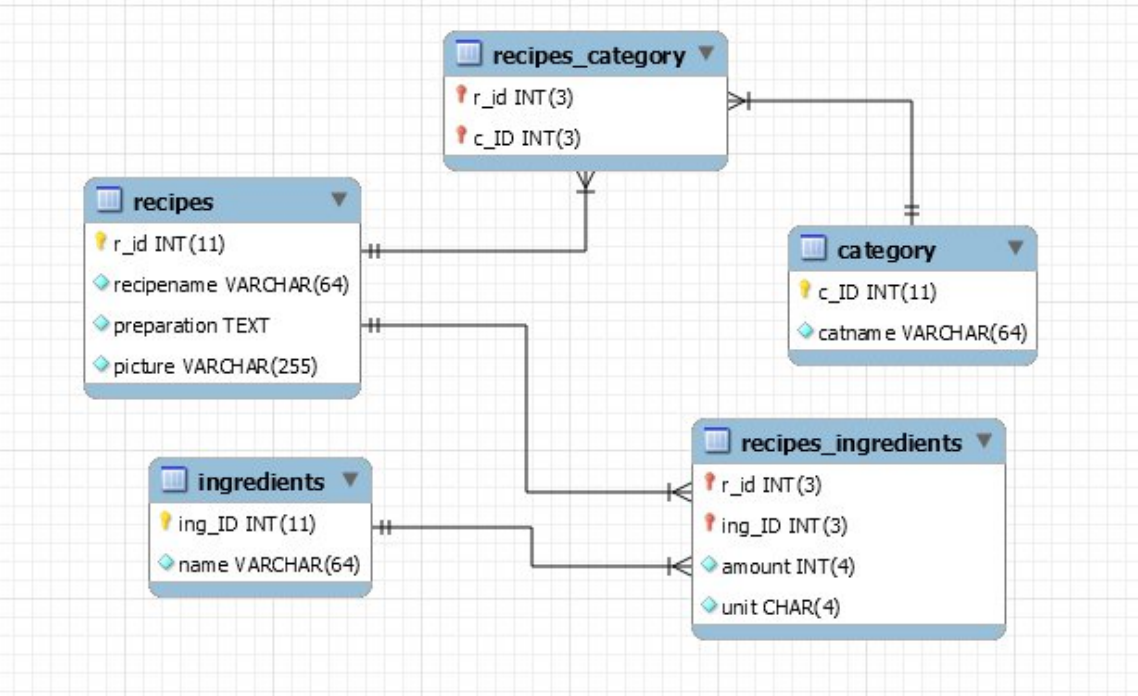
```php
    <section class="content">
        <?php
            if(isset($_SESSION['USER'])) echo '<a href="insert.php"><input type="submit" class="insertRecipe" value="Rezept hinzufügen"></a>';

         ?>
        <div id="container" class="js-masonry" data-masonry-options='{ "columnWidth": 200, "itemSelector": ".item" }'>

            <?php
                $recipe_pictures = array();
                $sth = $dbh->query("SELECT picture AS Bild FROM recipes WHERE picture IS NOT NULL ORDER BY rand() LIMIT 9;");
                //Ausgabe von 9 Random Bildern, die   mit dem jew. Rezept verlinkt sind.

                while ($row = $sth->fetch(PDO::FETCH_ASSOC))
                {
                    array_push($recipe_pictures, $row);
                }
                for($i=0; $i<count($recipe_pictures); $i++)
                {
                        echo '<div class="item"><a href="recipe.php?id="><img src="'.$recipe_pictures[$i]['Bild'].'"></a></div>';
                }

            ?>
        </div>

    </section>
```

Im **Content**-Bereich ist der Inhalt der Webseite. Hier unterscheiden sich die Seiten, da auf jeder Seite eine andere php-Funktion oder SQL – Abfrage implementiert ist.Die Abfrage auf der Startseite gibt 9 Random-Bilder aus der Datenbank aus, die mit dem Rezept verlinkt sind. Diese werden dann in einem div-Container der Masonry-Gallery ausgegeben.

```php
    <section class="footer">
    <div id="row-fluid">
        <div id="footerleft" class="span4">


            <ul><h2>Sitemap</h2>
               <li><a href="index.php">Home</a></li>
                <li><a href="category.php">Kategorien</a></li>
                <li><a href="login.php">Login</a></li>
                <li><a href="suche.php">Suche</a></li>
                <li><a href="cook.php">Kühlschrank-Kochen</a></li>
                <li><a href="impressum.php">Impressum</a></li>
            </ul>
        </div>

        <div id="footercenter" class="span2">

                <h2>Contact</h2>
                <p>Vera Karl<br />
                    Lebzelterweg 10<br />
                    4113 St.Martin/Mkr.
                </p>
                <p class="span2">
                   Mail: verakarl@gmx.at<br />
                   Tel.: 0680/2068951
                </p>

        </div>

        <div id="footerright" class="span4">
            <ul class="share-buttons">
                <li><a href="https://www.facebook.com/sharer/sharer.php?u=http%3A%2F%2Fsheldon.multimediatechnology.at%2F~fhs37231%2F&t=SocialMediaBox" title="
                Share on Facebook" target="_blank"><img src="../img/flat_web_icon_set/black/Facebook.png"></a></li>
                <li><a href="https://twitter.com/intent/tweet?source=http%3A%2F%2Fsheldon.multimediatechnology.at%2F~fhs37231%2F&
                text=SocialMediaBox:%20http%3A%2F%2Fsheldon.multimediatechnology.at%2F~fhs37231%2F&via=verakarl" target="_blank" title="Tweet"><img src="
                ../img/flat_web_icon_set/black/Twitter.png"></a></li>
                <li><a href="http://pinterest.com/pin/create/button/?url=http%3A%2F%2Fsheldon.multimediatechnology.at%2F~fhs37231%2F&description=all%20Access"
                target="_blank" title="Pin it"><img src="../img/flat_web_icon_set/black/Pinterest.png"></a></li>
                <li><a href="mailto:?subject=SocialMediaBox&body=all%20Access:%20http%3A%2F%2Fsheldon.multimediatechnology.at%2F~fhs37231%2F" target="_blank" title=
                "Email"><img src="../img/flat_web_icon_set/black/Email.png"></a></li>
            </ul>
        </div>
  </div>

    </section>
  </div>
</body>
```

Im**Footer**-Bereich befinden sich eine Sitemap, meine Kontaktdaten und die Social-Media-Buttons.

5.Datenbankstruktur



6.AJAX

Bei der Seite **cook.php** gibt es die Möglichkeit, anhand von Zutatenkombinationen Rezepte auszuwählen. Damit die Seite nicht immer neu geladen werden muss, habe ich hier AJAX verwendet, es werden also die ausgewählten Rezepte sofort nachgeladen, ohne die Seite refreshen zu müssen.

```
//function to check if the request is an AJAX request
function is_ajax() {
  return isset($_SERVER['HTTP_X_REQUESTED_WITH']) && strtolower($_SERVER['HTTP_X_REQUESTED_WITH']) == 'xmlhttprequest';
}

if (is_ajax()) {

    if (isset($_POST["recipes"]))
    {
        //recipes array extrahieren (JSON)
        $data = $_POST["recipes"];
        //SQL Abfrage
        if (count($_POST["recipes"]) == 3 )
        {
            $sth = $dbh->prepare("SELECT ri.r_id, r.recipename, r.picture FROM `recipes_ingredients` as ri, `recipes` as r
                    WHERE ri.r_id = r.r_id AND ri.ing_ID = :ingID1 AND ri.r_id
                     IN (
                      SELECT r_id FROM `recipes_ingredients`
                      WHERE  ing_ID = :ingID2 AND r_id
                       IN (
                         SELECT r_id FROM `recipes_ingredients`
                         WHERE  ing_ID = :ingID3
                       ))");
                $sth->bindParam(":ingID1", $data[0]);
                $sth->bindParam(":ingID2", $data[1]);
                $sth->bindParam(":ingID3", $data[2]);
                $sth->execute();

                    $result = $sth->fetchAll(PDO::FETCH_ASSOC);
                    echo json_encode($result);
```