**CakeWalk - Online Bakery Ordering System**

**1. PURPOSE OF THIS DOCUMENT**

The aim of this document is to describe the online cakeshop website CakeWalk. This document provides information about the technical information and the implementation of the website.

**1.1 INTRODUCTION**

CakeWalk is an e-commerce store website aimed to facilitate the sale and distribution of freshly baked cakes accross India. CakeWalk website provides easy access to choose and buy from a large variety of cakes and makes it easy to get the cakes delivered at a desired location.

**1.2 SCOPE**

The aim of the project is to develop a friendly e-commerce store website. The e-commerce website should target both desktop devices and mobile devices. The e-commerce website should have a front-end and back-end interface. The e-commerce website should target people who likes to buy cakes on both a regular basis or for special occasions.

**1.3 AIMS AND OBJECTIVES**

The main aim and objective of this project is to design and implement a website which consists of these basic features:

- Responsive Front-end interface

- Connection to a database for data storage

- Registering the customers

- Registering the admins

- Display the products on the website

- Enable users to filter products

- Enable the user to choose and buy desired products

**1.4 FUNCTIONAL REQUIREMENTS**

1. The website should consist of a home page.

2. The website should consist of an authentication page.

3. The website should allow the user to sign up to an account.

4. The website should allow the user to sign in to an account.

5. The website should consist of an account details page.

6. The website should allow the user to view his account details.

7. The website should allow the user update his account details.

8. The website should consist of a products page.

9. The website should allow the user to view the products.

10. The website should allow the user to add products to his/her cart.

11. The website should consist of an about page.

12. The website should consist of an admin panel.

13. The website should consist of a navigation bar.

**1.5 NON-FUNCTIONAL REQUIREMENTS**

1. The website should be implemented using HTML, CSS and JavaScript as the front-end language.

2. The website should be implemented using PHP and AJAX as the back-end language.

3. The website should be using MySQL database as the Real-time database.

4. The website should be responsive to most screen sizes.

5. The website should allow only allow one user to sign at a time.

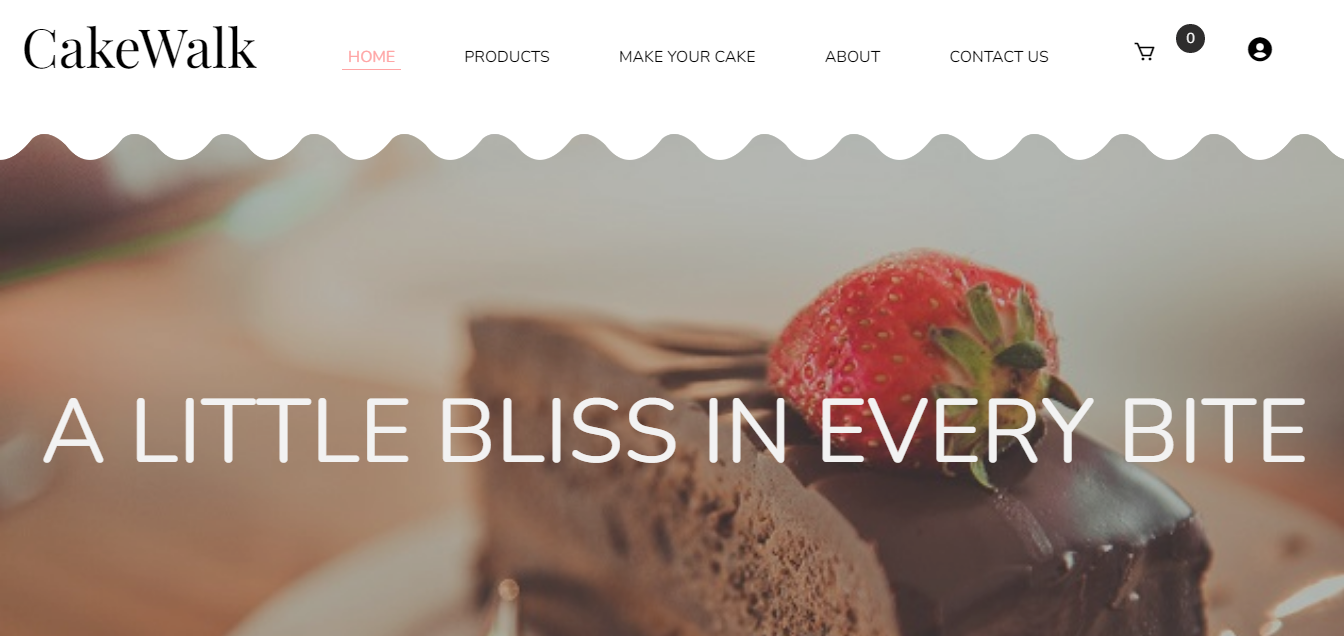
6. The website should save a session

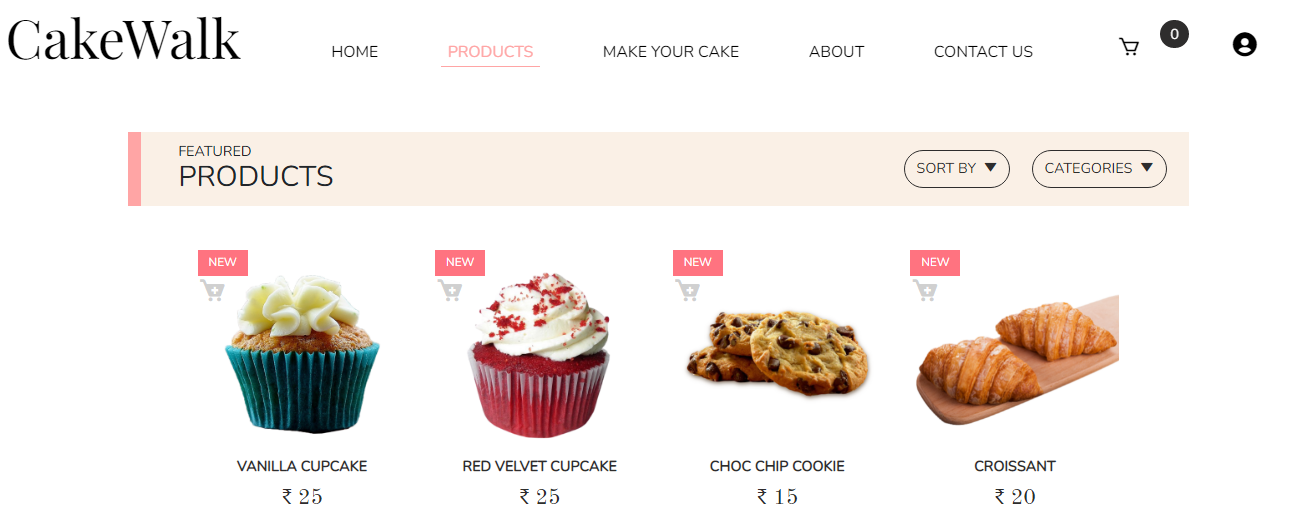
7. The website should be showing the navigation bar on every webpage.

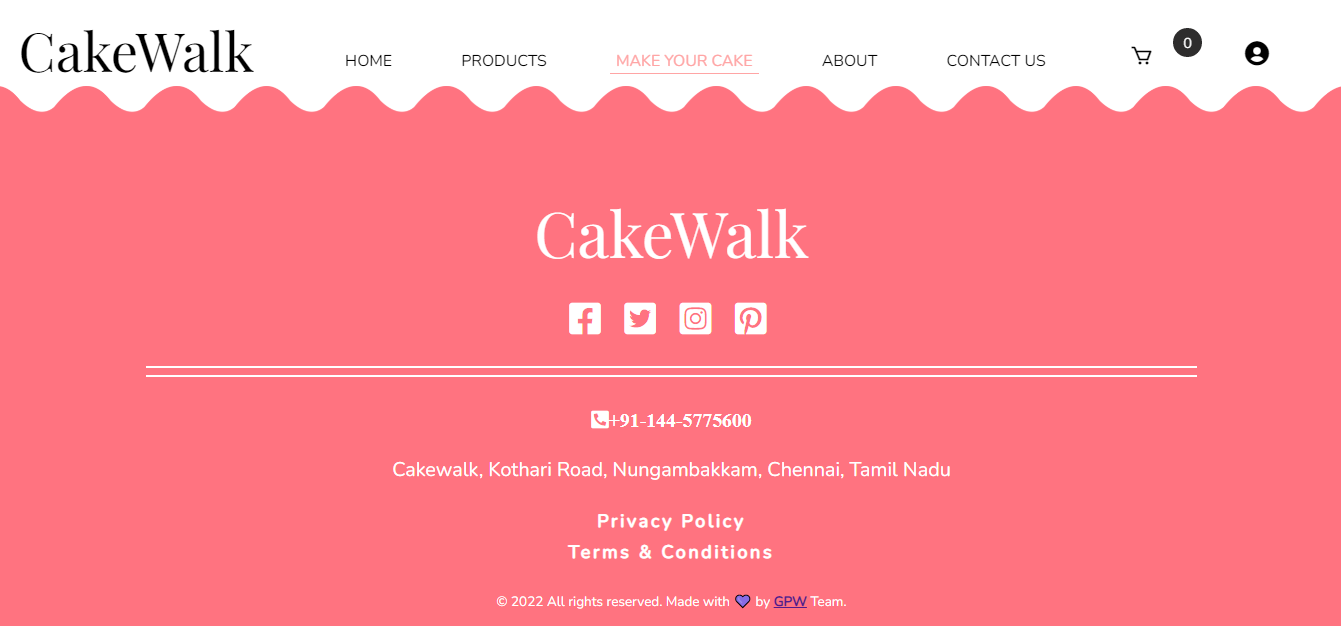
**2. IMPLEMENTATION**

This section will provide the details about each page and screenshots.

**2.2 SCREENSHOTS**

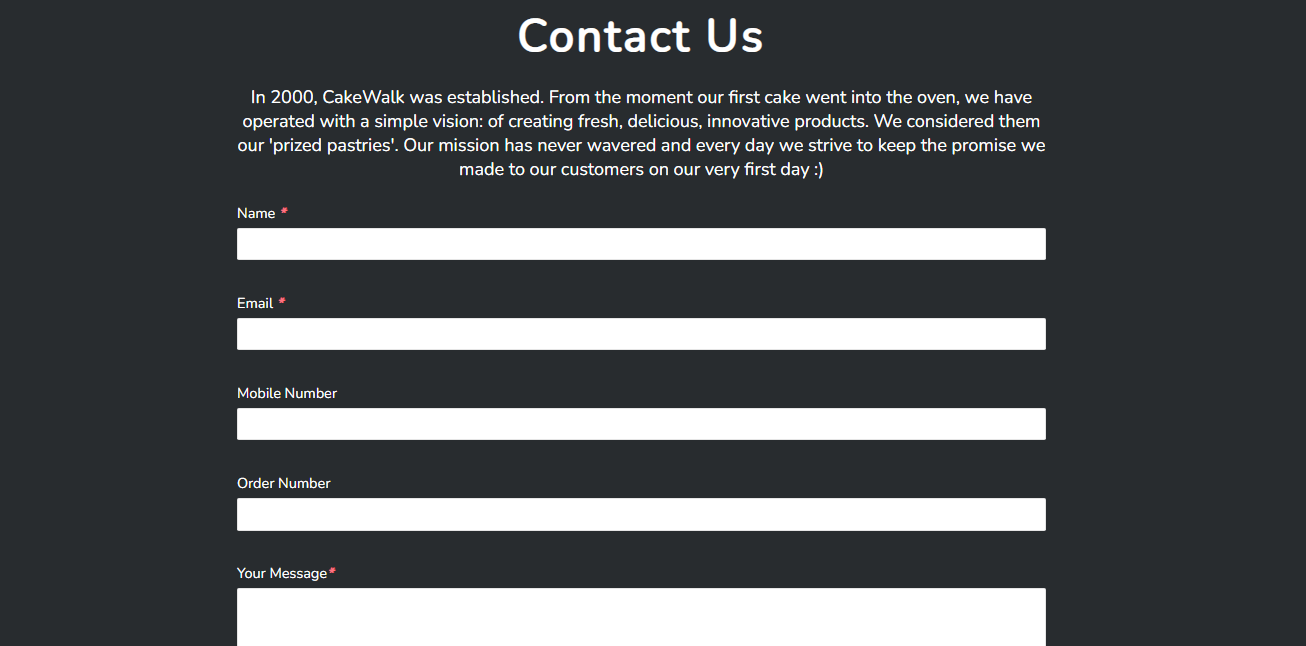


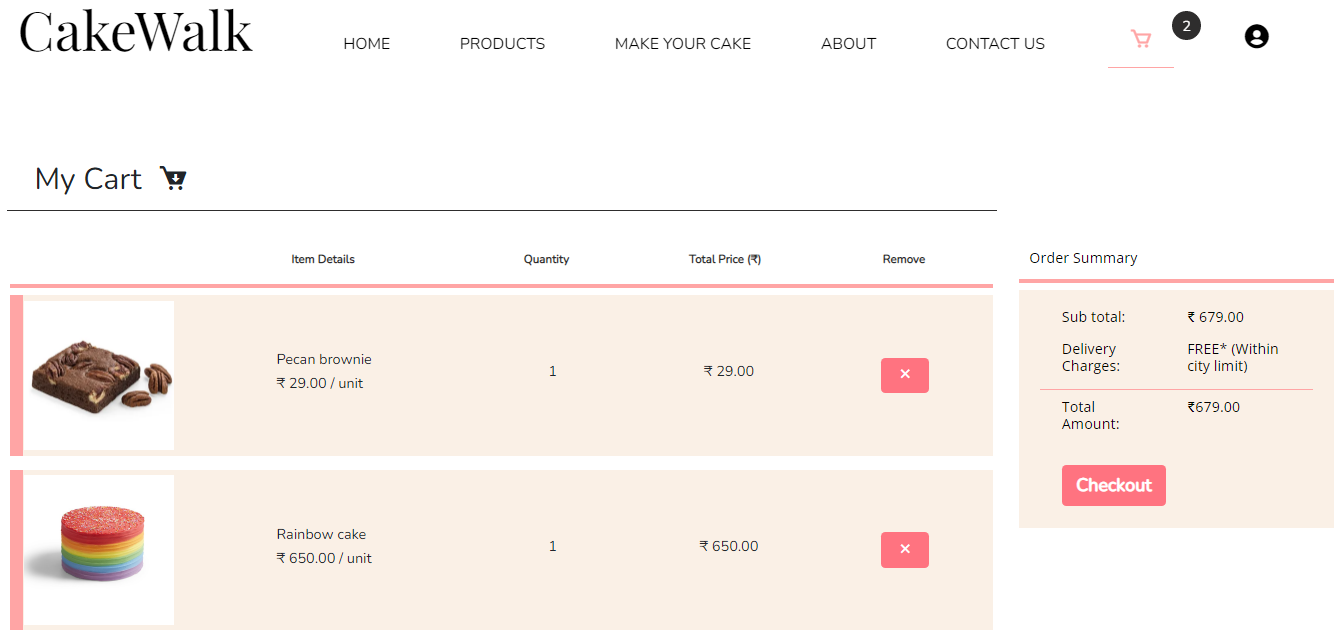


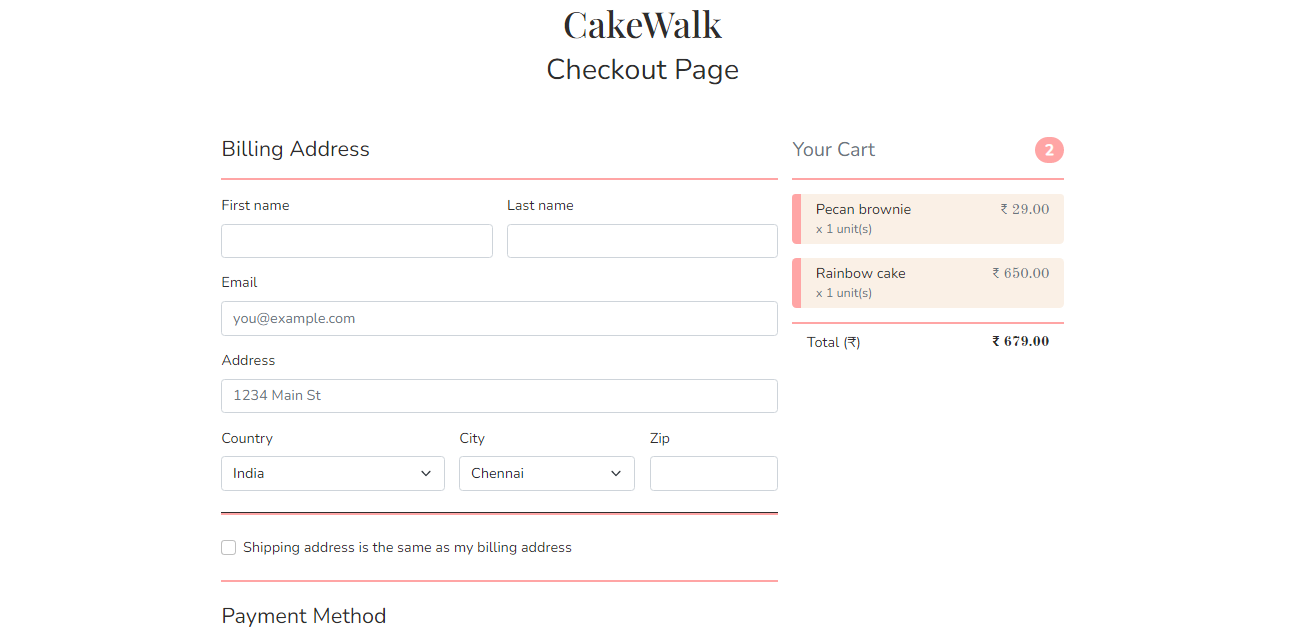


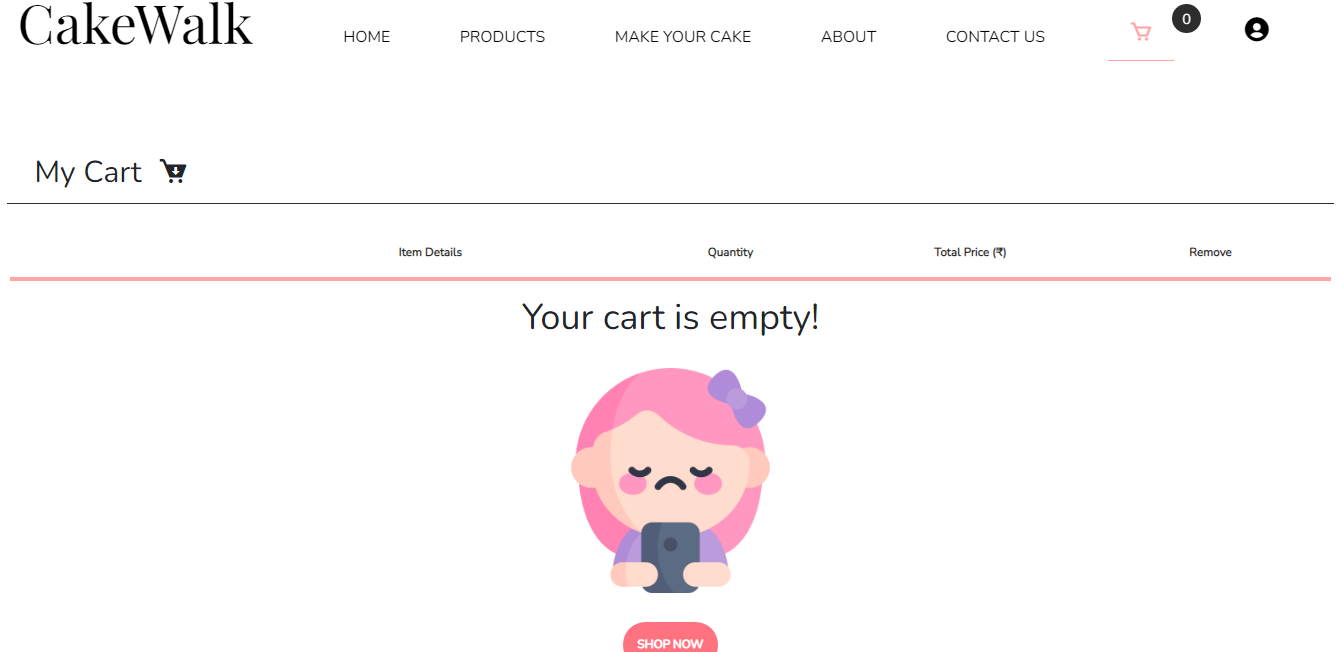


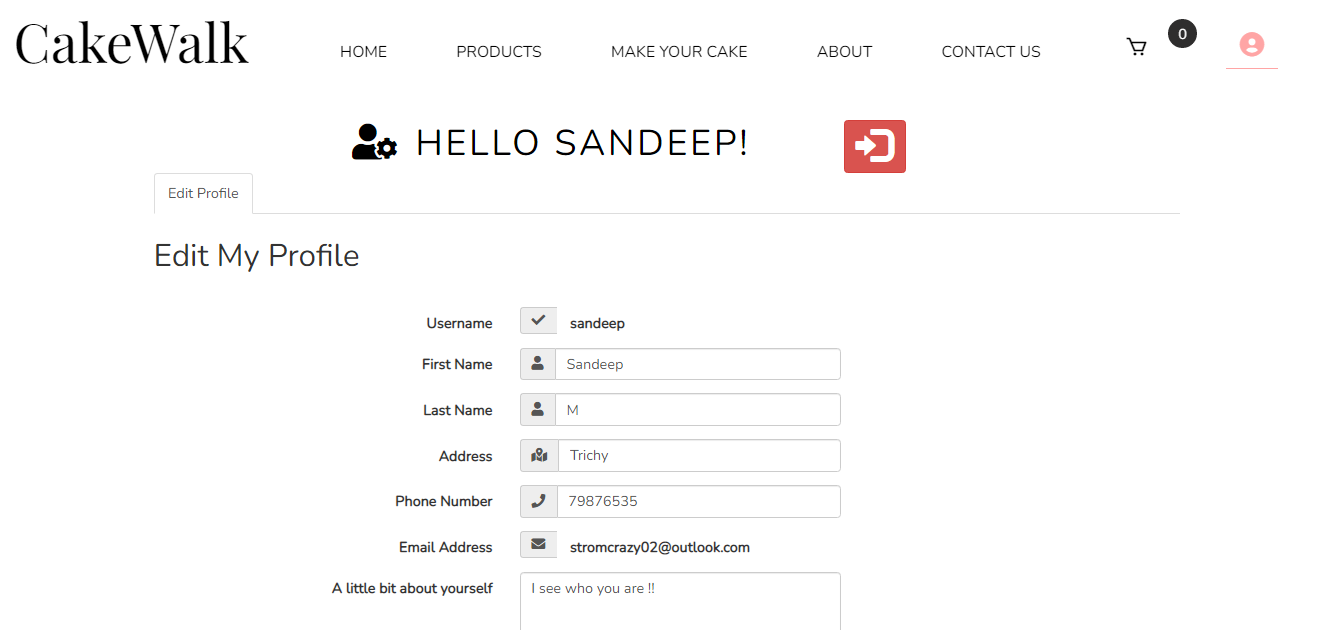












**2.3 PC BROWSER SIZE**

The following screenshots shows the website viewed in a web browser on a screen size larger than 1200 px.

**3. DATABASE**

The database has been designed and normalised to keep a record of all the information about a particular user and a particular product. The database contains 11 tables, each of which stores details about a particular aspect of the online shop.

**Cart:**

The cart table connects the unique cartID which is generated to a particular user.

**cartItem:**

The cartItem table stores all the information about a particular product which the user has added to the shopping cart.

**categories, product\_category:**

The categories table and product\_category table links the different categories which a product can have.

**product\_type, types:**

The product\_type and types table links the different types which a product can have.

**orderitem, userOrder:**

The orderItem and userOrder table stores information about a particular order and a particular item which has been ordered once the user press checkout.

**Transaction:**

The transaction table is used to record a particular transaction which has taken place after the user has accepted an order.

**User:**

The user table stores all the information about a particular user including admin users.

**4. INDIVIDUAL REPORT**

We have implemented the Home page, About page, Contact Us page, Login Page, Registration Page, Forget Password Page, User Profile Page, Admin Panel Page and Send Mail To Subscribers Page.

* **Home Page:**

The home page has been implemented with HTML, CSS, Javascript, PHP and AJAX. It is divided in several parts whereby the user can have a global idea of what the product website is selling with details about the services offered as well as instructions on how to order on the website. The user also has an overview of the people involved in the company. These parts have been implemented with HTML and CSS. The contact form and newsletter have been implemented with HTML, CSS, JavaScript, PHP and AJAX. The contact form enables the user to send a mail to CakeWalk bakery mailbox(your email id). The field validation is done using JavaScript and once the fields have been validated, the data is send to a PHP file to be processed using AJAX such that there is no need to reload the page in order to get a confirmation of the completion of the mail sending process. In order to be able to send mail, the sendMail.ini and php.ini file in XAMPP folder have been modified such that mails can be sent from localhost. The code snippet below execute the mail sending process using the ‘mail’ function in PHP:

The newsletter also uses JavaScript to validate the correct email format and then the data is sent to a PHP file to be processed using AJAX. The function has been designed in such a way that only registered users can subscribe to a newsletter. If an invalid email or a non registered email is entered, the user will be prompted with an appropriate message. The google map has been integrated using a widget available from developers.google.com. Only the width and height have been modified to adapt it to our website.

* **Include Files:**

The sections listed below are all common across all pages. They have been implemented separately using the \*\*\_‘include’\_\*\* function in PHP. This prevents the same code from being implemented in each page making the website more difficult to maintain.

* Contact Us Form
* Newsletter Form
* Google Map
* Navigation Bar
* Mobile Navigation Bar
* Bottom Navigation Bar
* Footer
* **About Page:**

This page has been implemented with HTML and CSS. It contains more details about the people involved in the bakery CakeWalk. Information such as bakers, awards and more information about the CEO of the bakery are available. The contact form and newsletter forms are also available here.

* **Contact Page:**

This contact page contains all the necessary information needed to get in contact with people at CakeWalk such as google map, contact form and newsletter.

* **Login Page:**

The login page has been implemented using HTML, CSS and PHP. The form validation and processing are all done on the server side using PHP. The server checks if the username and password combination corresponds and only after confirmation gives the user access to his account. The function \*\*\_‘checkAccount.php’\_\*\* checks if the user is an admin and if he is, he will be redirected to the admin panel instead of the regular user profile account.

* **Forgot Password Page:**

The forgot password page has been implemented using HTML, CSS and PHP. All the form validations and processes are done on the server side using PHP. If the user has forgotten his password or username, using his email address, he can recover both. Once the button ‘RESET PASSWORD’ has been pressed, the user will receive an email containing his username and a randomized new password.

* **Registration Page:**

The registration page has been implemented using HTML, CSS and PHP. All the form validations and processes are done on the server side using PHP. The username and email are checked if they already exist in the database. If it is the case, an error message is prompted to the user. A google captcha checkbox is also used to prevent any type of bot attacks on the website. Once all the input have been validated, the user will receive an email with a link to validate the entered email during registration. The user will be able to login only after validating his email otherwise an error message will be prompted if the user tries to login before validating the email.

* **User Profile Page:**

This page has been implemented with HTML, CSS, Bootstrap and PHP. The user can view and edit his personal information and his password from this page. He can also delete his account from this page. The input validation are performed on the server side and if the correct input have been entered, the user can update his profile or change his password. The user is prompted with an appropriate message if the data has been successfully updated. The user can also logout using the logout button.

* **Admin Panel Page:**

This page has been implemented using HTML, CSS, Bootstrap, JavaScript and PHP. Here also the user can view, edit and update his information including his password and also delete his account. In addition to these options, an admin can also write and send an email to the subscribers directly from the website using a rich text editor implemented with JavaScript. The editor has been implemented using a widget named ‘summernote’ and it has been modified using the code snippet below to add additional options to the editor. The mail is sent using the code snippet below in PHP and automatically search for all subscribers in the database and add them in Bcc to the mail.

* **Responsiveness:**

This website has been designed in two parts using media queries in the CSS files. All pages have been designed twice. Once for mobile screen size and once for large screen size.

* **Page Access:**

In order to prevent anyone from accessing some crucial pages directly from the URL, a function \*\*\_‘checkAccess.php’\_\*\* has been designed to give access only to pages which have a constant named 'Access' defined in the page. Only pages which have been given access at the appropriate time will be displayed otherwise the page will display access denied. For example, trying to access \*\*\_‘userAccount.php’\_\*\* from the URL without signing in will display the following message: Access Denied!