**B.M.S College of Engineering**

**P.O. Box No.: 1908 Bull Temple Road,**

**Bangalore-560 019**

**Department of INFORMATION Science & Engineering**



**Course – WEB APPLICATION DEVELOPMENT**

**Course Code – 19IS3PWWAD**

**AY 2020-21**

**Final report on mini Project work**

**PROJECT FOOD BLOG**

Submitted to – Faculty Name: Nalina V

Submitted by -

Moksh Jayanth GR Mohan D

1BM19IS094 1BM19IS092

**B.M.S College of Engineering**

**P.O. Box No.: 1908 Bull Temple Road,**

**Bangalore-560 019**

**Department of INFORMATION Science & Engineering**



**CERTIFICATE**

Certified that the Project has been successfully presented at **B.M.S College Of Engineering** by **Moksh Jayanth G R** And **Mohan D** bearing USN:**1BM19IS094** And USN:**1BM19IS092** respectively in partial fulfilment of the requirements for the III Semester degree in **Bachelor of Engineering in Information Science & Engineering**of **Visvesvaraya Technological University, Belgaum** as a part the course **WEB APPLICATION DEVELOPMENT *(*19IS3PWWAD)** during academic year 2020-2021.

**Faculty Name – Nalina V**

**Designation – Assistant Professor**

**Department of ISE, BMSCE**

CONTENTS

|  |  |
| --- | --- |
| **Abstract** | 4 |
| **Chapter 1** |  |
| **1.Introduction** | 5 |
| **1.2 Problem Definition** | 6 |
| **1.3 Objective** | 7 |
| **Chapter 2** |  |
| **2.1 Requirements Specification and tools used** | 8 |
| **2.2 System Architecture** | 9 |
| **Chapter 3** |  |
| **3.1 Implementation** | 10 |
| **Chapter 4** |  |
| **4.1 Results** | 17 |
| **Conclusions** | 22 |
| **References** | 23 |

# ABSTRACT

We have built a model website which mimics the basic layout, styling and functionalities of a food blog website. FOODNINJAS, our food blog model was built using HTML, CSS for

Styling , Javascript for client-side scripting and PHP for the server-side scripting.

Additionally, we also made use of Google’s MATERIALIZE CSS frameworks and jQuery Libraries for creating Javascript responses for the site.

Also, the project uses a local hosted database called ‘FoodBlogDb’ to store all the user relatable data and to also to store the food recipes posted by the user.

The main motive of our project is to give the user easy-and-pleasant experience while using the site. In other words, our project FOODNINJAS blog provides very user-friendly interface which makes navigating the website really easy.

# INTRODUCTION

# Following points provides an quick introduction to our project.

Our blog -FoodNinjas:-

1. showcases some popular food items in a section called Food Gallery.
2. provides recipes and instructions to several delicious-and-easy-to-prepare food items.
3. has a video guide to help the viewers have an easy and pleasant experience preparing their favourite food item.
4. provides the user an option to create an account in the blog and become an official FoodNinja.
5. also provides an option of printing all the instructions and recipes of the food item into a hard copy if and only if the user creates an account in FoodNinjas site.
6. also the user has the option of saving the recipes into a soft copy in .pdf format.
7. has an option to share their favourite food recipes with the other blog users.
8. has a eye-catchy design, all the images and styling used for the websites are bright and colourful.
9. basically, has easy-and-pleasant user interface which makes navigating the site very easy i.e. site has a user-friendly interface. The site also has a navigation bar for navigating to different sections of the page.

**PROBLEM DEFINATION:**

This project mainly focuses on providing the user with nice and pleasant experience while using the site. So, we have defined the problems from user’s perspective.

So, the user of this blog is mainly looking forward to find these things:-

1. some delicious food recipes which can be easily prepared with minimum efforts and equipment.
2. a proper and easy-to-follow guidance and instructions to prepare their favourite food.
3. if an user is interested to share his/her own favourite food recipe with other blog users, he/she must give an option to do so.
4. a nice-and-pleasant experience while navigating the website i.e. a user friendly website.

So, keeping all this in mind we have built the site which covers all of these problems. And the user’s expectations are

**OBJECTIVE**

The problems defined in the previous page and the objectives of the project are inter-relatable.

The primary objective of this project is to create a full functioning food blog which should be providing satisfactory user experience.

To go into more detail, the objectives of the project are:

1. to make use a single locally hosted database called ‘FoodBlogDb’ with two independent tables, named ‘users’ and ‘recipes’ respectively.
2. ‘users’ table stores all the login details of the user and ‘recipes’ table stores all the information of the food recipe posted by the user.
3. to provide all the verified users with access to all the video content of the food recipes, also provide them access to all printable documents in the food blog and also provide them an option to save the food recipes into a soft copy(.pdf format).
4. to get the users involved into the activities of the blog, rather than one way interaction, i.e. to provide an option for all the interested users to post their own food recipes into blog which can be viewed by other blog users.

Another subsidiary objective of the project is to make the blog design very eye-catchy.

So, we have made use of Google’s MATERIALIZE CSS frameworks to achieve this eye-catchy design.

**Requirements Specification And Tools Used**

Being done with the user’s point of view, now let’s see things in programmer’s point of view.

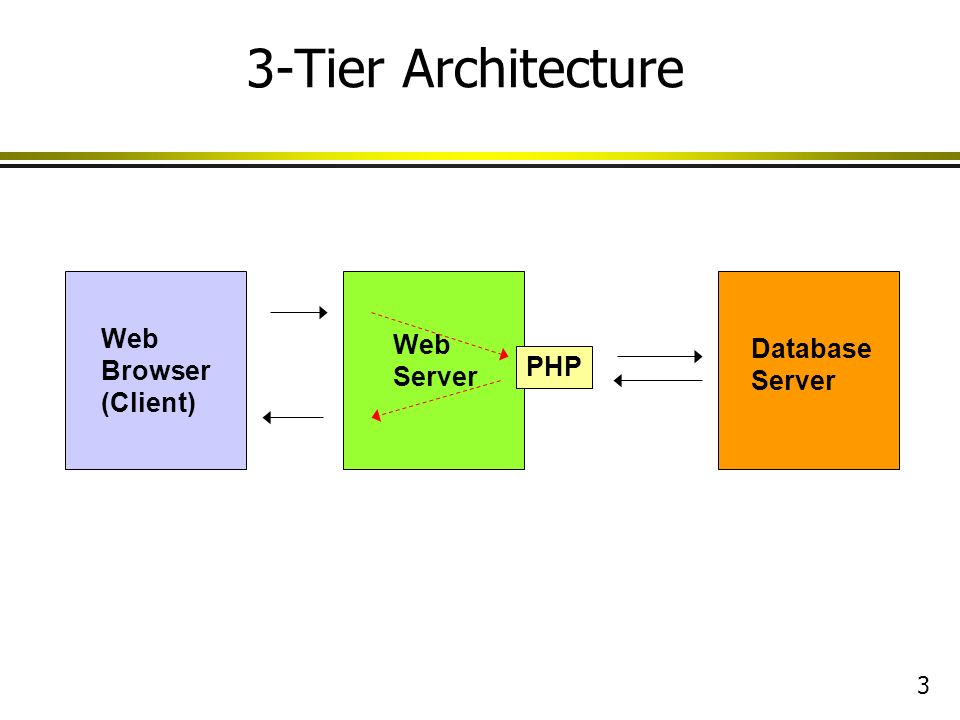
Basic requirements to run this program are as follows :

1. A Computer with a standard web browser (preferably google chrome).
2. A cross-platform web server solution stack package consisting mainly of the HTTP Server, a database, and interpreters for scripts written in the PHP programming language(preferably XAMPP).
3. A locally hosted database called ‘FoodBlogDb’ with two independent tables, named ‘users’ and ‘recipes’ respectively(pictures of the tables are included int report).

Tools used to build the project:

1. **Visual studio code editor**- used for all the coding purposes for this project.
2. **XAMPP** – used for creating ‘FoodBlogDb’ database and to run all the PHP Codes (server-side scripting programming language).
3. A standard web browser to host and display the website.
4. **MATERIALIZE** Minified CSS Package and **Google’s Material Icons** are used for Additional styling of the project.
5. **Minified JQuery library** is used for creating some JavaScript responses in the project.

**SYSTEM ARCHITECTURE**

****

Front-end

The Web browser is designed using HTML & CSS with some of the Materialize CSS tools. Along with several functions of JavaScript.

Back-end

Web server : APACHE & Database server : MySQL.

The PHP is used to interact with both the Web server & Database server.

**IMPLEMENTATION OF THE PROJECT**

Here are the main Code Snippets used in the project and their description:

Snippet 1 :- **Code for Printing out the recipe into a hard copy**

<script>

        function validateOreo(){

            <?php

            require\_once("login\_signup/config.php");

            if(!isset($\_SESSION["login\_sess"])) { ?>

            if(confirm("You have to login inorder to print the recipe."+"\n"+"Do you want to login?")){window.location.href="login\_signup/login.php";}

            <?php } else {?> printOreo(); <?php } ?>

       }

       document.getElementById("b-oreo").addEventListener("click", validateOreo);

        </script>

<script type="text/javascript">

            function printOreo()

            {

               var divsToPrint = document.getElementsByClassName('print-oreo'),n;

                var printContents = "";

                for (n = 0; n < divsToPrint.length; n++)

                {

                printContents += divsToPrint[n].innerHTML+"<br>";

                }

                var newWin= window.open('', 'win');

                newWin.document.write(printContents);

                newWin.location.reload();

                newWin.focus();

                newWin.print();

                newWin.close();

            }

</script>

**Description** – **validateOreo()** function mainly checks if the user is logged in or not. If the user is not logged in then function asks the user to login. After the user logs in, **validateOreo()** calls **printOreo()** , which is used to print the recipe of the food.

This **printOreo()** grabs the content of every <div> tag whose Class=’print-oreo’ (as one can observe the code in the above text). And then prints out the contents using .**print()**

Snippet 2 **:- Sign-up code**

<body>

<a href="../food-blog.php" class=" btn blue white-text" style="position:absolute;right:2%;top:15px;"><span>Home</span><i class="material-icons white-text right" >home</i></a>

  <br>

<div class="container">

  <div class="row">

    <div class="col-sm-4">

    </div>

         <div class="col-sm-4">

    </div>

     <div class="col-sm-4">

    </div>

  </div>

  <div class="row">

<?php

 if(isset($\_POST['signup']))

 {

  extract($\_POST);

  if(strlen($fname)<3){ // Minimum

      $error[] = 'Please enter First Name using 3 charaters atleast.';

        }

if(strlen($fname)>20){  // Max

      $error[] = 'First Name: Max length 20 Characters Not allowed';

        }

if(!preg\_match("/^[A-Za-z \_]\*[A-Za-z ]+[A-Za-z \_]\*$/", $fname)){

            $error[] = 'Invalid Entry First Name. Please Enter letters without any Digit or special symbols like ( 1,2,3#,$,%,&,\*,!,~,`,^,-,)';

        }

if(strlen($lname)<3){ // Minimum

      $error[] = 'Please enter Last Name using 3 charaters atleast.';

        }

if(strlen($lname)>20){  // Max

      $error[] = 'Last Name: Max length 20 Characters Not allowed';

        }

if(!preg\_match("/^[A-Za-z \_]\*[A-Za-z ]+[A-Za-z \_]\*$/", $lname)){

            $error[] = 'Invalid Entry Last Name. Please Enter letters without any Digit or special symbols like ( 1,2,3#,$,%,&,\*,!,~,`,^,-,)';

              }

      if(strlen($username)<3){ // Change Minimum Lenghth

            $error[] = 'Please enter Username using 3 charaters atleast.';

        }

  if(strlen($username)>50){ // Change Max Length

            $error[] = 'Username : Max length 50 Characters Not allowed';

        }

  if(!preg\_match("/^^[^0-9][a-z0-9]+([\_-]?[a-z0-9])\*$/", $username)){

       $error[] = 'Invalid Entry for Username. Enter lowercase letters without any space and No number at the start- Eg - username, unique\_user or myusername123';

        }

if(strlen($email)>50){  // Max

            $error[] = 'Email: Max length 50 Characters Not allowed';

        }

   if($passwordConfirm ==''){

            $error[] = 'Please confirm the password.';

        }

        if($password != $passwordConfirm){

            $error[] = 'Passwords do not match.';

        }

          if(strlen($password)<5){ // min

            $error[] = 'The password should be atleast 6 characters long.';

        }

         if(strlen($password)>20){ // Max

            $error[] = 'Password: Max length 20 Characters Not allowed';

        }

          $sql="select \* from users where (username='$username' or email='$email');";

      $res=mysqli\_query($dbc,$sql);

   if (mysqli\_num\_rows($res) > 0)

   {

     $row = mysqli\_fetch\_assoc($res);

     if($username==$row['username'])

     {

           $error[] ='Username alredy Exists.';

          }

       if($email==$row['email'])

       {

            $error[] ='Email alredy Exists.';

          }

      }

         if(!isset($error))

         {

              $date=date('Y-m-d');

            $options = array("cost"=>4);

            $password = password\_hash($password,PASSWORD\_BCRYPT,$options);

           $result = mysqli\_query($dbc,"INSERT into users values('','$fname','$lname','$username','$email','$password','$date')");

           if($result)

    {

     $done=2;

    }

    else{

      $error[] ='Failed : Something went wrong';

    }

 }

 } ?>

     <div class="col-sm-4">

 <?php

  if(isset($error)){

foreach($error as $error){

  echo '<p class="errmsg">&#x26A0;'.$error.' </p>';

}

}

?>

    </div>

    <div class="col-sm-4">

      <?php if(isset($done))

      { ?>

    <div class="successmsg"><span style="font-size:100px;">&#129305;</span> <br> You have registered successfully . <br> <a href="login.php" style="color:blue">Login here... </a> </div>

      <?php } else { ?>

       <div class="signup\_form">

    <form action="" method="POST">

  <div class="form-group">

        <label class="label\_txt white-text">First Name</label>

    <input type="text" class="form-control" style='color:#EAE9E5' name="fname" value="<?php if(isset($error)){ echo $\_POST['fname'];}?>" required="">

  </div>

  <div class="form-group">

    <label class="label\_txt white-text">Last Name </label>

    <input type="text" class="form-control" style='color:#EAE9E5' name="lname" value="<?php if(isset($error)){ echo $\_POST['lname'];}?>" required="">

  </div>

<div class="form-group">

    <label class="label\_txt white-text">Username </label>

    <input type="text" class="form-control" style='color:#EAE9E5' name="username" value="<?php if(isset($error)){ echo $\_POST['username'];}?>" required="">

  </div>

<div class="form-group">

    <label class="label\_txt white-text">Email </label>

    <input type="email" class="form-control" style='color:#EAE9E5' name="email" value="<?php if(isset($error)){ echo $\_POST['email'];}?>" required="">

  </div>

  <div class="form-group">

    <label class="label\_txt white-text">Password </label>

    <input type="password" name="password" style='color:#EAE9E5' class="form-control" required="">

  </div>

   <div class="form-group">

    <label class="label\_txt white-text">Confirm Password </label>

    <input type="password" name="passwordConfirm" style='color:#EAE9E5' class="form-control" required="">

  </div>

  <button type="submit" name="signup" class="btn btn-primary btn-group-lg form\_btn">SignUp</button>

      <br><br>

   <p class="white-text">Have an account?  <a href="login.php">Log in</a> </p>

</form>

<?php } ?>

</div>

    </div>

    <div class="col-sm-4">

    </div>

  </div>

</div>

</body>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/js/bootstrap.bundle.min.js"></script>

**Description** – The above code is used for signing-up into out website. The code uses regular expressions of php to validate the input of the user, and simple mysqli\_query() to store the data into the DBMS (using “Insert” command). And finally some classes of bootstrap for styling the sign-up page.

Snippet 3 :- **login and login-process code.**

* Login

<body>

<a href="../food-blog.php" class=" btn blue" style="position:absolute;right:2%;top:15px;">home<i class="material-icons right" >home</i></a>

<div class="container">

  <div class="row">

    <div class="col-sm-4">

    </div>

    <div class="col-sm-4">

      <br><br><br>

      <div class="login\_form">

  <form action="login\_process.php" method="POST">

  <div class="form-group"> <br>

<?php

if(isset($\_GET['loginerror'])) {

  $loginerror=$\_GET['loginerror'];

}

 if(!empty($loginerror)){  echo '<p class="errmsg">Invalid login credentials, Please Try Again..</p>'; } ?>

    <label class="label\_txt white-text">Username or Email </label>

    <input type="text" name="login\_var" value="<?php if(!empty($loginerror)){ echo  $loginerror; } ?>" class="form-control" required="">

  </div>

  <div class="form-group">

    <label class="label\_txt white-text">Password </label>

    <input type="password" name="password" class="form-control" required="">

  </div>

  <button type="submit" name="sublogin" class="btn btn-primary btn-group-lg form\_btn">Login</button>

</form>

   <br> <br>

    <p>Don't have an account? <a href="signup.php"><b>Sign up</b></a> </p>

    </div>

    <div class="col-sm-4">

    </div>

    </div>

  </div>

</div>

</body>

* Login-process

<?php

require\_once("config.php");

if(isset($\_POST['sublogin']))

{

$login = $\_POST['login\_var'];

$password = $\_POST['password'];

$query = "select \* from users where ( username='$login' OR email = '$login')";

$res = mysqli\_query($dbc,$query);

$numRows = mysqli\_num\_rows($res);

if($numRows  == 1)

{

        $row = mysqli\_fetch\_assoc($res);

        if(password\_verify($password,$row['password']))

        {

             $\_SESSION["login\_sess"]="1";

             $\_SESSION["login\_email"]= $row['email'];

            header("location:../food-blog.php");

        }

        else{

     header("location:login.php?loginerror=".$login);

        }

    }

    else{

  header("location:login.php?loginerror=".$login);

    }

}

?>

**Description** – The above code is used to login into one’s account. The login code uses bootstrap to style the login code and it is targeted to ‘login-process’ code.

The login-process code uses the mysqli\_query() to get the data from the DBMS (using “select” command) and compares it with the input and proceeds to the main page if the comparison is successful or displays the errors accordingly.

**RESULTS**

Some of the key results achieved are as follows:

1. Firstly, the layout and design and the website is very eye-catching.
2. Using the navigation bar at the top, we are able to navigate to different sections of the site just by clicking buttons in the navigation bar.
3. With help of Materialize CSS and JQuery, the website is Compatible for any dimensions of the screen. The website is completely operational for mobile screens as well as large TV screens.
4. Signup and Login forms are successfully added, and all the user login data is stored in the ‘FoodBlogDb’ database(See the Picture below).
5. Other features like Modals, Tabs and Footer are added and styled using MATERIALIZE CSS.
6. One of the Noteworthy Result is that the addition of print button which on clicked, the food recipes can be printed into a hardcopy.
7. Good Interactivities with user was achieved, since the user of the blog has option to post his/her own food recipes with the other users of the blog. A table named ‘recipes’ stores all the information posted by the user(See the Picture below).

The pictures below shows our completed Project, our mini food blog- FoodNinjas.

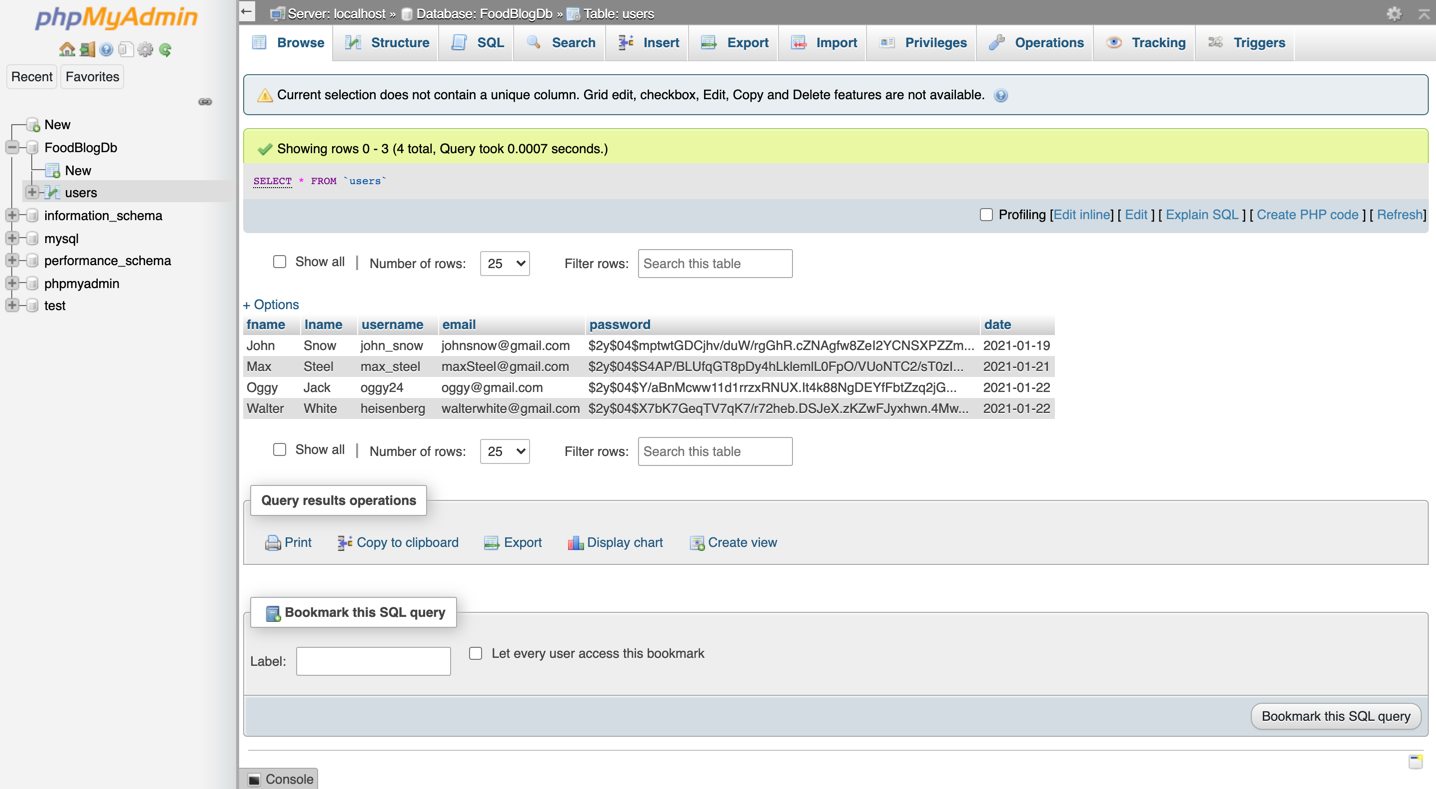
****

Fig 1.1: ‘FoodBlogDb’ Database with table name ‘users’

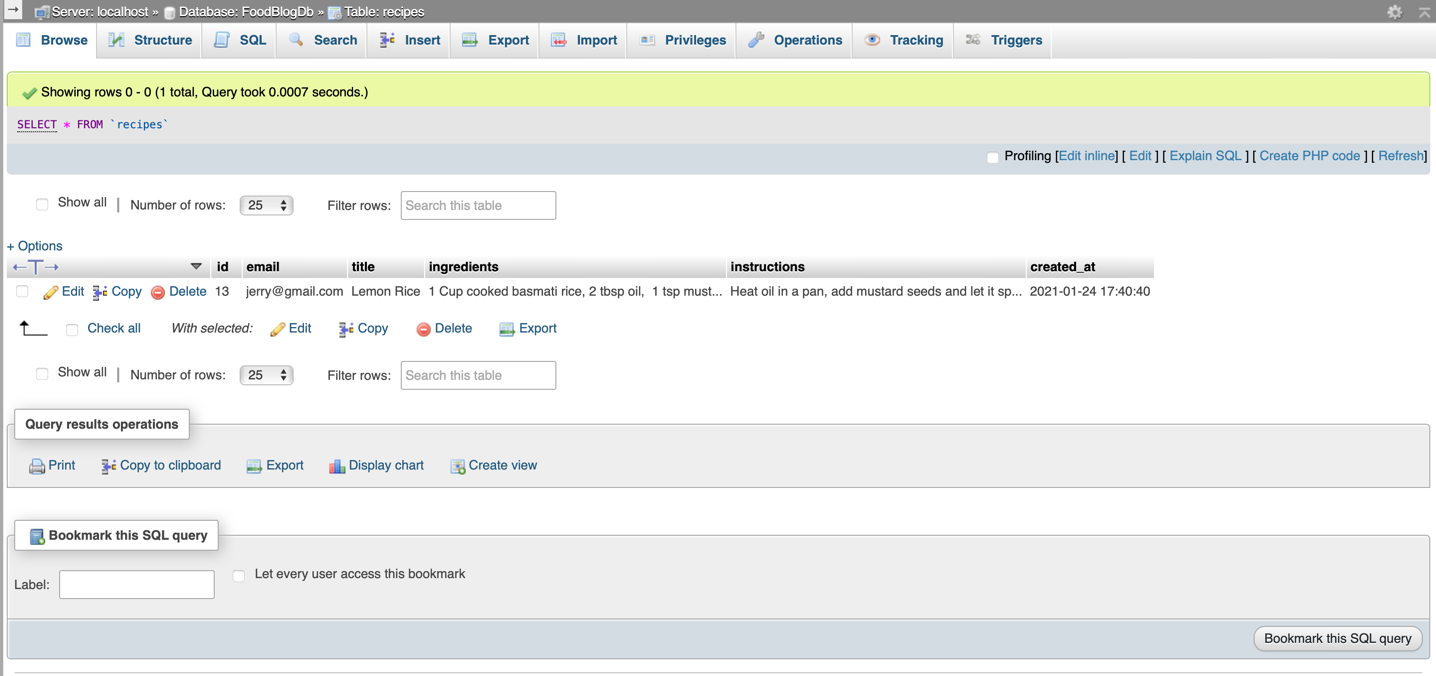
****

Fig 1.2: ‘FoodBlogDb’ Database with table name ‘recipes’

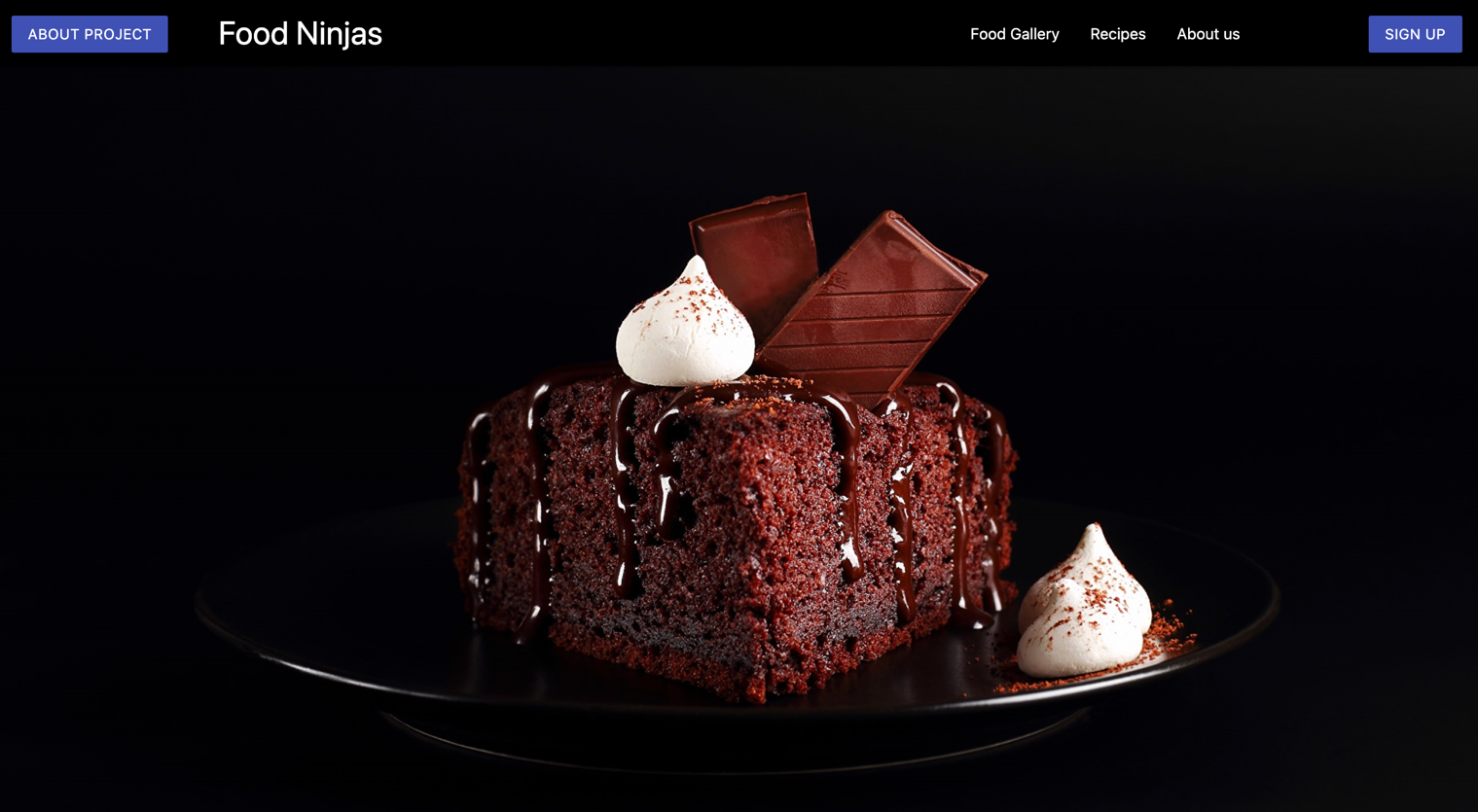
****

Fig 1.3: Home Page – header section

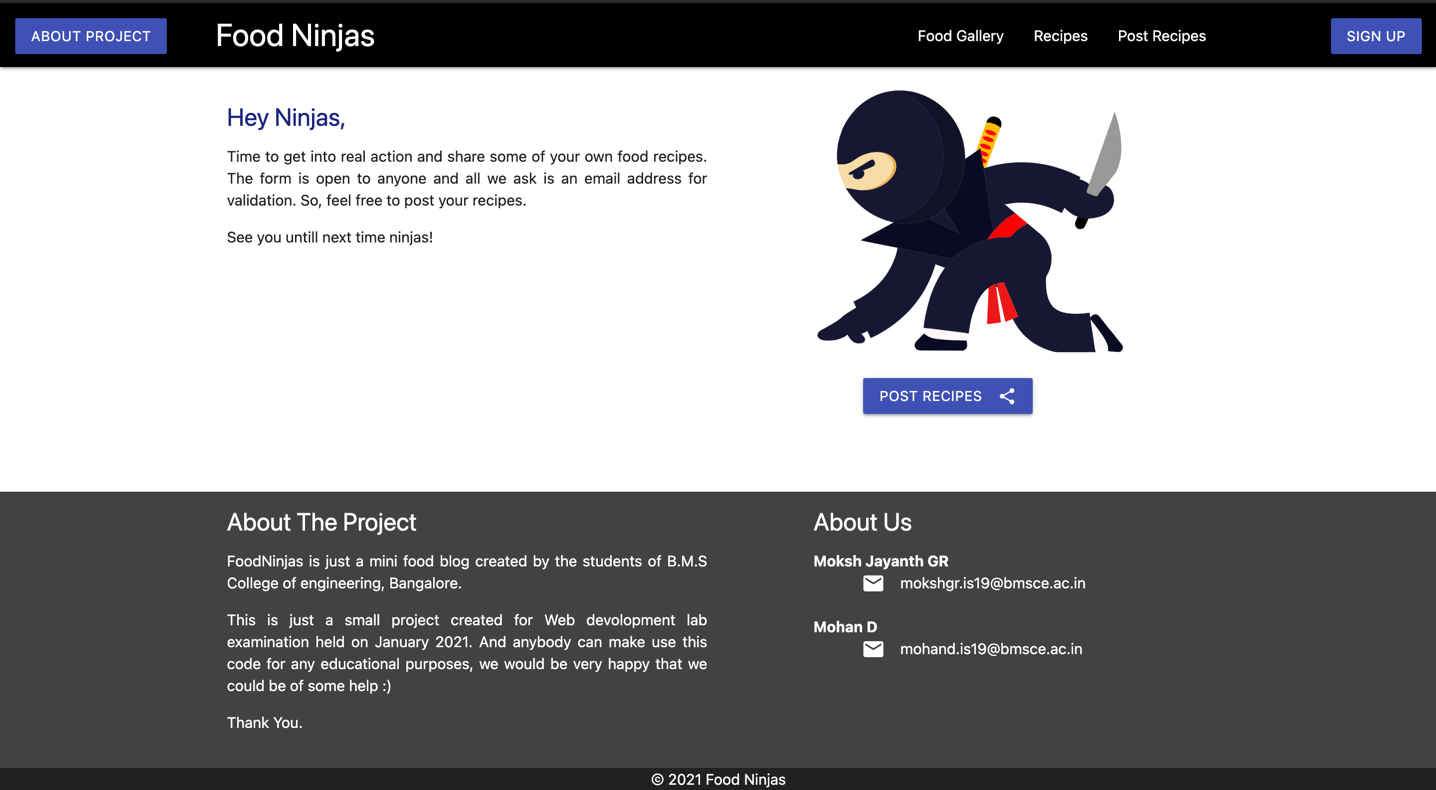
****

Fig 1.4: Home Page – footer and post recipes section

****

Fig 1.5 : Login Page

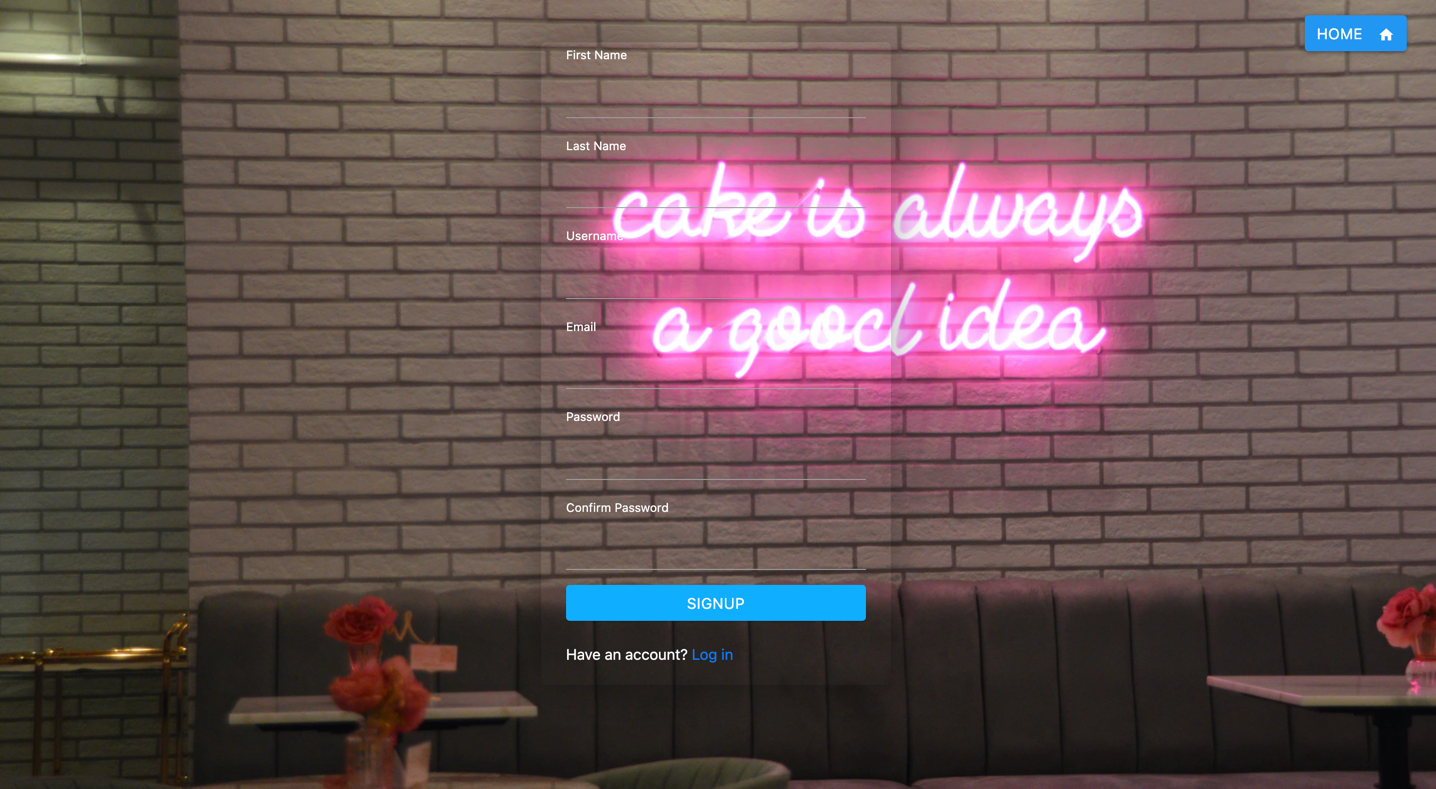
****

Fig 1.6: Signup Page

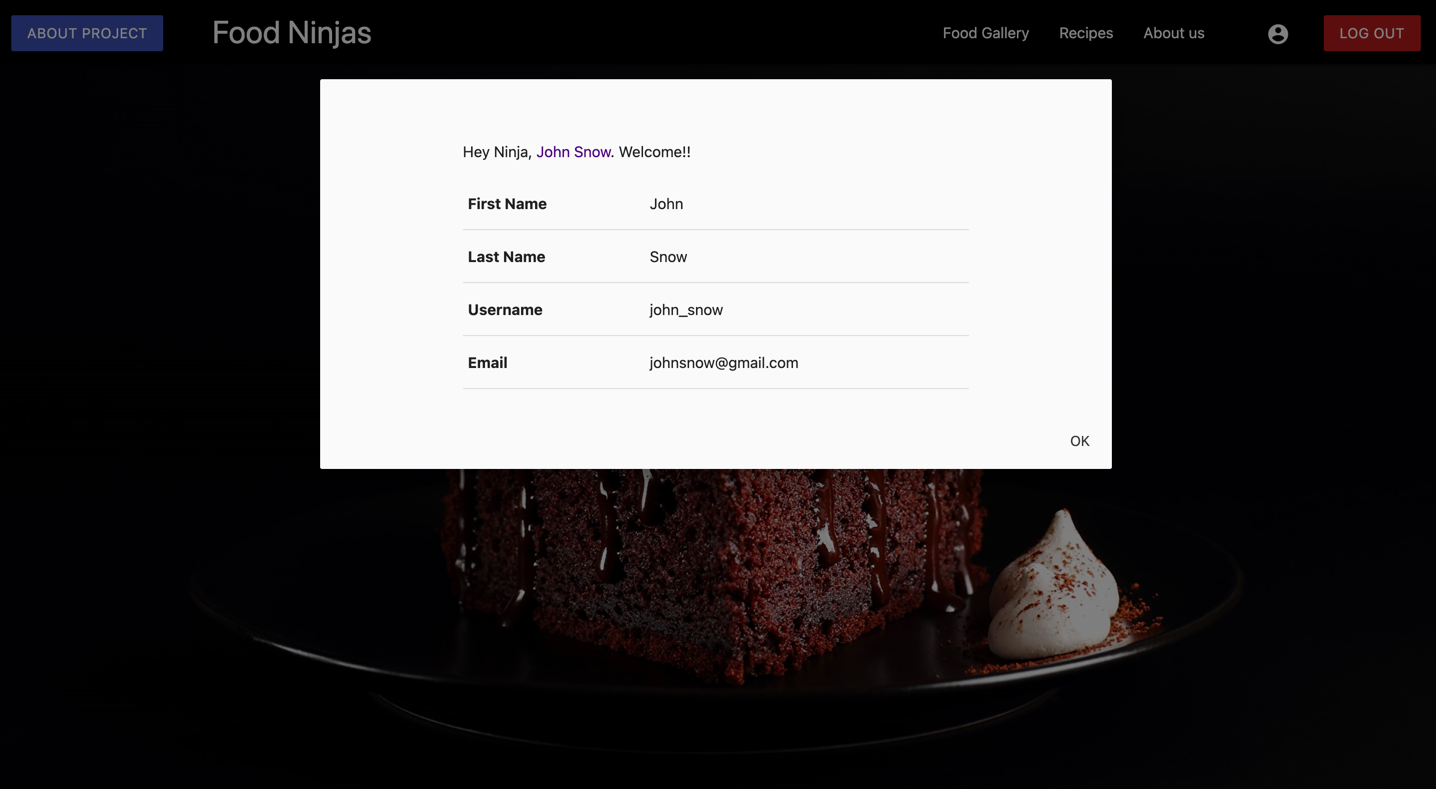
****

Fig 1.7: User’s Profile Modal

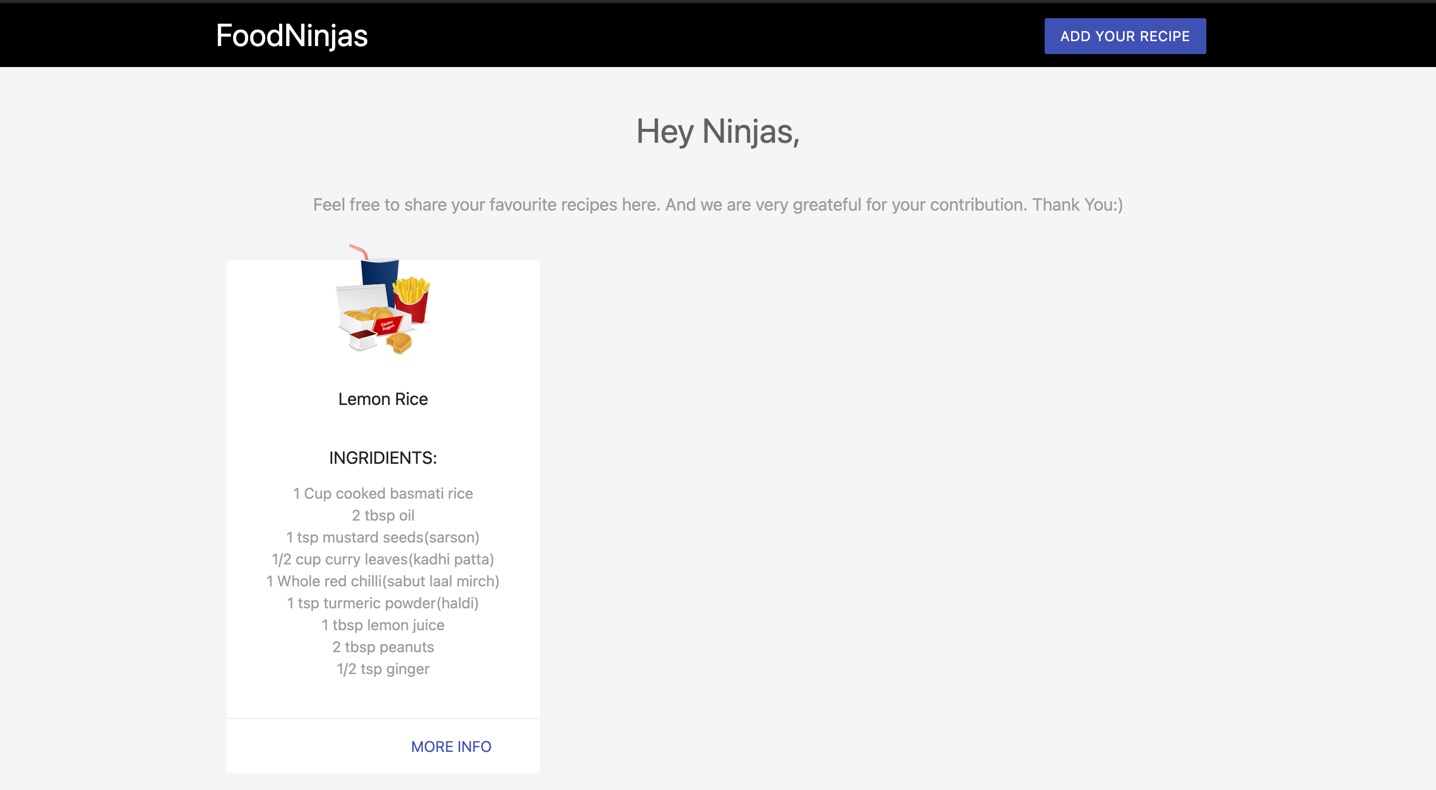
****

Fig 1.8: Post Recipes Page

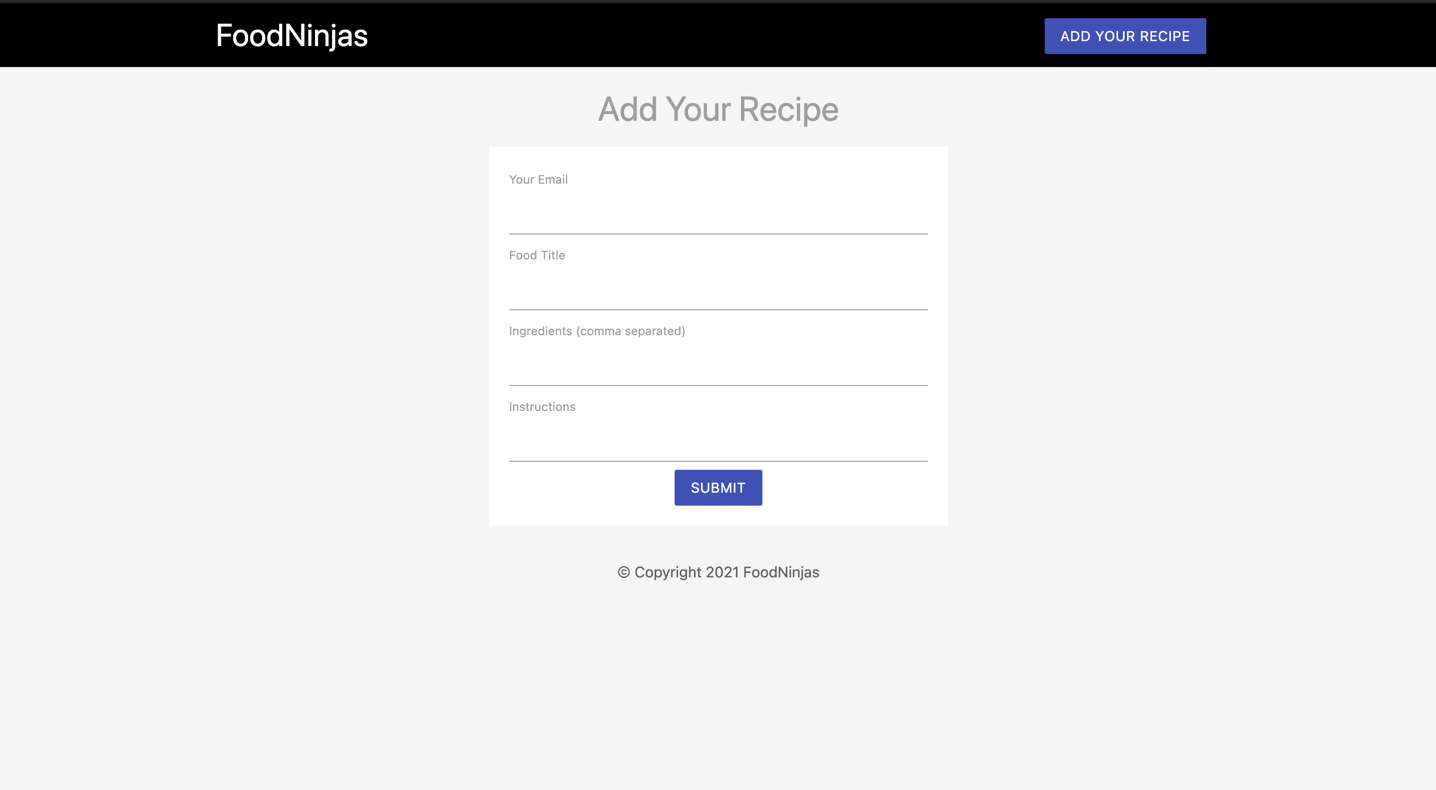
****

Fig 1.9: Submit Recipe Form**CONCLUSIONS**

In conclusion, using HTML, CSS, Javascript and PHP, we were able to build a full functioning mini food blog.

We have made use of most of the concepts learnt in web development classes of 3rd semester in building this mini-project. Along with MATERIALIZE CSS and JQuery.

* **Further Improvements of the Project**

There is always room for improvements, a lots improvements can be done, like the blog can be made more interactive by adding a comment section where in the users can discuss their views on the food recipes, adding the like count for a particular food recipe, adding a recipe to their favorites-list and much more.

* **Functioning of the site.**

We were able to build a full functioning mini food-blog. Cloud databases such a Firestore database, IBM database etc, were more efficient and better options to build this mini-project, but we indeed had to demonstrate the concepts which we learnt in the classes, so we chose to host the database locally.

* **User's experience enhanced**

The website is designed in such a way that it provides the user a very elegant and pleasant experience.

* **Final few words**

We made the website's design and layout as interactive as possible using the concepts that we have learnt in classes and using few of online frameworks. But has said before, there is always a room for improvement.

**REFERENCES**

* The Net Ninja YouTube Channel - <https://www.youtube.com/channel/UCW5YeuERMmlnqo4oq8vwUpg>
* Materialize Documentations - <https://materializecss.com/>
* PHP Documentations - <https://www.php.net/>
* Stack Overflow - <https://stackoverflow.com/>