Implementation Manual

OUR MAIL

An Email system for a group

Nandita Baishya

OUR MAIL: An Email System for a group

Implementation Manual

COT – 402

WEB ENGINEERING

DEPARTMENT OF COMPUTER ENGINEERING

NIT KURUKSHETRA

SUBMITTED TO:

Dr. MINAKSHI SHARMA

NIT KURUSHETRA

SUBMITTED BY:

NANDITA BAISHYA

11610150

Co – 1

TABLE OF CONTENTS

Serial Number	Section Name	Page Number
1.	Introduction to System	3
2.	Technology Stack	3
3.	Modularization	5
4.	Implementation Chart	7
5.	Database Setup	8
6.	Application Snippets	9
7.	Implementation Summary	12

Introduction to System

“OURMAIL” is an implementation of minuscule version of an Email System. The registered users in a group can communicate with each other using their respective E-mail Id mentioned during the registration. After the registration process, the user will be part of the group and listed along with the other users who registered earlier. After the registration, An Email area will be automatically allocated to the user where the user can communicate with others by exchanging mails. The received mails will be present in the inbox which initially depicts all the message headers and if the user is intended to check a particular mail, the corresponding message content along with the sender address and timestamp of message arrival will be displayed.

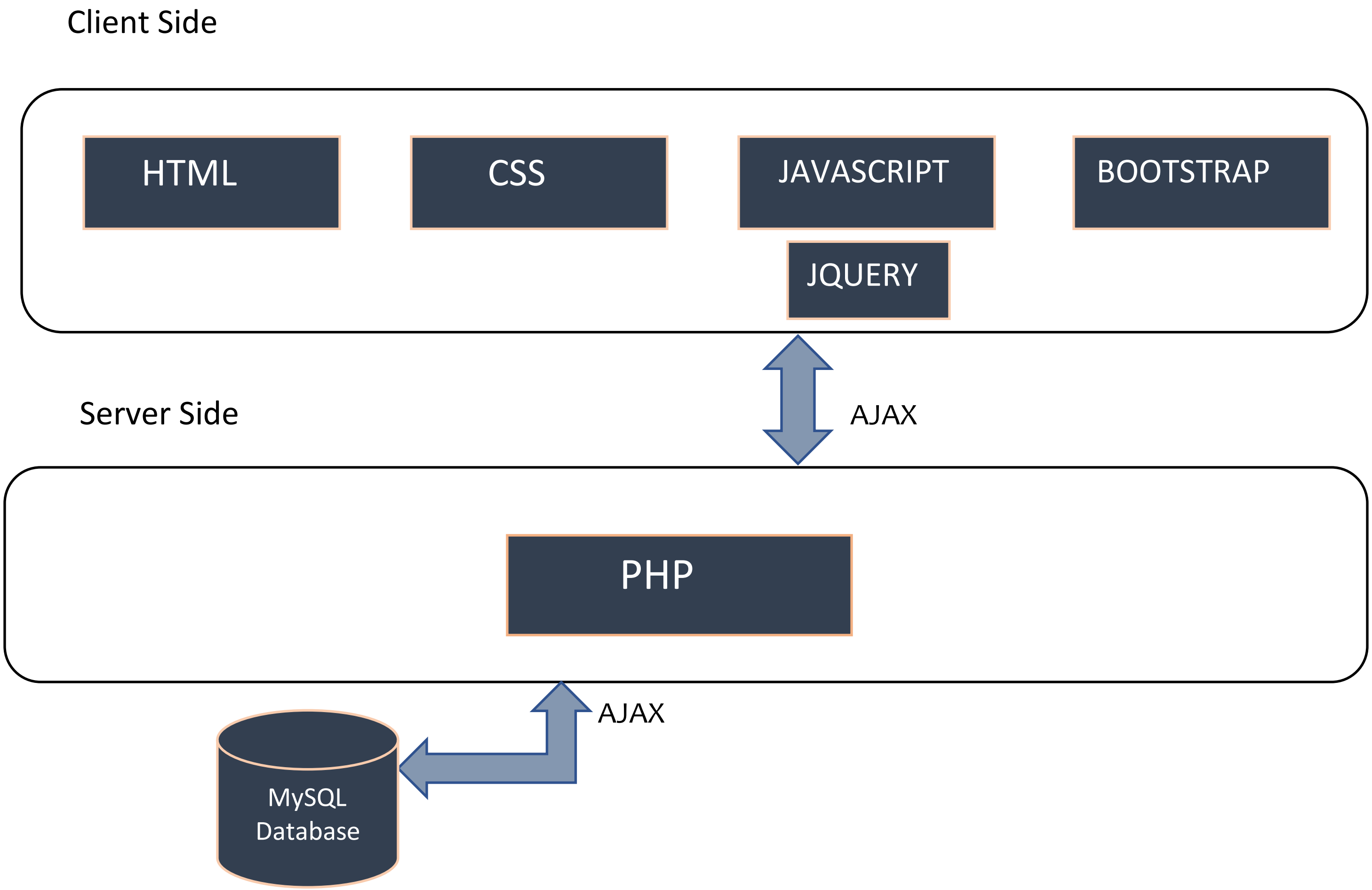
Technology Stack

There are various front end and back end technologies are used during the implementation of application. The user interface is written in HTML (Hypertext Markup Language), which is the standard markup language for documents designed to be displayed in a web browser and these web pages are styled by CSS (cascading style sheets). In order to enhance the interactive nature and user experience Javascript is used, which is a programming language that conforms to the ECMAScript specification. JavaScript is high-level, often just-in-time compiled, and multi-paradigm. It has curly-bracket syntax, dynamic typing, prototype-based object-orientation, and first-class functions. All the web pages are made more responsive using the bootstrap, which is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains CSS- and JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components. In order to simplify the HTML DOM tree traversal and manipulation, as well as event handling, CSS animation and Ajax, a Javascript library called jQuery is used.

The backend part of the application, i.e., the server-side scripting is done by PHP which handles all the actions performed in the client side and manipulates the data in the MySQL database as response. The communication between client and server takes place through Ajax calls. Ajax is a set of web development techniques using many web technologies on the client side to create asynchronous web applications. With Ajax, web applications can send and retrieve data from a server asynchronously without interfering with the display and behavior of the existing page. The communication between server-side script and database is done through interface called PDO (the PHP Data Objects), which defines a lightweight interface for accessing databases in PHP. It provides a data-access abstraction layer for working with databases in PHP. It defines consistent API for working with various database systems. During the implementation process, the entire system functionalities are exercised using the XAMPP server.

***Technologies used**: HTML ,CSS, Javascript, JQuery, Ajax, Bootstrap, PHP, MySQL*

Technology Stack Block Diagram



Modularization

1.Home page

This module is the initial web page of the application. It contains the title of the application along with the one line description. It also provides a link to view the LinkedIn profile of the developer. It provides user with two options, the first one is to login to the application if the user has registered earlier. The second one is to sign up for the application for the new users.

2. Login form

The login form contains two fields, username and password. In order to enter the application, the user who is already part of the group must provide both. The form also provides an option to remember the current user, which upon checking, drops a cookie for the user in the application that expires 30 days. Once the user gets logged in, the corresponding email area will be displayed.

3. Registration form

The registration form for a new user, who is intended to be part of the existing group, contains four fields such as Full name, Email Id, User name and password. The Email Id acts as the medium of communication in this entire application. The user name of a user must be unique in the entire group. The account password is required in order to secure the email area of the respective user. None of these fields must be left empty in order to register into the group.

4. Addition of user data

This module invokes the database connection module at the beginning. Upon the submission of the valid registration form, this module adds the registered user to the list of users group in the email system present in the database. After this entire process, it also invokes another module for the creation of respective email area for the new user. At the termination point of the module, it displays the status of registration in the form of a message to the user.

5. Database Connection

This module attempts to connect to the database, using the PHP Data Objects (PDO) interface. After this attempt, the connection status is displayed to user in the form of a message.

6. Creation of Email Area

Upon the addition of user details to the list of users in the system, this module establishes its own connection to the database and creates an email area for new user in the form of a table in the database, which basically holds the details of message identifier, the address of message sender, the timestamp of message arrival, the content of the message and the header of the message.

7. Navigation bar

This module holds the module to send new message and module which displays the received messages in a particular email area intact. It also let the user to exit the corresponding email area.

8. New Message

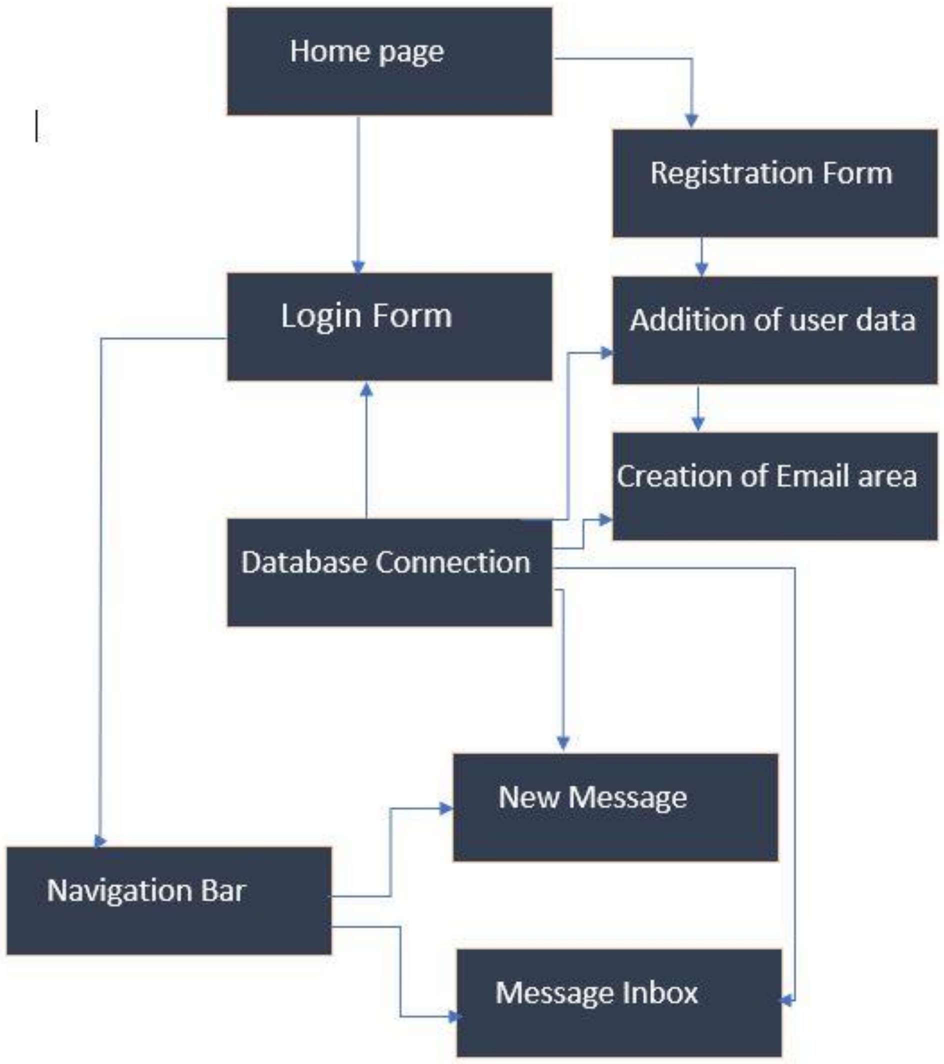
This module let a user to send an email to another user in the same group. The user basically needs to enter the recipient email address, the header of the message being sent and the body of the message. Upon the submission of this message, the application authenticates the message and send an acknowledgement message to the user about the successful delivery. This module invokes the database connection module and it writes the message which was sent to the recipient email area in the database. Later when the recipient logs into the system, this message will be displayed in the inbox of the email area.

9. Message Inbox

This module initially displays the headers of all the messages received from various users in the group. If the user is intended to view the details of a particular message received, then the module displays the content of the message and the associated details in a particular frame of the email area.

Implementation Chart

The following depiction is a block diagram that assists in understanding the implementation of the entire Email system based on the mentioned modules:



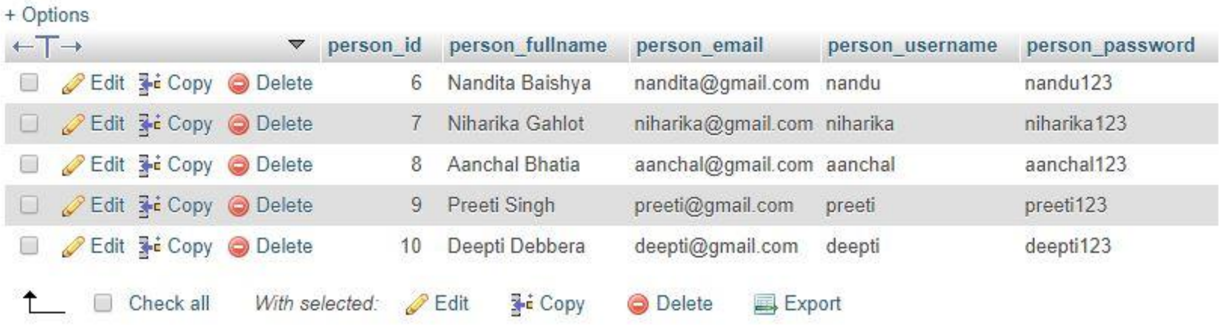
Database Setup

Database Name : “ourmail”

1. Users Detail in a group

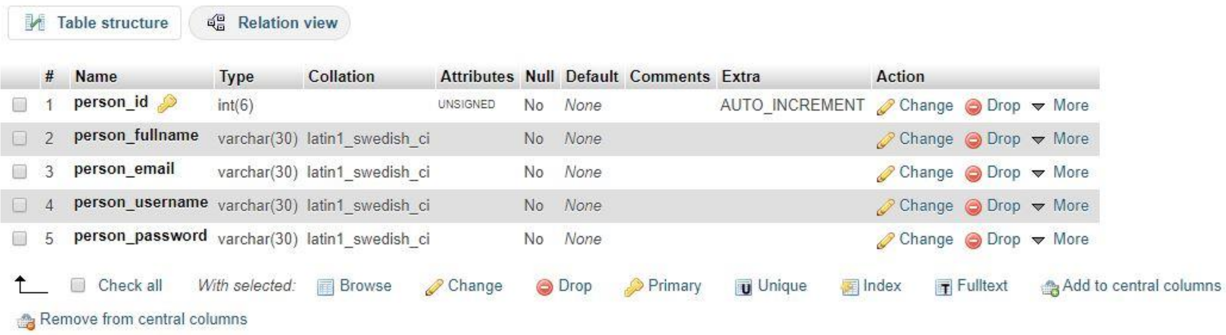
Table Name: “users_july”

a) Instance



person_id	person_fullname	person_email	person_username	person_password
6	Nandita Baishya	nandita@gmail.com	nandu	nandu123
7	Niharika Gahlot	niharika@gmail.com	niharika	niharika123
8	Aanchal Bhatia	aanchal@gmail.com	aanchal	aanchal123
9	Preeti Singh	preeti@gmail.com	preeti	preeti123
10	Deepti Debbera	deepti@gmail.com	deepti	deepti123

b) Structure

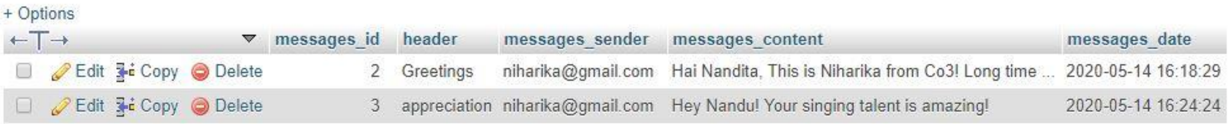


#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	person_id	int(6)		UNSIGNED	No	None		AUTO_INCREMENT	Change Drop More
2	person_fullname	varchar(30)	latin1_swedish_ci		No	None			Change Drop More
3	person_email	varchar(30)	latin1_swedish_ci		No	None			Change Drop More
4	person_username	varchar(30)	latin1_swedish_ci		No	None			Change Drop More
5	person_password	varchar(30)	latin1_swedish_ci		No	None			Change Drop More

2. An Email Area

Table Name: {user name}

a) Instance



+ Options

←T→		messages_id	header	messages_sender	messages_content	messages_date
□	Edit Copy Delete	2	Greetings	niharika@gmail.com	Hai Nandita, This is Niharika from Co3! Long time ...	2020-05-14 16:18:29
□	Edit Copy Delete	3	appreciation	niharika@gmail.com	Hey Nandu! Your singing talent is amazing!	2020-05-14 16:24:24

b) Structure

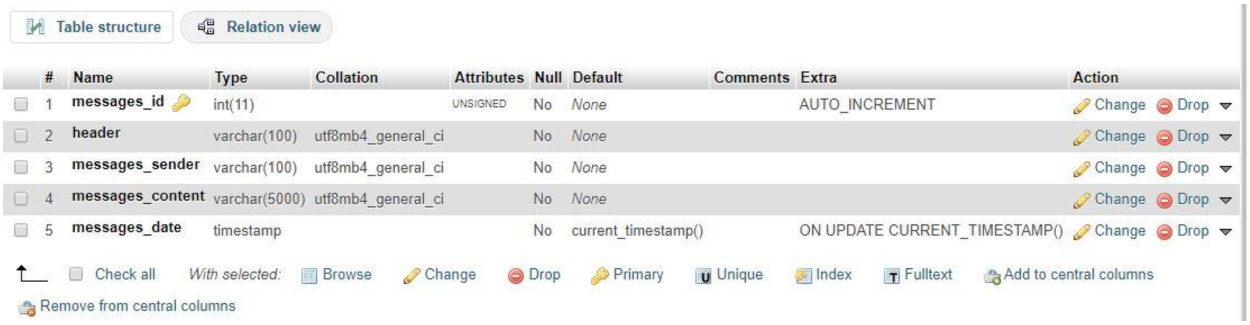


Table structure | Relation view

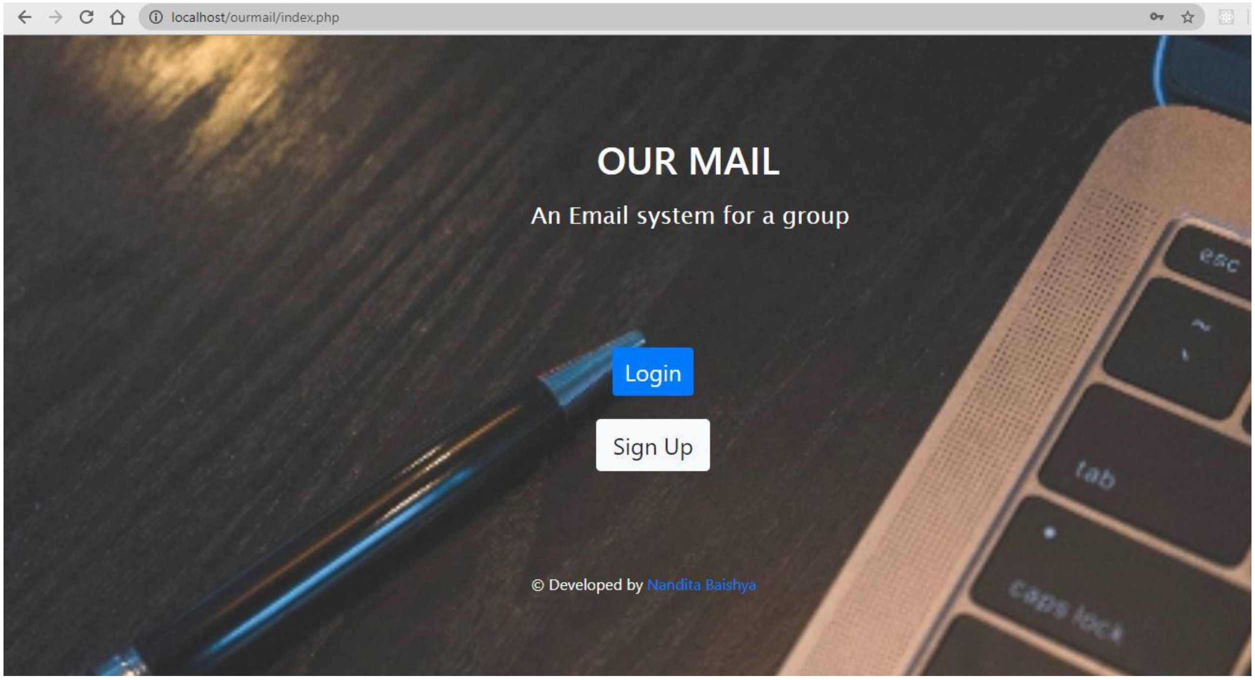
	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
□	1	messages_id	int(11)		UNSIGNED	No	*None*		AUTO_INCREMENT	Change Drop
□	2	header	varchar(100)	utf8mb4_general_ci		No	*None*			Change Drop
□	3	messages_sender	varchar(100)	utf8mb4_general_ci		No	*None*			Change Drop
□	4	messages_content	varchar(5000)	utf8mb4_general_ci		No	*None*			Change Drop
□	5	messages_date	timestamp			No	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()	Change Drop

□ Check all *With selected:* Browse Change Drop Primary Unique Index Fulltext Add to central columns

Remove from central columns

Application Snippets

1. Home page



localhost/ourmail/index.php

OUR MAIL

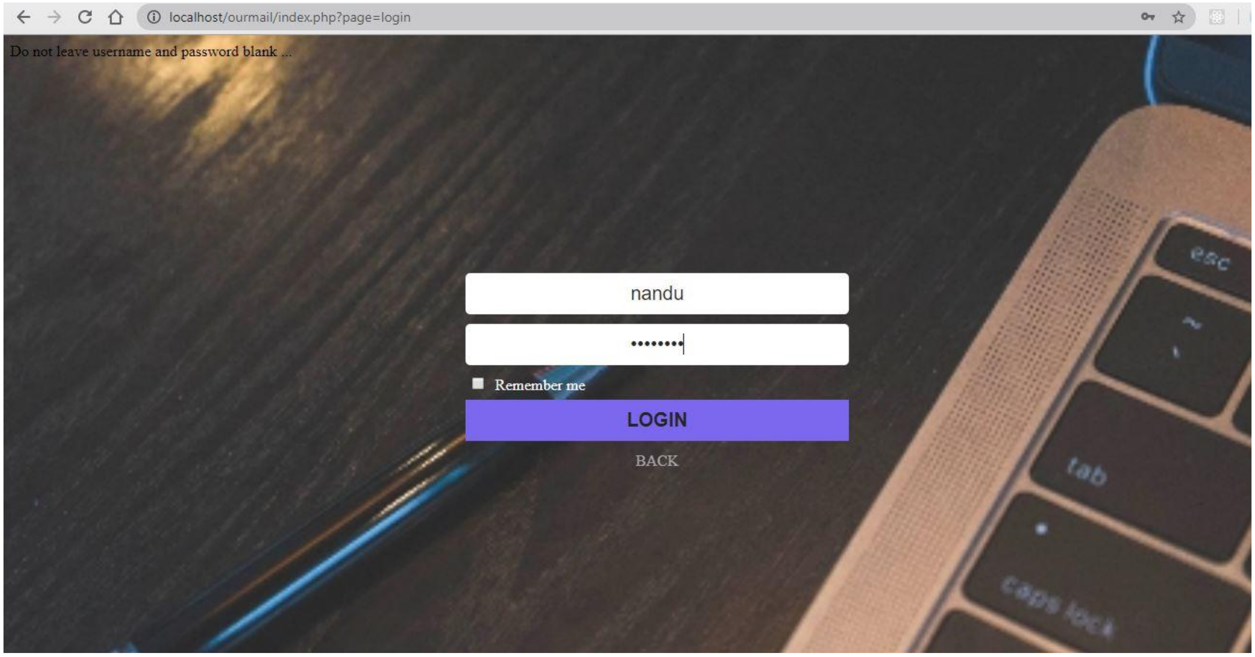
An Email system for a group

Login

Sign Up

© Developed by Nandita Baishya

2. Login page



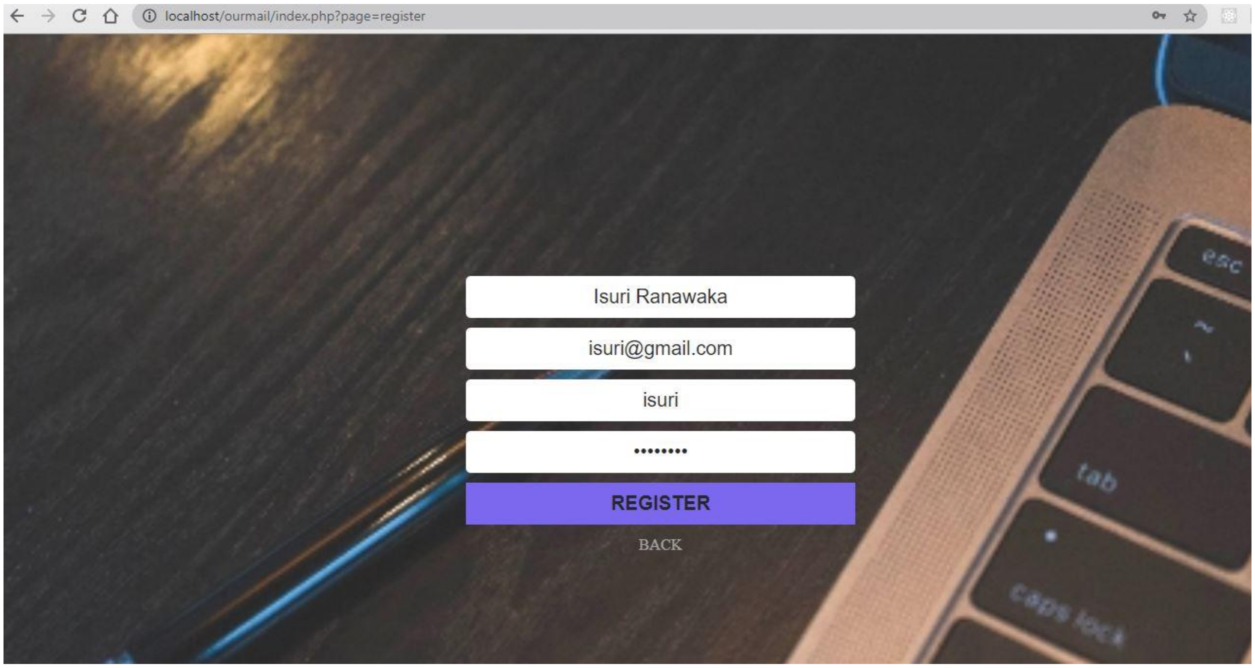
localhost/ourmail/index.php?page=login

Do not leave username and password blank ...

nandu

••••••••

3. Sign up page



localhost/ourmail/index.php?page=register

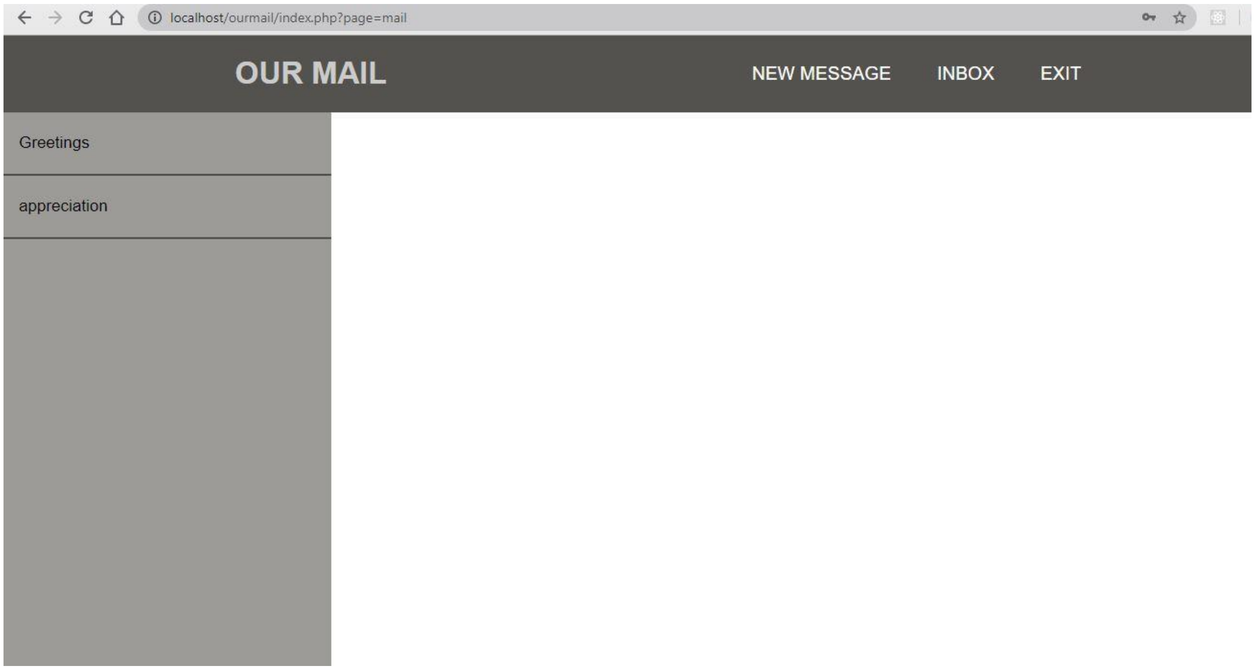
Isuri Ranawaka

isuri@gmail.com

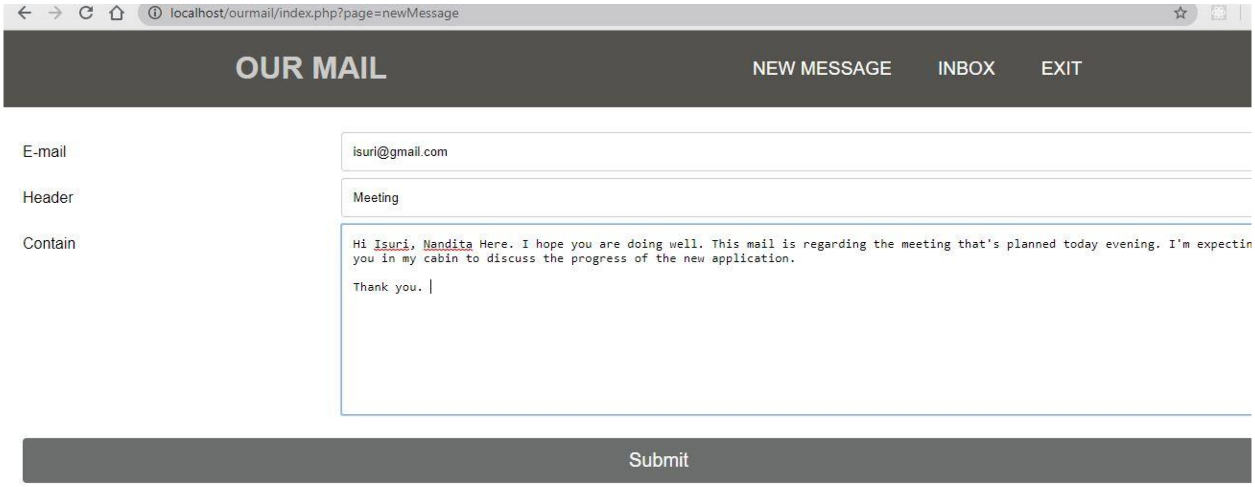
isuri

••••••••

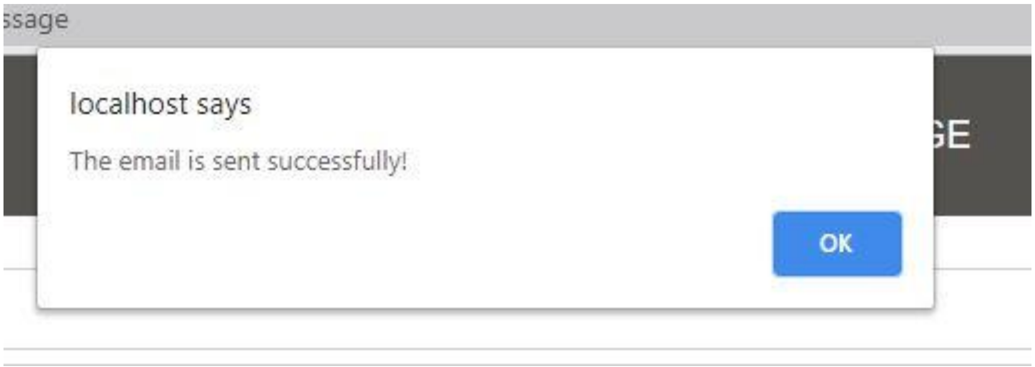
4. An Email Area



a) New Message page



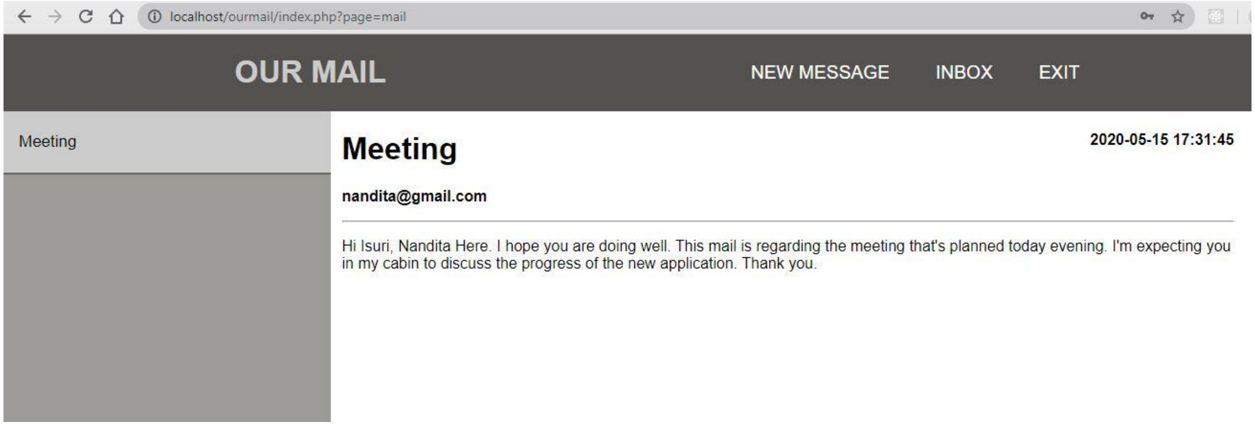
E-mail	isuri@gmail.com
Header	Meeting
Contain	Hi Isuri, Nandita Here. I hope you are doing well. This mail is regarding the meeting that's planned today evening. I'm expectin you in my cabin to discuss the progress of the new application. Thank you.



localhost says

The email is sent successfully!

b) Detailed Message display



Meeting

2020-05-15 17:31:45

nandita@gmail.com

Hi Isuri, Nandita Here. I hope you are doing well. This mail is regarding the meeting that's planned today evening. I'm expecting you in my cabin to discuss the progress of the new application. Thank you.

Implementation Summary

“OURMAIL” is a very useful tool in order to ensure a secure communication with in a group and an organization can add its own security measures to the system. Even though, the application emphasizes on the communication through email, it can easily be changed to other medium of communication handle such as user identity numbers, similar to the concept of apple “small talk”. The implementation bifurcates the functionality of a module and styling of a webpage using external cascading style sheets, which makes easy to modify the appearance of the application without tampering the core functions. Currently, this application is a miniscule version of an email system which satisfies the primary requirements, many features can be incorporated in order to make the application more robust, versatile and secure.

Written By:

Nandita Baishya