

**INFORMATICS INSTITUTE OF TECHNOLOGY**

**In Collaboration with**

**UNIVERSITY OF WESTMINSTER**

**Department of Computing  
B Eng. Software Engineering**

**HOLIDAY CONNECT:  
An Social Networking Application**

**Module: 6COSC022C Advanced Server-Side Web Programming**

**Module Leader: Mr. Ruwan Egodawatte**

**Coursework 2**

**<https://github.com/ChirathSiri/Holiday-Connect.git>**

<b>Student Name</b>	<b>Chirath Thusaraka Sirimanna</b>
<b>Student ID</b>	<b>20191168</b>
<b>UoW ID</b>	<b>W1809835</b>

**Date of Submission – 13<sup>th</sup> of May 2024**



## Table of Contents

<b>1. Holiday Connect Web Application URL .....</b>	<b>1</b>
<b>2. Source Code of Holiday Connect Web Application .....</b>	<b>1</b>
<b>2.1 Controller PHP Codes.....</b>	<b>1</b>
LoginCon.php.....	1
ProfileCon.php .....	5
Home.php .....	6
CreatePost.php .....	8
<b>2.2 Model PHP Codes.....</b>	<b>12</b>
UserModelFile.php .....	12
PostModelFile.php.....	16
<b>2.3 View PHP Codes .....</b>	<b>19</b>
login.php .....	19
signup.php .....	22
passwordreset.php.....	25
createpost.php.....	27
editprofile.php.....	30
home.php .....	34
location.php.....	37
myprofile.php .....	39
navigation.php.....	42
post.php .....	46
userprofile.php .....	50
<b>2.4 CSS Files.....</b>	<b>54</b>
Login.css.....	54
navigation_page.css .....	57
home.css .....	63
profile.css .....	68
viewpost.css .....	72
edituserprofile.css.....	76
createnewpost.css .....	79
<b>2.5 SQL Codes for Holiday Connect Database .....</b>	<b>82</b>



Table structure for table `comments` .....	82
Dumping data for table `comments` .....	82
Table structure for table `following` .....	83
Dumping data for table `following` .....	83
Table structure for table `likes` .....	84
Dumping data for table `likes` .....	84
Table structure for table `location` .....	85
Dumping data for table `location` .....	86
Table structure for table `notification` .....	93
Dumping data for table `notification` .....	93
Table structure for table `posts` .....	95
Dumping data for table `posts` .....	96
Table structure for table `users` .....	96
Dumping data for table `users` .....	97
<b>Indexes for dumped tables .....</b>	<b>98</b>
Indexes for table `comments` .....	98
Indexes for table `following` .....	98
Indexes for table `likes` .....	98
Indexes for table `location` .....	98
Indexes for table `notification` .....	99
Indexes for table `posts` .....	99
Indexes for table `users` .....	99
<b>AUTO_INCREMENT for dumped tables .....</b>	<b>99</b>
AUTO_INCREMENT for table `comments` .....	99
AUTO_INCREMENT for table `following` .....	100
AUTO_INCREMENT for table `likes` .....	100
AUTO_INCREMENT for table `notification` .....	100
AUTO_INCREMENT for table `posts` .....	100
AUTO_INCREMENT for table `users` .....	100
<b>Constraints for dumped tables .....</b>	<b>100</b>
Constraints for table `comments` .....	101
Constraints for table `following` .....	101
Constraints for table `likes` .....	101
Constraints for table `notification` .....	101
Constraints for table `posts` .....	102



<b>3. Evaluation of Design from Coursework 1 .....</b>	<b>103</b>
3.1 Changes to requirements from Coursework 1 .....	103
3.2 Database Script Changes.....	103
3.3 Class Model Changes .....	103
3.4 Discussion Paragraph .....	104
<b>4. Discussion of Article.....</b>	<b>105</b>
4.1 Question 1 .....	105
4.2 Question 2 .....	106
<b>5. References.....</b>	<b>107</b>



## 1. Holiday Connect Web Application URL

- GitHub Link: - <https://github.com/ChirathSiri/Holiday-Connect.git>

## 2. Source Code of Holiday Connect Web Application

### 2.1 Controller PHP Codes

#### LoginCon.php

```
<?php

defined('BASEPATH') OR exit('No direct script access allowed');

require APPPATH . '/libraries/REST_Controller.php';

class LoginCon extends \Restserver\Libraries\REST_Controller {

    function __construct() {
        parent::__construct();
        $this->load->model('UserModelFile');

        Header('Access-Control-Allow-Origin: *');
        Header('Access-Control-Allow-Headers: *');
        Header('Access-Control-Allow-Methods: GET, POST, OPTIONS, PUT, DELETE');
    }

    // Display the login interface
    public function index_get(){
        $this->load->view('login');
    }

    // Display the signup interface
    public function signup_get(){
        $this->load->view('signup');
    }

    public function login_get(){
        // Show login with an error message if the provided data is invalid
        if (isset($this->session->login_error) && $this->session->login_error ==
True) {

            $this->session->unset_userdata('login_error');
            $this->load->view('login',
```



```

        array('login_error_msg' => "Invalid Username or
Password. Try Again !"));
    }
    else {
        $this->load->view('login');
    }
}

// Update the session status to logged_out by setting logged_in to false
public function logout_get(){
    $this->session->is_logged_in = False;
    $this->login_get();
}

// Display the userprofile interface
public function userprofile_get(){
    if ($this->UserModelFile->is_logged_in()) {
        $username = $this->get('username');
        $this->load->view('navigation',array('username' => $this->session-
>username));
        $this->load->view('userprofile',array('username' => $username));
    }
    else {
        $this->load->view('login');
    }
}

// Endpoint to retrieve all users from the API
public function user_get(){
    $result = $this->UserModelFile->getUsers();
    $this->response($result);
}

public function user_post() {
    // If the action is "signup", initiate user action.
    if($this->get('action') == 'signup') {
        $username = $this->post('username');
        $password = $this->post('password');
        $email = $this->post('email');
        $name = $this->post('name');
        $result = $this->UserModelFile->create($username, $password, $email,
$name);
        if ($result === FALSE) {
            $this->response(array('result' => 'failed'));
        } else {
            $this->response(array('result' => 'success'));
        }
    }

    // If the action is "login", verify the data and include the user in the
session.

```



```

} else if($this->get('action') == 'login') {
    $username = $this->post('username');
    $password = $this->post('password');

    $result = $this->UserModelFile->login($username,$password);

    if ($result === false) {
        $this->session->login_error = True;
        $this->response(array('result' => 'failed'));
    }
    else {
        $this->session->is_logged_in = true;
        $this->session->username = $username;
        $this->response(array('result' => 'success'));
    }
    // If the action is "checker", verify whether the user is already present
    in the database.
} else if($this->get('action') == 'checkuser') {
    $username = $this->post('username');
    $result = $this->UserModelFile->checkUser($username);
    $this->response($result);
}

// If the action is "password-reset", modify the user's password.
else if($this->get('action') == 'passwordreset') {
    $username = $this->post('username');
    $password = $this->post('password');
    $result=$this->UserModelFile->passwordreset($username, $password);
    if ($result) {
        if ($this->UserModelFile->is_logged_in()){
            $this->response(array('result' => 'logged'));
        }
        else{
            $this->response(array('result' => 'success'));
        }
    }
    else {
        $this->response(array('result' => 'failed'));
    }
}

// If the action is "search user", retrieve user details.
else if($this->get('action') == 'searchuser') {
    if ($this->UserModelFile->is_logged_in()) {
        $username = $this->post('username');
        $result=$this->UserModelFile->searchUser($username);
        $this->response($result);
    }
    else {

```



```

        $this->load->view('login');
    }
}

//Display password reset interface
public function passwordreset_get(){
    $this->load->view('passwordreset');
}

// Endpoint to fetch details for the specified user from the API
public function userdetails_get() {
    if ($this->UserModelFile->is_logged_in()) {
        $username = $this->get('username');
        $userlist = $this->UserModelFile->getUser($username);
        if ($userlist) {
            $this->response($userlist);
        } else {
            $this->response(NULL);
        }
    }
    else {
        $this->load->view('login');
    }
}

// Update user details using a POST request.
public function editprofile_put(){
    if ($this->UserModelFile->is_logged_in()) {
        $username = $this->put('username');
        $bio = $this->put('bio');
        $name = $this->put('name');
        $email = $this->put('email');
        $userimage = $this->put('userimage');
        $result=$this->UserModelFile->editprofile($username, $bio, $name,
$email, $userimage);
        if ($result) {
            $this->response(array('result' => 'done'));
        }
        else {
            $this->response(array('result' => 'fail'));
        }
    }
    else {
        $this->load->view('login');
    }
}
}

```





## ProfileCon.php

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

require APPPATH . '/libraries/REST_Controller.php';

class ProfileCon extends \Restserver\Libraries\REST_Controller {

    public function __construct() {
        parent::__construct();
        $this->load->model('UserModelFile');
        $this->load->model('PostModelFile');

        Header('Access-Control-Allow-Origin: *');
        Header('Access-Control-Allow-Headers: *');
        Header('Access-Control-Allow-Methods: GET, POST, OPTIONS, PUT, DELETE');
    }
    //The index method to display my profile page.
    public function index_get(){
        if ($this->UserModelFile->is_logged_in()) {
            $this->load->view('navigation',array('username' => $this->session->username));
            $this->load->view('myprofile',array('username' => $this->session->username));
        }
        else {
            $this->load->view('login');
        }
    }
    //The edit profile view.
    public function editprofile_get(){
        if ($this->UserModelFile->is_logged_in()) {
            $this->load->view('navigation',array('username' => $this->session->username));
            $this->load->view('editprofile',array('username' => $this->session->username));
        }
        else {
            $this->load->view('login');
        }
    }
    //An API endpoint to retrieve details of a user's posts.
    public function myposts_get(){
        $username = $this->session->username;
        $result = $this->PostModelFile->getPostsfomUsername($username);
        $this->response($result);
    }
}
```



```

//A POST API endpoint to follow a user.
public function follow_post(){
    if ($this->UserModelFile->is_logged_in()) {
        $username = $this->session->username;
        $isfollowing = $this->post('isfollowing');
        $result=$this->UserModelFile->followuser($username, $isfollowing);
        $this->response($result);
    }
    else {
        $this->load->view('login');
    }
}

//An API endpoint to check if a user is already following another user.
public function checkfollow_get(){
    if ($this->UserModelFile->is_logged_in()) {
        $username = $this->session->username;
        $isfollowing = $this->get('isfollowing');
        $result=$this->UserModelFile->checkfollowing($username, $isfollowing);
        $this->response($result);
    }
    else {
        $this->load->view('login');
    }
}

//An API endpoint to retrieve follower/following count.
public function followcount_get(){
    $username = $this->get('username');
    $result=$this->UserModelFile->followcount($username);
    $this->response($result);
}

//An API endpoint to retrieve all notifications for a user.
public function notifications_get(){
    if ($this->UserModelFile->is_logged_in()) {
        $username = $this->session->username;
        $result=$this->UserModelFile->notifications($username);
        $this->response($result);
    }
    else {
        $this->load->view('login');
    }
}
}
}

```

## Home.php

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');

```



```

require APPPATH . '/libraries/REST_Controller.php';

class Home extends \Restserver\Libraries\REST_Controller {

    public function __construct() {
        parent::__construct();
        $this->load->model('UserModelFile');
        $this->load->model('PostModelFile');

        Header('Access-Control-Allow-Origin: *');
        Header('Access-Control-Allow-Headers: *');
        Header('Access-Control-Allow-Methods: GET, POST, OPTIONS, PUT, DELETE');
    }
    //Holiday Connect Home Page View
    public function index_get()
    {
        //Verify if the user is logged in; if not, redirect them to the login page.
        if ($this->UserModelFile->is_logged_in()) {
            $this->load->view('navigation',array('username' => $this->session->
            >username));
            $this->load->view('home',array('username' => $this->session->
            >username));
        }
        else {
            $this->load->view('login');
        }
    }
    //An API endpoint to retrieve posts from users being followed.
    public function followingposts_get(){
        if ($this->UserModelFile->is_logged_in()) {
            $username = $this->get('username');
            $result=$this->PostModelFile->getPostsofFollowing($username);
            $this->response($result);
        }
        else {
            $this->load->view('login');
        }
    }
    //An API endpoint to retrieve comments on posts.
    public function comments_get(){
        if ($this->UserModelFile->is_logged_in()) {
            $postid = $this->get('postid');
            $result=$this->PostModelFile->getComments($postid);
            $this->response($result);
        }
        else {
            $this->load->view('login');
        }
    }
    //A POST API request to add comments.

```



```

    public function comments_post(){
        if ($this->UserModelFile->is_logged_in()) {
            $username = $this->session->username;
            $postid = $this->post('postid');
            $comment = $this->post('comment');
            $result=$this->PostModelFile->addComments($postid, $comment,
$username);
            $this->response($result);
        }
        else {
            $this->load->view('login');
        }
    }
}
//An API endpoint to check if a user has already liked a post.
public function checklikes_get(){
    if ($this->UserModelFile->is_logged_in()) {
        $username = $this->session->username;
        $postid = $this->get('postid');
        $result=$this->PostModelFile->checklikes($username, $postid);
        $this->response($result);
    }
    else {
        $this->load->view('login');
    }
}
}
//POST requests to like posts.
public function like_post(){
    if ($this->UserModelFile->is_logged_in()) {
        $username = $this->session->username;
        $username = $this->post('username');
        $postid = $this->post('postid');
        $result=$this->PostModelFile->likepost($username, $postid);
        $this->response($result);
    }
    else {
        $this->load->view('login');
    }
}
}
}

```

### CreatePost.php

```
<?php
```

```

defined('BASEPATH') OR exit('No direct script access allowed');

require APPPATH . '/libraries/REST_Controller.php';

```



```

class CreatePost extends \Restserver\Libraries\REST_Controller {

    function __construct() {
        parent::__construct();
        $this->load->model('UserModelFile');
        $this->load->model('PostModelFile');

        Header('Access-Control-Allow-Origin: *');
        Header('Access-Control-Allow-Headers: *');
        Header('Access-Control-Allow-Methods: GET, POST, OPTIONS, PUT, DELETE');
    }

    //Set up routes to render the create post view.
    public function index_get(){
        if ($this->UserModelFile->is_logged_in()) {
            $this->load->view('navigation',array('username' => $this->session->username));
            $this->load->view('createpost');
        }
        else {
            $this->load->view('login');
        }
    }

    //Save the post image in the designated folder.
    public function store_post() {
        if ($this->UserModelFile->is_logged_in()) {
            $config['upload_path'] = './images/userposts/";//path
            $config['allowed_types'] = 'gif|jpg|png';//file types allowed
            $this->load->library('upload', $config);
            if (!$this->upload->do_upload('image')) {
                $error = array('result'=>'failed','error' => $this->upload->display_errors());
                $this->response($error);
            } else {
                $data = array('result'=>'done','image_metadata' => $this->upload->data());
                $this->response($data);
            }
        }
        else {
            $this->load->view('login');
        }
    }

    //Save the profile picture image in the designated folder.
    public function propic_post() {
        if ($this->UserModelFile->is_logged_in()) {
            $config['upload_path'] = './images/profilepics/';
            $config['allowed_types'] = 'gif|jpg|png';
            $this->load->library('upload', $config);
            if (!$this->upload->do_upload('image')) {

```



```

        $error = array('result'=>'failed','error' => $this->upload-
>display_errors());
        $this->response($error);
    } else {
        $data = array('result'=>'done','image_metadata' => $this->upload-
>data());
        $this->response($data);
    }
}
else {
    $this->load->view('login');
}
}

//A POST request to create a new post.
public function create_post() {
    if ($this->UserModelFile->is_logged_in()) {
        $username = $this->session->username;
        $locationid = $this->post('locationid');
        $postImage = $this->post('postImage');
        $caption = $this->post('caption');
        $result = $this->PostModelFile->createPost($username, $locationid,
$postImage, $caption);

        $this->response($result);
        if ($result) {
            $this->response(array('result' => 'done'));
        } else {
            $this->response(array('result' => 'failed'));
        }
    }
    else {
        $this->load->view('login');
    }
}

//An API endpoint to retrieve all posts from a specific user.
public function userposts_get(){
    $username = $this->get('username');
    $result = $this->PostModelFile->getPostsfromUsername($username);
    $this->response($result);
}

public function location_get() {
    //The action retrieves all locations.
    if($this->get('action') == 'all') {
        $locations = $this->PostModelFile->getLocations();
        if ($locations) {
            $this->response($locations);
        } else {
            $this->response(NULL);
        }
    }
}

```



```

    }
    //The action retrieves the post by its ID.
    if($this->get('action') == 'id') {
        $locationid = $this->get('locationid');
        $locations = $this->PostModelFile->getLocationById($locationid);
        $this->response($locations);
    }
}
//Retrieve the location view.
public function locations_get() {
    if ($this->UserModelFile->is_logged_in()) {
        $locationid = $this->get('locationid');
        $this->load->view('navigation',array('username' => $this->session->username));
        $this->load->view('locations',array('locationid' => $locationid));
    }
    else {
        $this->load->view('login');
    }
}
public function post_get() {
    if ($this->UserModelFile->is_logged_in()) {
        $postid = $this->get('postid');
        //If the action is "view", retrieve post details based on its ID.
        if($this->get('action') == 'view') {
            $result = $this->PostModelFile->postfromid($postid);
            $this->response($result);
        }
        //Otherwise, load the post view.
        else{
            $this->load->view('navigation',array('username' => $this->session->username));
            $this->load->view('post',array('postid' => $postid,'username' => $this->session->username));
        }
    }
    else {
        $this->load->view('login');
    }
}
//An API endpoint to retrieve posts from a specified location.
public function locationposts_get(){
    if ($this->UserModelFile->is_logged_in()) {
        $locationid = $this->get('locationid');
        $result = $this->PostModelFile->postsFromLocation($locationid);
        $this->response($result);
    }
    else {
        $this->load->view('login');
    }
}

```



```

    }
}
//An API endpoint to retrieve the like count.
public function likecount_get(){
    if ($this->UserModelFile->is_logged_in()) {
        $postid = $this->get('postid');
        $result = $this->PostModelFile->likeCount($postid);
        $this->response($result);
    }
    else {
        $this->load->view('login');
    }
}
}
}

```

## 2.2 Model PHP Codes

### UserModelFile.php

```

<?php
defined('BASEPATH') or exit('No direct script access allowed');

// Model for user-related database tasks
class UserModelFile extends CI_Model
{
    public function __construct()
    {
        parent::__construct();
        $this->load->database();
    }

    // Insert user data during sign-up
    function create($username, $password, $email, $name)
    {
        $hashed = password_hash($password, PASSWORD_DEFAULT);
        $data = array('Username' => $username, 'Password' => $hashed, 'Email' =>
$email, 'Name' => $name);

        if ($this->db->insert('users', $data)) {
            return True;
        } else {
            return False;
        }
    }

    // Model to fetch all data from the user table

```





```

public function getUsers()
{
    $query = $this->db->get('users');
    if ($query) {
        return $query->result();
    }
    return NULL;
}

// Model to verify user credentials during login
function login($username, $password)
{
    $res = $this->db->get_where('users', array('Username' => $username));
    if ($res->num_rows() != 1) {
        return false;
    } else {
        $row = $res->row();
        if (password_verify($password, $row->Password)) {
            return true;
        } else {
            return false;
        }
    }
}

// Model to check if a user is logged in session
function is_logged_in()
{
    if (isset($this->session->is_logged_in) && $this->session->is_logged_in ==
True) {
        return True;
    } else {
        return False;
    }
}

// Model for updating the password column of the user table
public function passwordreset($username, $password){
    $hashed = password_hash($password, PASSWORD_DEFAULT);
    $res = $this->db->get_where('users', array('Username' => $username));
    if ($res->num_rows() != 1){
        return false;
    }
    else{
        $dataToChange = array('Password' => $hashed);
        $id = array('Username' => $username);
        $query = $this->db->update('users', $dataToChange, array('Username' =>
$username));
        if ($query) {

```



```

        return $query;
    }
    else{
        return false;
    }
}

}

// Model for searching users in the user table
public function searchUser($username){
    $query=$this->db->query("SELECT * FROM users WHERE Username LIKE
    '".$username."%");
    return $query->result();
}

// Model for handling follow actions and notification updates
public function followuser($username,$isfollowing){
    $users = $this->db->get_where('users', array('Username' => $username));
    $userId= $users->row()->UserId;
    $following = $this->db->get_where('users', array('Username' =>
    $isfollowing));
    $isfollowing= $following->row()->UserId;
    $res = $this->db->get_where('following', array('UserId' =>
    $userId,'IsFollowing' => $isfollowing));
    if ($res->num_rows() == 1){
        $query=$this->db->delete('following', array('UserId' =>
    $userId,'IsFollowing' => $isfollowing));
        $this->db->delete('notification', array('FromUser' => $userId,'UserId'
    => $isfollowing,'Notification'=>'Followed you!'));
        return "deleted";
    }
    else{
        $query=$this->db->insert('following', array('UserId' =>
    $userId,'IsFollowing' => $isfollowing));
        $this->db->insert('notification', array('FromUser' => $userId,'UserId'
    => $isfollowing,'Notification'=>'Followed you!'));
        return "added";
    }
}

// Model for querying if a user is following another user
public function checkfollowing($username,$isfollowing){
    $users = $this->db->get_where('users', array('Username' => $username));
    $userId= $users->row()->UserId;
    $following = $this->db->get_where('users', array('Username' =>
    $isfollowing));
    $isfollowing= $following->row()->UserId;
    $res = $this->db->get_where('following', array('UserId' =>
    $userId,'IsFollowing' => $isfollowing));

```



```

        if ($res->num_rows() == 1){
            return true;
        }
        else{
            return false;
        }
    }
}

// Model for getting counts of followers/following
public function followcount($username){
    $users = $this->db->get_where('users', array('Username' => $username));
    $userId= $users->row()->UserId;
    $res1 = $this->db->get_where('following', array('UserId' => $userId));
    $following=$res1->num_rows();
    $res2 = $this->db->get_where('following', array('IsFollowing' => $userId));
    $followers=$res2->num_rows();
    return array('following' => $following,'followers' => $followers);
}

//Retrieve all records from the "users" table where the username matches.
public function getUser($username)
{
    $res = $this->db->get_where('users', array('Username' => $username));
    return $res->row();
}

//A query to verify the user's availability.
public function checkUser($username)
{
    $res = $this->db->get_where('users', array('Username' => $username));
    return $res->num_rows();
}

//Update records in the user table.
public function editprofile($username, $bio, $name, $email, $userimage){
    $query=$this->db->query("UPDATE users SET
Name='".$name."',Email='".$email."',UserBio='".$bio."',UserImage='".$userimage.'"
WHERE Username='".$username.'";
    return $query;
}

//Retrieve all notifications associated with a user.
public function notifications($username){
    $users = $this->db->get_where('users', array('Username' => $username));
    $userId= $users->row()->UserId;

```



```

        $query = $this->db->query("SELECT notification.*, users.Username FROM
notification JOIN users ON users.UserId=notification.FromUser WHERE
notification.UserId='".$userId.'" ORDER BY Timestamp DESC");
        if ($query) {
            return $query->result();
        }
        return NULL;
    }
}

```

### PostModelFile.php

```

<?php
defined('BASEPATH') or exit('No direct script access allowed');

//A model for handling tasks related to posts in the database.
class PostModelFile extends CI_Model
{
    public function __construct()
    {
        parent::__construct();
        $this->load->database();
    }

    //Insert rows into the post table.
    function createPost($username, $locationid, $postImage, $caption)
    {
        $users = $this->db->get_where('users', array('Username' => $username));
        $userId = $users->row()->UserId;
        $data = array('UserId' => $userId, 'LocationId' => $locationid, 'PostImage'
=> $postImage, 'Caption' => $caption);
        if ($this->db->insert('posts', $data)) {
            return True;
        } else {
            return False;
        }
    }

    //Query the post table to retrieve posts authored by a specific user.
    function getPostsfromUsername($username)
    {
        $users = $this->db->get_where('users', array('Username' => $username));
        $userId = $users->row()->UserId;
        $query=$this->db->query( "SELECT * FROM posts WHERE UserId='".$userId.'"
ORDER BY Timestamp DESC");
        return $query->result();
    }
}

```



```

//Retrieve all locations from the database.
function getLocations(){
    $query = $this->db->get('location');
    if ($query) {
        return $query->result();
    }
    return NULL;
}

//Query the database to retrieve posts from users being followed.
function getPostsofFollowing($username){
    $users = $this->db->get_where('users', array('Username' => $username));
    $userId= $users->row()->UserId;
    $query=$this->db->query("SELECT posts.*, users.Username,
location.LocationName FROM posts JOIN users ON users.UserId=posts.UserId JOIN
location ON location.LocationId=posts.LocationId WHERE posts.userid IN
(SELECT IsFollowing FROM following WHERE following.UserId=".$userId.")
ORDER BY Timestamp DESC");
    return $query->result();
}

//Query the table to retrieve comments for each post.
function getComments($postid){
    $comments = $this->db->query( "SELECT comments.*,users.Username FROM
comments JOIN users ON users.UserId=comments.UserId WHERE PostId=".$postid." ORDER
BY Timestamp DESC");
    return $comments->result();
}

//Insert rows into the comments table.
function addComments($postid, $comment, $username){
    $users = $this->db->get_where('users', array('Username' => $username));
    $userId= $users->row()->UserId;
    $posts = $this->db->get_where('posts', array('PostId' => $postid));
    $postuser= $posts->row()->UserId;
    $query=$this->db->insert('comments', array('UserId' => $userId,'PostId' =>
$postid,'CommentBody' => $comment));
    $this->db->insert('notification', array('FromUser' => $userId,'UserId' =>
$postuser, 'PostId' => $postid, 'CommentBody' => $comment,
'Notification'=>'Commented on your post!'));
    return $query;
}

//Query the like table to verify if a user has liked a post.
public function checklikes($username, $postid){
    $users = $this->db->get_where('users', array('Username' => $username));
    $userId= $users->row()->UserId;
    $res = $this->db->get_where('likes', array('UserId' => $userId,'PostId' =>
$postid));

```



```

        if ($res->num_rows() == 1){
            return true;
        }
        else{
            return false;
        }
    }
}

//Insert a row into both the likes and notification tables.
public function likepost($username, $postid){
    $users = $this->db->get_where('users', array('Username' => $username));
    $userId= $users->row()->UserId;
    $posts = $this->db->get_where('posts', array('PostId' => $postid));
    $postuser= $posts->row()->UserId;
    $res = $this->db->get_where('likes', array('UserId' => $userId, 'PostId' =>
$postid));
    if ($res->num_rows() == 1){
        $query=$this->db->delete('likes', array('UserId' => $userId, 'PostId' =>
$postid));
        $this->db->delete('notification', array('FromUser' => $userId, 'UserId'
=> $postuser, 'PostId' => $postid, 'Notification'=>'Liked your post!'));
        return "deleted";
    }
    else{
        $query=$this->db->insert('likes', array('UserId' => $userId, 'PostId' =>
$postid));
        $this->db->insert('notification', array('FromUser' => $userId, 'UserId'
=> $postuser, 'PostId' => $postid, 'Notification'=>'Liked your post!'));
        return "added";
    }
}

//Query the post table to retrieve posts from a specific location.
public function postsFromLocation($locationid){
    $res = $this->db->get_where('posts', array('LocationId' => $locationid));
    return $res->result();
}

//Query locations by their location ID.
public function getLocationById($locationid){
    $res = $this->db->get_where('location', array('LocationId' =>
$locationid));
    return $res->row();
}

//Query posts by their post ID.
public function postfromid($postid){

```



```

        $res = $this->db->query( "SELECT posts.*, users.Username, users.UserImage,
location.LocationName FROM posts JOIN users ON users.UserId=posts.UserId JOIN
location ON location.LocationId=posts.LocationId WHERE posts.PostId =".$postid);
        return $res->row();
    }

    //Retrieve the number of rows from the likes table based on the post ID.
    public function likeCount($postid){
        $res2 = $this->db->get_where('likes', array('PostId' => $postid));
        $likes=$res2->num_rows();
        return $likes;
    }
}

```

## 2.3 View PHP Codes

### login.php

```

<?php defined('BASEPATH') OR exit('No direct script access allowed'); ?>
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Holiday Connect</title>

    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"
type="text/javascript"></script>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/underscore.js/1.8.3/underscore-min.js"
type="text/javascript"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/backbone.js/1.2.3/backbone-
min.js" type="text/javascript"></script>
    <link rel="stylesheet" type="text/css" href="<?php echo
base_url()?>css/login.css">

</head>

<body>
<!-- Logo section -->
<div class="logodiv">
    <!-- Logo image -->
    
</div>
<!-- Login form section -->
<div class="loginform">

    <div class="logheading">

```



```

        <span>Welcome !</span>
    </div>

    <!-- error message if necessary -->

    <?php if (isset($login_error_msg)) { ?>
    <div class="errmsg">
        <?php echo $login_error_msg ?>
    </div> <?php } ?>

    <form class="authforms" name="loginform">
        <!-- Username input -->
        <div class="input">
            <input class="loginfield" type="text" id="username" name='username'
onkeyup='checkinputs();' required/>

            <label class="loginlabel"> Username :
                <span style="color:#EB9494">*</span>
            </label>

        </div>

        <div class="input">
            <!-- Password input -->
            <input class="loginfield" type="password" id="password" name='password'
onkeyup='checkinputs();' required/>

            <label class="loginlabel"> Password :
                <span style="color:#EB9494">*</span>
            </label>
        </div>
        <!-- Login button -->
        <div class="action">
            <input class="loginbtn" type="submit" disabled="disabled" id="login"
value="LOGIN" />
        </div>
    </form>

    <div class="loginspandiv">

        <a href="<?php echo base_url()?>index.php/logincon/passwordreset">Forgot
Your Password Here ?</a>
        <br>
        <span>Don't have an account? Please
            <a href="<?php echo base_url()?>index.php/logincon/signup">Sign Up</a>
            here !
        </span>
    </div>

```





```

</div>

<script type="text/javascript" lang="javascript">
    //check if all inputs are not empty
    function checkinputs() {
        if (document.forms["loginform"]["username"].value != "" &&
document.forms["loginform"]["password"].value != "") {
            document.getElementById('login').disabled = false;
        }
        else{
            document.getElementById('login').disabled = true;
        }
    }
    $(document).ready(function () {
        //when login is clicked
        $('#login').click(function (event) {
            event.preventDefault();
            userLogin();
        });
    });
    // Define Backbone Model for handling login data
    var Login = Backbone.Model.extend({
        // Specify the URL for the login action on the server
        url:"<?php echo base_url()?>index.php/logincon/user/action/login"
    });
    // Define Backbone Collection for managing groups of login models
    var LoginCollection = Backbone.Collection.extend({
        // Specify the type of model that this collection will contain
        model: Login, // This collection contains instances of the Login model
    });

    var loginCollection = new LoginCollection();
    function userLogin() {
        var newLogin = new Login();
        newLogin.set('username',
"@"+$("#username").val().toLowerCase()); //username is converted to lowercase
        newLogin.set('password', $("#password").val());
        loginCollection.create(newLogin,{
            success: function(response){
                var result=response.changed.result;
                if (result=="success") { //redirect to home
                    location.href="<?php echo base_url()?>index.php/home/";
                } else { //else redirect to login
                    location.href="<?php echo
base_url()?>index.php/logincon/login";
                }
            }
        });
    }
}

```



```

    </script>
</body>

<footer>
    &copy; Holiday Connect 2024 | W1809835
</footer>

</html>

```

### signup.php

```

<?php defined('BASEPATH') or exit('No direct script access allowed'); ?>
<!DOCTYPE html>
<html lang="en">

<head>
    <title>Holiday Connect Signup</title>

    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"
        type="text/javascript"></script>

    <script
src="https://cdnjs.cloudflare.com/ajax/libs/underscore.js/1.8.3/underscore-min.js"
        type="text/javascript"></script>

    <script src="https://cdnjs.cloudflare.com/ajax/libs/backbone.js/1.2.3/backbone-
min.js"
        type="text/javascript"></script>

    <link rel="stylesheet" type="text/css" href="<?php echo base_url()
?>css/login.css">

</head>

<body>
    <div class="logodiv"> </div>

    <div class="signform">
        <div class="logheading">
            <span> Enter Your Details </span>
        </div>

        <form class="authforms" name="signupform">
            <!-- Error message section -->
            <div class="errmsgsg" id="errmsgsg"></div>
            <!-- Username input -->

```



```

<div class="input">

    <input class="signfield" type="text" id="username" name='username'
        onkeyup='checkusername(); checkinputs();' required />
    <label class="signlabel">Enter Username :<span
style="color:#EB9494">*</span></label>
</div>
<!-- Email input -->
<div class="input">
    <input class="signfield" type="text" id="email" name='email'
onkeyup='checkinputs(); validateemail()'
        required />
    <label class="signlabel">Enter Email :<span
style="color:#EB9494">*</span></label>
</div>
<!-- Full name input -->
<div class="input">
    <input class="signfield" type="text" id="name" name='name'
onkeyup='checkinputs();' required />
    <label class="signlabel">Enter Full Name :</label>
</div>
<!-- Password input -->
<div class="input">
    <input class="signfield" type="password" id="password"
name='password' onkeyup='checkinputs();'
        required />
    <label class="signlabel">Enter Password :<span
style="color:#EB9494">*</span></label>
</div>
<!-- Submit button -->
<div class="action">
    <input class="signupbtn" type="submit" id="createUser"
disabled="disabled" value="SIGN UP" />
</div>
</form>
<div class="signspandiv">
    <span>Already have a Holiday CONNECT account? Please <a href="<?php
echo base_url() ?>index.php/logincon/login">Login</a> </span>
</div>
</div>

<script type="text/javascript" lang="javascript">
// Check if the required inputs are empty.
function checkinputs() {
    if (document.forms["signupform"]["username"].value != "" &&
document.forms["signupform"]["email"].value != "" &&
        document.forms["signupform"]["password"].value != "" &&
document.getElementById("errmsg").innerHTML == "") {
        document.getElementById('createUser').disabled = false;

```



```

    }
    else{
        document.getElementById('createUser').disabled = true;
    }
}

function validateemail() {// Validate the email before sending it through the
API.
    var x = document.forms["signupform"]["email"].value;
    var atposition = x.indexOf("@");
    var dotposition = x.lastIndexOf(".");
    if (atposition < 1 || dotposition < atposition + 2 || dotposition + 2 >=
x.length) {
        document.getElementById("errmsg").innerHTML = "Please enter a valid
Email Address !";
    }
    else {
        document.getElementById("errmsg").innerHTML = "";
        checkinputs();
    }
}

function checkusername() {// Verify if the username is already taken or not.
$.ajax({
    url: "<?php echo base_url()
?>index.php/logincon/user/action/checkuser",
    data: { 'username': "@" + $('#username').val().toLowerCase() },
    method: "POST"// Utilize the POST method.
}).done(function (data) {
    if (data == 0) {
        document.getElementById("errmsg").innerHTML = "";
        checkinputs();
    }
    else {
        document.getElementById("errmsg").innerHTML = "User Already
Exists !"
    }
});
});

$(document).ready(function () {
    $('#createUser').click(function (event) {// Invoke a function when the
button is clicked.
        event.preventDefault();
        userSignup();
    });
});

// Define Backbone Model for handling signup data
var User = Backbone.Model.extend({
    // Specify the URL for the signup action on the server
    url: "<?php echo base_url() ?>index.php/logincon/user/action/signup"
});

```



```
// Define Backbone Collection for managing groups of signup models
var UserCollection = Backbone.Collection.extend({
    // Specify the type of model that this collection will contain
    model: User// This collection contains instances of the signup model
});
var usersCollection = new UserCollection();
function userSignup() {
    var newUser = new User();
    newUser.set('username', "@" + $("#username").val().toLowerCase());//
Convert the username to lowercase.
    newUser.set('email', $("#email").val());
    newUser.set('name', $("#name").val());
    newUser.set('password', $("#password").val());// Redirect back to the home
page.
    usersCollection.create(newUser, {
        success: function () {// Redirect to the login success page.
            location.href = "<?php echo base_url() ?>index.php/logincon/login";
        }
    });
}
</script>
</body>

</html>
```

### passwordreset.php

```
<?php defined('BASEPATH') OR exit('No direct script access allowed'); ?>
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Holiday Connect Reset Password</title>

    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"
        type="text/javascript"></script>
    <link rel="stylesheet"
        type="text/css" href="../../css/login.css">

</head>
<body>
    <!-- Logo section -->
<div class="logodiv">
    
</div>
<!-- Password reset form -->
<div class="pwform">
```



```

<!-- Heading -->
<div class="pwresetheading"><span> RESET YOUR PASSWORD </span></div>
<!-- Error message section -->
<div class="errormsg" id="errormsg"></div>
<!-- Form -->
<form class="authforms" name="loginform">
    <div class="input">
        <!-- Username input -->
        <input class="loginfield" type="text" id="username" name='username'
onkeyup='checkinputs();' required/>
        <label class="loginlabel"> Enter Your Username :<span
style="color:#EB9494">*</span></label>
    </div>
    <div class="input">
        <!-- Password input -->
        <input class="loginfield" type="password" id="password" name='password'
onkeyup='checkinputs();' required/>
        <label class="loginlabel"> Enter New Password :<span
style="color:#EB9494">*</span></label>
    </div>
    <!-- Submit button -->
    <div class="action">
        <input class="loginbtn" type="submit" disabled="disabled" id="changepw"
value="RESET" />
    </div>
</form>
<!-- Sign up link -->
<div class="loginspandiv">
    <span>Or <a href="<?php echo base_url()?>index.php/logincon/signup"> Sign
Up </a> Here !</span>
</div>
</div>

<script type="text/javascript" lang="javascript">
//check inputs to see if theyre are empty
function checkinputs() {
    if (document.forms["loginform"]["username"].value != "" &&
document.forms["loginform"]["password"].value != "") {
        document.getElementById('changepw').disabled = false;
    }
    else{
        document.getElementById('changepw').disabled = true;
    }
}
//when button is clicked to change password
$("#changepw").click(function(event) {
    event.preventDefault();
    var pwdata = {
        username: "@" + ($('#username').val()).toLowerCase(),

```



```

        password: $('#password').val()
    };
    $.ajax({
        url: "<?php echo base_url()
?>index.php/logincon/user/action/passwordreset",
        data: JSON.stringify(pwdata),
        contentType: "application/json",
        method: "POST"
    }).done(function (data) {
        var result = data.result;
        if (result == "success") {
            location.href = "<?php echo base_url() ?>index.php/logincon/login";
        }
        else if (result == "logged") { //if result is logged, means user is changing
pw while logged in
            location.href = "<?php echo base_url() ?>index.php/profilecon";
        }
        else {
            document.getElementById("errmsg").innerHTML = "Username Does Not
Exist !"
        }
    });
});
</script>
</body>

</html>

```

### createpost.php

```

<?php defined('BASEPATH') OR exit('No direct script access allowed'); ?>
<!DOCTYPE html>
<html>
<head>
    <title>Holiday Connect Create New Post</title>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"
type="text/javascript"></script>
    <link rel="stylesheet" type="text/css" href="<?php echo
base_url()?>css/createnewpost.css">
</head>
<body>
    <!-- Upload post container -->
    <div class="uppostcontainer">
        <!-- File upload section -->
        <div class="filediv">
            <!-- Error message section -->
            <div class="errmsg" id="errmsg"></div>

```



```

        <!-- Post picture display -->
        <div class="postpicdiv">
            <img id='postpicid' src='<?php echo base_url()
?>images/userposts/image.png' alt='picture' />
        </div>
        <!-- File input for image upload -->
        <input type="file" id="image" name="image" required="required"
onchange="readURL(this);"/>
        <!-- Upload picture label -->
        <div class="dummy"><p>Upload Picture</p></div>
    </div>
    <!-- Caption section -->
    <div class="captiondiv">
        <!-- Caption label -->
        <div class="caplabel"> <label for="caption">Write a
Caption...</label></div>
        <!-- Caption input -->
        <div><textarea name="caption" id="caption"
maxlength="100"></textarea></div>
        <!-- Location label -->
        <div class="loclabel"><label for="locations">Add a Location
Tag</label></div>
        <!-- Location select -->
        <div>
            <select onchange='getLocation();' id="locations">
                <option id='locationName' value=""></option>
            </select>
        </div>
    </div>
    <!-- Post submit button -->
    <div class="postsubmitdiv"><div id="uploadpost">Post</div></div>
</div>

<script type="text/javascript" lang="javascript">
    var postImage="";
    var $locationid = "1";
    // Load location posts at startup and display them in the dropdown.
    $.ajax({
        url: "<?php echo base_url()
?>index.php/createpost/location/action/all",
        method: "GET"
    }).done(function (data) {
        $('#locations option').remove();
        for (i = 0; i < data.length; i++) {
            var option = "<option id='locationName'
value="+data[i].LocationId+">"+data[i].LocationName+"</option>";
            $('#locations').append(option);
        }
    });

```





```

// Retrieve the location value from the form element.
function getLocation() {
    $locationid = document.getElementById("locations").value;
}

// Display the uploaded image.
function readURL(input) {
    document.getElementById("errmsg").innerHTML = "";
    if (input.files && input.files[0]) {
        var reader = new FileReader();
        var formdata = new FormData();
        var files = $('#image')[0].files;
        if(files.length > 0 ){
            formdata.append('image',files[0]);
            reader.onload = function (e) {
                $('#postpicid').attr('src', e.target.result);
            };
            reader.readAsDataURL(input.files[0]);
        }
    }
}

// Function called when the upload button is clicked.
$("#uploadpost").click(function(event) {
    event.preventDefault();
    var formdata = new FormData();
    var files = $('#image')[0].files;
    if(files.length > 0 ){
        formdata.append('image',files[0]);
    }
    $.ajax({
        url: "<?php echo base_url() ?>index.php/createpost/store", // Store
the image in the folder.
        data: formdata,
        method: "POST",
        contentType: false,
        processData: false
    }).done(function (data) {
        var result=data.result;
        if (result=="done") {
            $postImage = data.image_metadata.file_name; // Retrieve the
filename from the saved image.
            var postdata = {
                locationid: $locationid,
                postImage: $postImage,
                caption: $('#caption').val()
            };
            $.ajax({
                url: "<?php echo base_url() ?>index.php/createpost/create",
// Send data to the database.
                data: JSON.stringify(postdata),
                contentType: "application/json",

```



```

        method: "POST"
    }).done(function (data) {
        var result = data.result;
        if (result == "done") { // Redirect to my profile upon
success.
            location.href="<?php echo
base_url()?>index.php/profilecon";
        }
        else { // Otherwise, display an error message.
            document.getElementById("errmsg").innerHTML = "Post
Creation Unsuccessful !";
        }
    });
} else {
    $error=data.error.slice(3,-4);
    document.getElementById("errmsg").innerHTML = $error;
}
});
}
else{
    document.getElementById("errmsg").innerHTML = "Please Upload a
Image !";
}
});
</script>
</body>

<footer>
    &copy;Holiday Connect 2024 | W1809835
</footer>

</html>

```

### editprofile.php

```

<?php defined('BASEPATH') OR exit('No direct script access allowed'); ?>
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Holiday Connect Edit Profile</title>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"
type="text/javascript"></script>
    <link rel="stylesheet" type="text/css" href="<?php echo
base_url()?>css/edituserprofile.css">
</head>

<body>

```



```

<div class="editprofcontainer">
  <!-- Text editing section -->
  <div class="texteditdiv">
    <!-- Error message section -->
    <div class="errormsg" id="errormsg2"></div>
    <!-- Name change section -->
    <div class="namechangediv"></div>
    <!-- Email change section -->
    <div class="emailchangediv"></div>
    <!-- Bio change section -->
    <div class="biochangediv"></div>
    <!-- Edit image icon -->
    </a>
    <!-- Change password link -->
    <div class="changepwdiv"><a href="<?php echo
base_url() ?>index.php/logincon/passwordreset">Reset Password</a></div>
  </div>
  <!-- Profile picture editing section -->
  <div class="piceditdiv">
    <!-- Error message section -->
    <div class="errormsg" id="errormsg"></div>
    <!-- Profile image display section -->
    <div class="profimagediv"></div>
    <!-- File input for image upload -->
    <input type="file" id="image" name="image" onchange="readURL(this);" />
    <!-- Upload profile picture label -->
    <div class="dummy">Upload Profile Picture</div>
  </div>
  <!-- Save changes button -->
  <div class="editprofdiv"><div id="editprofile"> Save Changes </div></div>
</div>

<script type="text/javascript" lang="javascript">
  var username="<?php echo $username ?>";
  var userimage="";
  var bio="";
  var name="";
  var email="";

  $(document).ready(function () {
    event.preventDefault();
    // Fetch user details upon loading and render them.
    $.ajax({
      url: "<?php echo base_url()
?>index.php/logincon/userdetails?username="+username,
      method: "GET"
    }).done(function (data) {
      userimage=data.UserImage;
      bio=data.UserBio;
    });
  });

```



```

        name=data.Name;
        email=data.Email;
        var div ="<img id='profpicid' src='<?php echo base_url()
?>images/profilepics/"+data.UserImage+" alt='picture' />";
        $('.profimagediv').append(div);
        var namediv="<div class='labeldiv'>Name</div><div
class='inputdiv'><input class='inputedit' onkeyup='getname()' type='text' id='name'
name='name' value='"+data.Name+" /></div>";
        $('.namechangediv').append(namediv);
        var emaildiv="<div class='labeldiv'>Email</div><div
class='inputdiv'><input class='inputedit' onkeyup='validateemail()' type='text'
id='email' name='email' value='"+data.Email+" /></div>";
        $('.emailchangediv').append(emaildiv);
        var biodiv="<div class='labeldiv'>Bio</div><div
class='inputdiv'><textarea name='bio' id='bio' onkeyup='getbio()'
maxlength='120'>"+data.UserBio+"</textarea></div>";
        $('.biochangediv').append(biodiv);
    });
});
// Validate the edited email address.
function validateemail() {
    var x = $("#email").val();
    var atposition = x.indexOf("@");
    var dotposition = x.lastIndexOf(".");
    if (atposition < 1 || dotposition < atposition + 2 || dotposition + 2 >=
x.length) {
        document.getElementById("errormsg2").innerHTML = "Please enter a valid
Email Address !";
    }
    else {
        document.getElementById("errormsg2").innerHTML = "";
        email = x;
    }
}
function getname() {
    name = $("#name").val();
}
function getbio() {
    bio = $("#bio").val();
}
// Display the uploaded image.
function readURL(input) {
    if (input.files && input.files[0]) {
        var reader = new FileReader();
        var formdata = new FormData();
        var files = $('#image')[0].files;
        if(files.length > 0 ){
            formdata.append('image',files[0]);
            $.ajax({

```



```

        url: "<?php echo base_url()
?>index.php/createpost/profpic", //store the image in folder
        data: formdata,
        method: "POST",
        contentType: false,
        processData: false
    }).done(function (data) {
        var result=data.result;
        if (result=="done") {
            userimage = data.image_metadata.file_name; //get file name
            reader.onload = function (e) {
                $('#profpicid').attr('src', e.target.result);
            };
            reader.readAsDataURL(input.files[0]);
            document.getElementById("errormsg").innerHTML = '';
        } else {
            $error=data.error.slice(3,-4);
            document.getElementById("errormsg").innerHTML = $error;
        }
    });
}
}
}
// When the button is clicked to save changes.
$("#editprofile").click(function(event) {
    event.preventDefault();
    var postdata = {
        userimage: userimage,
        username: username,
        bio: bio,
        name: name,
        email: email
    }
    $.ajax({
        url: "<?php echo base_url() ?>index.php/logincon/editprofile",
        data: JSON.stringify(postdata),
        contentType: "application/json",
        method: "PUT"
    }).done(function (data) {
        var result=data.result;
        if(result=="done"){ // Redirect to my profile if the operation is
successful.
            location.href="<?php echo base_url() ?>index.php/profilecon";
        }
        else{
            document.getElementById("errormsg2").innerHTML = "Could not
able to save your changes !";
        }
    });
});

```



```

    });
</script>
</body>

<footer>
    &copy;Holiday Connect 2024 | W1809835
</footer>

</html>

```

## home.php

```

<?php defined('BASEPATH') OR exit('No direct script access allowed'); ?>
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Holiday Connect Home</title>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"
type="text/javascript"></script>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/underscore.js/1.8.3/underscore-min.js"
type="text/javascript"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/backbone.js/1.2.3/backbone-
min.js" type="text/javascript"></script>
    <script src="https://kit.fontawesome.com/b9008b61cc.js"
crossorigin="anonymous"></script>
    <link rel="stylesheet" type="text/css" href="<?php echo
base_url()?>css/home.css">
</head>

<body>
<!-- Container for feed content -->
<div class="feedcontainer">
    <!-- Container for users not followed -->
    <div class='notfollowing'>
        <!-- Heading -->
        <div class='heading'> Follow New Users to View Posts in Your Timeline
    </div>
        <!-- User listing -->
        <div class='userlisting'></div>
    </div>
</div>

<script type="text/javascript" lang="javascript">//inline javascript with backbone
framework
    var username="<?php echo $username ?>";
    $(document).ready(function () {

```



```

        event.preventDefault();
        $.ajax({// Retrieve the follow count of the specific user.
            url: "<?php echo base_url()
?>index.php/profilecon/followcount?username="+username,
            method: "GET"
        }).done(function (data) {
            if (data.following==0){// If the follow count is 0, show no posts and
suggest users to follow.
                $.ajax({
                    url: "<?php echo base_url() ?>index.php/logincon/user",
                    method: "GET"
                }).done(function (data) {
                    for (i = 0; i < 5; i++) {
                        var div ="<a href='<?php echo base_url()
?>index.php/logincon/userprofile/?username="
                            +data[i].Username+"'><div class='users'><div class="
'profilepicdiv'><img class='profilepic' src='<?php echo base_url()
?>images/profilepics/"
+data[i].UserImage+"'></div>"+data[i].Username+"</br>"+data[i].Name+"</div></a>";
                        $('<.userlisting'>).append(div);
                    }
                });
            }
            else{
                $('<.notfollowing'>).remove();
            }
        });
        postCollection.fetch();// Fetch backbone collection on startup.
    });

    var PostCollection = Backbone.Collection.extend({
        url: "<?php echo base_url()
?>index.php/home/followingposts?username="+username,
    });
    var html = "";
    var PostDisplay = Backbone.View.extend({
        el: "<.feedcontainer'",
        initialize: function () {
            this.listenTo(this.model, "add", this.showResults);
        },
        showResults: function (m) {// Show all posts in the Backbone view.
            html = html + "<div class='postdiv'><div class='locationdiv'><a
href='<?php echo base_url() ?>index.php/createpost/locations?locationid="
                + m.get('LocationId') +'><span><i class='fa-solid fa-location-
dot'></i>"
                + m.get('LocationName') + "</span></a></div><a href='<?php echo
base_url() ?>index.php/createpost/post?postId="

```



```

        + m.get('PostId') + "'><img class='postimage' src='<?php echo base_url()
?>images/userposts/"
        + m.get('PostImage') + "'/></a><div class='userlikediv'><div
class='usernameidiv'><a href='<?php echo base_url()
?>index.php/logincon/userprofile/?username="
        + m.get('Username') + "'><span>" + m.get('Username')
+ "</span></a></div><div class='likediv' id='likediv"
        + m.get('PostId') + "'><i onclick='like("+m.get('PostId')+");'
class='fa-solid fa-heart'></i></div></div><div class='captiondiv'>"
        + m.get('Caption') + "</div><div class='commentsdiv' id='commentsdiv"
        + m.get('PostId') + "'></div></div>";
        this.$el.html(html);
        // Retrieve comments for each post and render them.
        $.ajax({
            url: "<?php echo base_url()
?>index.php/home/comments?postId="+m.get('PostId'),
            method: "GET"
        }).done(function (res) {
            if(res.length!=0){
                for (i = 0; i < res.length; i++) {
                    if(i<2){
                        var div ="<span><a class='commuserlink' href='<?php
echo base_url()
?>index.php/logincon/userprofile/?username="+res[i].Username+">" +res[i].Username+"
</a>"

                        +res[i].CommentBody+"</span></br>";
                        $('#commentsdiv'+m.get('PostId')).append(div);
                    }
                }
            }
        });
        $.ajax({// Verify if the user has already liked them or not and adjust
the color accordingly.
            url: "<?php echo base_url()
?>index.php/home/checklikes?username="+username+"&postId="+m.get('PostId'),
            method: "GET"
        }).done(function (res) {
            if(res){
                document.getElementById("likediv"+m.get('PostId')).style.color
= "#FC6464";
            }
            else{
                document.getElementById("likediv"+m.get('PostId')).style.color
= "#666666";
            }
        });
    }
    });
    var postCollection = new PostCollection();

```





```

var postDisplay = new PostDisplay({model: postCollection})

// Handling click events on like buttons.
function like($postid){
    $.ajax({
        url: "<?php echo base_url() ?>index.php/home/like",
        data: JSON.stringify({username: username, postid:$postid}),
        contentType: "application/json",
        method: "POST"
    }).done(function (data) {
        $.ajax({
            url: "<?php echo base_url()
?>index.php/home/checklikes?username="+username+"&postid="+$postid,
            method: "GET"
        })
        .done(function (res) {
            if(res){
                document.getElementById("likediv"+$postid).style.color =
"#FC6464";
            }
            else{
                document.getElementById("likediv"+$postid).style.color =
"#666666";
            }
        });
    });
}
</script>
</body>

<footer>
    &copy; Holiday Connect 2024 | W1809835
</footer>

</html>

```

### location.php

```

<?php defined('BASEPATH') OR exit('No direct script access allowed'); ?>
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Holiday Connect Locations</title>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"
type="text/javascript"></script>

```



```

<script
src="https://cdnjs.cloudflare.com/ajax/libs/underscore.js/1.8.3/underscore-min.js"
type="text/javascript"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/backbone.js/1.2.3/backbone-
min.js" type="text/javascript"></script>
<script src="https://kit.fontawesome.com/b9008b61cc.js"
crossorigin="anonymous"></script>
<link rel="stylesheet" type="text/css" href="<?php echo
base_url()?>css/home.css">
</head>

<body>
<div class="locationcontainer">
<!-- Location list -->
<div class="locationlistdiv">
<!-- Display the location name -->
<span id='locationname'></span>
<!-- Location list -->
<div id="locationlist"> </div>
</div>
<!-- Post display area -->
<div class="postlocadiv"></div>
</div>

<script type="text/javascript" lang="javascript">
var locationid="<?php echo $locationid ?>";
$(document).ready(function () {
    event.preventDefault();
    $.ajax({// Get all posts from the given location ID at the start and
display the posts
        url: "<?php echo base_url()
?>index.php/createpost/location/action/id?locationid="+locationid,
        method: "GET"
    }).done(function (data) {
        document.getElementById("locationname").innerHTML = "<i class='fa-
solid fa-location-dot'></i>"+data.LocationName;
    });
    $.ajax({
        url: "<?php echo base_url()
?>index.php/createpost/location/action/all",
        method: "GET"
    })
    .done(function (data) {
        for (i = (locationid-8); i < (+locationid+8); i++) {
            if(data[i]!=null){// Display a few other locations in the list for
easier browsing
                var span ="<a href='<?php echo base_url()
?>index.php/createpost/locations?locationid="

```



```

+data[i].LocationId+"><span>"+data[i].LocationName+"</span></a></br>";
    $('#locationlist').append(span);
    }
    });
    // Define a Backbone collection to fetch posts based on location ID
    postCollection.fetch();
    });
    // Define a Backbone view to display posts
    var PostCollection = Backbone.Collection.extend({
        url: "<?php echo base_url()
?>index.php/createpost/locationposts?locationid="+locationid,
    });

    var html = "";
    var PostDisplay = Backbone.View.extend({
        el: ".postlocadiv", // Element to show the view
        initialize: function () {
            this.listenTo(this.model, "add", this.showResults);
        },
        showResults: function (m) {
            // Render each post as HTML and append to the element
            html = html + "<div class='imagelocdiv'><a href='<?php echo base_url()
?>index.php/createpost/post?postid="
            +m.get('PostId')+"><img class='locpostimage' src='<?php echo
base_url() ?>images/userposts/" +m.get('PostImage')+"'/></a></div>";
            this.$el.html(html);
        }
    });
    // Create an instance of the PostCollection
    var postCollection = new PostCollection();
    // Create an instance of the PostDisplay view with the PostCollection as its
    model
    var postDisplay = new PostDisplay({model: postCollection})

</script>
</body>

<footer>
    &copy;Holiday Connect 2024 | W1809835
</footer>

</html>

```

**myprofile.php**



```

<?php defined('BASEPATH') OR exit('No direct script access allowed'); ?>
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Holiday Connect Profile</title>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"
type="text/javascript"></script>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/underscore.js/1.8.3/underscore-min.js"
type="text/javascript"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/backbone.js/1.2.3/backbone-
min.js" type="text/javascript"></script>
    <link rel="stylesheet" type="text/css" href="<?php echo
base_url()?>css/profile.css">
</head>

<body>
<div class="profilecontainer">
    <!-- Profile details section -->
    <div class="profiledeetdiv">
        <!-- Top section containing profile picture and follow details -->
        <div class="topdiv">
            <div class="profpicdiv"></div>
            <!-- Profile picture display -->
            <div class="followdiv">
                <!-- Show followers and following count in the user profile -->
                <div class="flabel"> Following </div>
                <div class="fcount" id="followingc"></div>
                <div class="flabel"> Followers </div>
                <div class="fcount" id="followerc"></div>
            </div>
        </div>
        <!-- Username display -->
        <div class="usernameidiv"><?php echo $username ?></div>
        <!-- Full name display -->
        <div class="namediv"></div>
        <!-- Bio display -->
        <div class="bioidiv"></div>
        <!-- Edit profile button -->
        <div class="profbottomdiv">
            </a>
            <a class="editprlink" href="<?php echo
base_url()?>index.php/profilecon/editprofile">edit profile</a>
        </div>
    </div>
    <!-- Posts section -->
    <div class="postsdiv" id="postsdiv"></div>
</div>

```



```

<div class="addlinkbutton">
  <!-- Add post link -->
  <a href="<?php echo base_url()?>index.php/createpost">
    </a>
</div>

<script type="text/javascript" lang="javascript">
  var username="<?php echo $username ?>";
  // Backbone fetches the post collection on startup.
  $(document).ready(function () {
    event.preventDefault();
    postCollection.fetch();
    $.ajax({// Retrieve user details via API.
      url: "<?php echo base_url()
?>index.php/logincon/userdetails?username="+username,
      method: "GET"
    }).done(function (data) {
      var div ="<img class='profileimage' src='<?php echo base_url()
?>images/profilepics/'+data.UserImage+' />";
      $('#.profpicdiv').append(div);
      var name ="<span>"+data.Name+"</span>";
      $('#.namediv').append(name);
      var bio ="<span>"+data.UserBio+"</span>";
      $('#.biodiv').append(bio);
    });
    $.ajax({// Retrieve follower/following counts.
      url: "<?php echo base_url()
?>index.php/profilecon/followcount?username="+username,
      method: "GET"
    }).done(function (data) {
      document.getElementById("followingc").innerHTML = data.following
      document.getElementById("followerc").innerHTML = data.followers
    });
  });
  // Define Backbone model for a post
  var Post = Backbone.Model.extend({
    // URL to fetch post data from the server
    url: "<?php echo base_url() ?>index.php/profilecon/myposts"
  });
  // Define Backbone collection for posts
  var PostCollection = Backbone.Collection.extend({
    // URL to fetch post collection data from the server
    url: "<?php echo base_url() ?>index.php/profilecon/myposts",
    // Define the model associated with this collection
    model: Post,
    // Parse the fetched data
    parse: function(data) {
      return data;
    }
  }

```



```

});
// Backbone view to display the posts.
var PostDisplay = Backbone.View.extend({
  // Element to bind the view to
  el: "#postsdiv",
  // Initialize the view
  initialize: function () {
    this.listenTo(this.model, "add", this.showResults);
  },
  showResults: function () {
    var html = "";
    // Iterate over each model in the collectio
    this.model.each(function (m) {
      html = html + "<div class='postimagediv'><a href='<?php echo
base_url() ?>index.php/createpost/post?postid="
      + m.get('PostId') + "'><img class='postimage' src='<?php echo
base_url() ?>images/userposts/"
      + m.get('PostImage') + "'/></a></div>";
    });
    // Set the HTML content of the element
    this.$el.html(html);
  }
});
// Instantiate post collection and view
var postCollection = new PostCollection();
var postDisplay = new PostDisplay({model: postCollection})
</script>
</body>

<footer>
  &copy;Holiday Connect 2024 | W1809835
</footer>

</html>

```

### navigation.php

```

<?php defined('BASEPATH') OR exit('No direct script access allowed'); ?>
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Holiday Connect</title>
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"
    type="text/javascript"></script>

  <link rel="stylesheet" type="text/css" href="<?php echo
base_url()>css/navigation_page.css">

```



```

</head>
<body>
    <div class="navigationdiv">
        <!-- Logo -->
        <div class="logodiv">
            <a href="<?php echo base_url()?>index.php/home">
                </a>
            </div>
            <!-- Search bar -->
            <div class="searchdiv">
                <input type="text" class="search" id="search" placeholder="Search
for user..." onkeyup='searchusers()'/>
                
            </div>
            <!-- Navigation link -->
            <div class="linkdiv">
                <!-- Home link -->
                <div class="linkelement">
                    <a href="<?php echo base_url()?>index.php/home">
                        </a>
                    </div>
                    <!-- Notification bell icon -->
                    <div class="linkelement">
                        
                    </div>
                    <!-- Logout Link -->
                    <div class="linkelement">
                        <a href="<?php echo base_url()?>index.php/logincon/logout">
                            </a>
                        </div>
                        <div class="container">
                            
                        </div>
                    </div>
                    <!-- User profile link -->
                    <div class="profilediv">
                        <div class="userlink">
                            <a href="<?php echo base_url()?>index.php/profilecon"
class="profilelink"><span><?php echo $username ?></span></a></div>
                        </div>
                    </div>
                    <!-- Search Result and Notification -->
                    <div class="searchresults" id="searchresults"></div>

```



```

<div class="notifications" id="notifications"></div>

<script type="text/javascript" lang="javascript">
var username="<?php echo $username ?>";
function searchusers() {
    if($('#search').val().length==0){
        document.getElementById("searchresults").style.display = "none";
    }
    else{// Overlay displayed only when something is typed.
        document.getElementById("searchresults").style.display = "block";
    }
    var userdata = {
        username: "@" + $('#search').val().toLowerCase()
    };
    $.ajax({// Retrieve users based on the search string.
        url: "<?php echo base_url()
?>index.php/logincon/user/action/searchuser",
        data: JSON.stringify(userdata),
        contentType: "application/json",
        method: "POST"
    }).done(function (data) {
        $('#searchresults div').remove();
        $('#searchresults a').remove();
        if(data.length==0){// Show "No results" if the array length is 0.
            var div ="<div class ='user noresult'>No Results</div>";
            $('#searchresults').append(div);
        }
        else{//show user with their user profile
            for (i = 0; i < data.length; i++) {
                var div ="<a class='userlinks' href='<?php echo base_url()
?>index.php/logincon/userprofile/?username="
                    +data[i].Username+"><div class ='user'><div class=
'seauserimagediv'><img class='seauserimage' src='<?php echo base_url()
?>images/profilepics/"
                    +data[i].UserImage+""/></div><div
class='searuserdeet'>"+data[i].Username+"<br>"+data[i].Name+"</div></div></a>";
                $('#searchresults').append(div);
            }
        }
    });
}

// Upon clicking the notification button
function notifications(){// The overlay is only displayed when clicked.
    if(document.getElementById("notifications").style.display == "none"){
        document.getElementById("notifications").style.display = "block";
        $.ajax({// Retrieve notifications for the user.
            url: "<?php echo base_url()
?>index.php/profilecon/notifications",
            data: JSON.stringify({username: username}),

```





```

        contentType: "application/json",
        method: "GET"
    }).done(function (data) {
        $('#notifications div').remove();
        $('#notifications a').remove();
        if(data.length==0){// No notifications available.
            var div ="<div class ='>No Notifications</div>";
            $('#notifications').append(div);
        }
        else{
            for (i = 0; i < data.length; i++) {
                // If there is a comment body, it indicates a comment
notification.
                if (data[i].CommentBody!=null && data[i].PostId!=null){
                    var div ="<a href='<?php echo base_url()
?>index.php/createpost/post?postid="
                    +data[i].PostId+"><div>"+data[i].Username+"
"+data[i].Notification+"</br><span class='commentspam'>"+
                    +data[i].CommentBody+"</span></div></a>";
                    $('#notifications').append(div);
                }// If there is a post ID but no comment, it signifies a
post like.
                else if(data[i].CommentBody==null &&
data[i].PostId!=null){
                    var div ="<a href='<?php echo base_url()
?>index.php/createpost/post?postid="
                    +data[i].PostId+"><div>"+data[i].Username+"
"+data[i].Notification+"</div></a>";
                    $('#notifications').append(div);
                }// Otherwise, it indicates a follow notification.
                else if(data[i].CommentBody==null &&
data[i].PostId==null){
                    var div ="<a href='<?php echo base_url()
?>index.php/logincon/userprofile/?username="
                    +data[i].Username+"><div>"+data[i].Username+"
"+data[i].Notification+"</div></a>";
                    $('#notifications').append(div);
                }
            }
        }
    });
}
else{
    document.getElementById("notifications").style.display = "none";
}
}
</script>

<script>

```



```

const themeToggle = document.getElementById('theme-toggle');
const body = document.body;

themeToggle.addEventListener('click', () => {
    body.classList.toggle('dark-mode');
    const currentSrc = themeToggle.getAttribute('src');
    const newSrc = currentSrc.includes('sun') ? '<?php echo base_url()
?>images/moon.png' : '<?php echo base_url() ?>images/sun.png';
    themeToggle.setAttribute('src', newSrc);
});
</script>
</body>

<footer>
    &copy;Holiday Connect 2024 | W1809835
</footer>

</html>

```

### post.php

```

<?php defined('BASEPATH') OR exit('No direct script access allowed'); ?>
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Holiday Connect Post</title>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"
type="text/javascript"></script>
    <script src="https://kit.fontawesome.com/b9008b61cc.js"
crossorigin="anonymous"></script>
    <link rel="stylesheet" type="text/css" href="<?php echo
base_url()?>css/viewpost.css">
</head>
<body>
    <!-- Post container -->
    <div class='postcontainer'>
        <!-- Left section -->
        <div class='leftdiv'>
            <!-- Post image -->
            <div class='postimagediv'></div>
            <!-- Location and likes -->
            <div class='locationlikediv'>
                <!-- Like button -->
                <div class='likediv' id='likediv'></div>
                <!-- Like count -->
                <div class='likecount'></div>
            <!-- Location -->

```



```

        <div class='locationdiv'></div>
    </div>
</div>
<!-- Right section -->
<div class='rightdiv'>
    <!-- User details -->
    <div class='usernameimgdiv'></div>
    <!-- Post caption -->
    <div class='captiondiv'></div>
    <!-- Comment area -->
    <div class='commentareadiv'>
        <textarea onkeyup='checkinputs();' name='comment' id='comment'
maxlength='50'></textarea>
        <button onclick='postcomment();' id='commentbtn'
disabled='disabled'>Comment</button>
    </div>
    <!-- Comments -->
    <div class='commentsdiv'></div>
</div>
</div>

<script type='text/javascript' lang='javascript'>
```



```

        var div4 = "<i onclick='like();' class='fa-solid fa-heart'></i>";
        $('.likediv').append(div4);
        var div5 = data.Caption ;
        $('.captiondiv').append(div5);
    });
    $.ajax({ // Verify if the user has already liked the post.
        url: "<?php echo base_url()
?>index.php/home/checklikes?username="+username+"&postid="+postid,
        method: "GET"
    }).done(function (res) {
        if(res){
            document.getElementById("likediv").style.color = "#FC6464";
        }
        else{
            document.getElementById("likediv").style.color = "#666666";
        }
    });
});
// Disable the comment button unless there is a value in the input.
function checkinputs() {
    if ($("#comment").val() != "") {
        document.getElementById('commentbtn').disabled = false;
    }
    else{
        document.getElementById('commentbtn').disabled = true;
    }
}
// Upon liking the post, send a POST request and change the color of the
button.
function like(){
    $.ajax({
        url: "<?php echo base_url() ?>index.php/home/like",
        data: JSON.stringify({username: username, postid: postid}),
        contentType: "application/json",
        method: "POST"
    }).done(function (data) {
        $.ajax({
            url: "<?php echo base_url()
?>index.php/home/checklikes?username="+username+"&postid="+postid,
            method: "GET"
        })
        .done(function (res) {
            if(res){
                document.getElementById("likediv").style.color = "#FC6464";
                likecount();
            }
            else{
                document.getElementById("likediv").style.color = "#666666";
                likecount();
            }
        });
    });
}

```



```

    }
    });
});
}
// Upon clicking the comment button, add a comment to the database.
function postcomment(){
    var comment = {
        postid: postid,
        comment:$("#comment").val()
    };
    $.ajax({
        url: "<?php echo base_url() ?>index.php/home/comments",
        data: JSON.stringify(comment),
        contentType: "application/json",
        method: "POST"
    }).done(function (data) {
        if (data) {
            document.getElementById('comment').value = '';
            getComments();
        }
    });
}
// Retrieve all comments in the post.
function getComments(){
    $.ajax({
        url: "<?php echo base_url()
?>index.php/home/comments?postid="+postid,
        method: "GET"
    }).done(function (res) {
        if(res.length!=0){
            $('#commentsdiv div').remove();
            for (i = 0; i < res.length; i++) {
                var div ="<div class='comments'><a href='<?php echo base_url()
?>index.php/logincon/userprofile/?username="+res[i].Username+"'>"+res[i].Username+"
</a>"

                +res[i].CommentBody+"</div>";
                $('#commentsdiv').append(div);
            }
        }
    });
}
// Retrieve the like count of the post.
function likecount(){
    $.ajax({
        url: "<?php echo base_url()
?>index.php/createpost/likecount?postid="+postid,
        method: "GET"
    }).done(function (res) {
        $('#likecount span').remove();
    });
}

```



```

        if(res==1){
            var div ="<span>"+res+" like</span>";
        }
        else{
            var div ="<span>"+res+" likes</span>";
        }
        $('#likecount').append(div);
    });
}
</script>
</body>

<footer>
    &copy;Holiday Connect 2024 | W1809835
</footer>

</html>

```

### userprofile.php

```

<?php defined('BASEPATH') OR exit('No direct script access allowed'); ?>
<!DOCTYPE html>
<html lang="en">
<head>
<title>Holiday Connect User Account</title>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"
type="text/javascript"></script>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/underscore.js/1.8.3/underscore-min.js"
type="text/javascript"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/backbone.js/1.2.3/backbone-
min.js" type="text/javascript"></script>
    <link rel="stylesheet" type="text/css" href="<?php echo
base_url()?>css/profile.css">
</head>

<body>
<div class="profilecontainer">
<!-- Profile details section -->
<div class="profiledeetdiv">
    <!-- Top section containing profile picture and follow details -->
    <div class="topdiv">
        <!-- Profile picture display -->
        <div class="profpicdiv"></div>
        <!-- Follow details -->
        <div class="followdiv">
            <!-- Following count -->

```



```

        <div class="flabel">Following</div>
        <div class="fcount" id="followingc"></div>
        <!-- Followers count -->
        <div class="flabel">Followers</div>
        <div class="fcount" id="followerc"></div>
    </div>
</div>
<!-- Username display -->
<div class="usernameid" id="nametag"><?php echo $username ?></div>
<!-- Full name display -->
<div class="namediv"></div>
<!-- Bio display -->
<div class="biodiv"></div>
<!-- Follow button -->
<div class="profbottomdiv">
    <!-- Follow button -->
    <a href="#" onclick='follow();'><div id="followbutton"> </div></a>
</div>
</div>
<!-- Posts section -->
<div class="postsdiv" id="postsdiv"></div>
</div>

<script type="text/javascript" lang="javascript">
    var username="<?php echo $username ?>";
    $(document).ready(function () {
        event.preventDefault();
        postCollection.fetch();// Fetch posts from the collection upon startup.
        checkfollow();// Verify if the current profile follows this user.
        $.ajax({// Retrieve user details and show them.
            url: "<?php echo base_url()
?>index.php/logincon/userdetails?username="+username,
            method: "GET"
        }).done(function (data) {
            var div ="<img class='profileimage' src='<?php echo base_url()
?>images/profilepics/"+data.UserImage+"'/>";
            $('.profpicdiv').append(div);
            var name ="<span>"+data.Name+"</span>";
            $('.namediv').append(name);
            var bio ="<span>"+data.UserBio+"</span>";
            $('.biodiv').append(bio);
        });
        followcount();
    });
    // Backbone model for a post
    var Post = Backbone.Model.extend({
        // URL to fetch user posts from the server
        url: "<?php echo base_url()
?>index.php/createpost/userposts?username="+username

```



```

});
// Backbone collection for user posts
var PostCollection = Backbone.Collection.extend({
  // URL to fetch user posts collection from the server
  url: "<?php echo base_url()
?>index.php/createpost/userposts?username="+username,
  // Specify the model associated with this collection
  model: Post,
  // Parse the fetched data
  parse: function(data) {
    return data;
  }
});
// Backbone view to display user posts
var PostDisplay = Backbone.View.extend({
  // Root element to bind the view
  el: "#postsdiv",
  // Initialize the view
  initialize: function () {
    // Listen to 'add' event on the model
    this.listenTo(this.model, "add", this.showResults);
  },
  // Show the posts
  showResults: function () {
    // Iterate over each model in the collection
    var html = "";
    // Generate HTML for each post
    this.model.each(function (m) {
      html = html + "<div class='postimagediv'><a href='<?php echo
base_url() ?>index.php/createpost/post?postId="
      + m.get('PostId') + "'><img class='postimage' src='<?php echo
base_url() ?>images/userposts/"
      + m.get('PostImage') + "'/></a></div>";
    });
    // Set the HTML content of the root element
    this.$el.html(html);
  }
});
var postCollection = new PostCollection();
var postDisplay = new PostDisplay({model: postCollection})
function followcount(){
  $.ajax({// Retrieve follower/following count and show it.
    url: "<?php echo base_url()
?>index.php/profilecon/followcount?username="+username,
    method: "GET"
  }).done(function (data) {
    document.getElementById("followingc").innerHTML = data.following
    document.getElementById("followerc").innerHTML = data.followers
  });
}

```





```

    }
    function follow(){// Handling click events on the follow button.
        $.ajax({
            url: "<?php echo base_url() ?>index.php/profilecon/follow",
            data: JSON.stringify({isfollowing: username}),
            contentType: "application/json",
            method: "POST"
        }).done(function () {
            checkfollow();
            followcount();
        });
    }
    function checkfollow(){// Verify if the user has already followed and update
the button accordingly.
        $.ajax({
            url: "<?php echo base_url()
?>index.php/profilecon/checkfollow?isfollowing="+username,
            method: "GET"
        }).done(function (data) {
            if (data) {
                document.getElementById("followbutton").innerHTML = "Unfollow"
            }
            else {
                document.getElementById("followbutton").innerHTML = "Follow"
            }
        });
    }
</script>
</body>

<footer>
    &copy;Holiday Connect 2024 | W1809835
</footer>

</html>

```



## 2.4 CSS Files

### Login.css

```
body {
    font-family: "DM Sans", sans-serif;
    line-height: 1.5;
    background-color: #b4dae0;
}
input:-webkit-autofill,
input:-webkit-autofill:hover,
input:-webkit-autofill:focus,
input:-webkit-autofill:active{
    -webkit-box-shadow: 0 0 30px white inset !important;
}
.logodiv {
    margin:3% auto 0 auto;
    height:fit-content;
    width:fit-content;
}
.logoimage {
    width: 350px;
    height: auto;
}

.signform, .loginform, .pwform {
    display: flex;
    flex-direction: column;
    width: 100%;
    max-width: 425px;
    background-color: #fff;
    border-radius: 40px;
    padding: 1.5%;
}
.signform{
    margin: 1% auto;
    box-shadow: 0 0 10px 4px rgba(247, 153, 206, 0.25);
}
.loginform{
    margin: 1% auto;
    box-shadow: 0 0 10px 4px rgba(199, 182, 237, 0.25);
}
.pwform{
    margin:8% auto;
    box-shadow: 0 0 10px 4px rgba(199, 182, 237, 0.25);
}

.authforms {
    padding: 0 3%;
}
```



```

}

.input {
  display: flex;
  flex-direction: column-reverse;
  position: relative;
  padding-top: 1.5rem;
}
.input + .input {
  margin-top: 1.5rem;
}

.signlabel, .loginlabel {
  color: #BABABA;
  position: absolute;
  top: 1.5rem;
  transition: 0.25s ease;
}

.signfield, .loginfield {
  border: 0;
  z-index: 1;
  background-color: transparent;
  font: inherit;
  font-size: 1.125rem;
  padding: 0.25rem 0;
}
.signfield{
  border-bottom: 2px solid #B6E3EC;
}
.loginfield{
  border-bottom: 2px solid #B6EDDA;
}
.signfield:focus, .signfield:valid, .signfield:hover {
  outline: 0;
  border-bottom-color: #95C2E5;
}
.loginfield:focus, .loginfield:valid, .loginfield:hover {
  outline: 0;
  border-bottom-color: #7EC4CF;
}

.signfield:focus + .signlabel, .signfield:valid + .signlabel, .signfield:hover +
.signlabel{
  color: #058ef6;
  transform: translateY(-1.5rem);
}
.loginfield:focus + .loginlabel, .loginfield:valid + .loginlabel, .loginfield:hover
+ .loginlabel {

```



```
    color: #3db649;
    transform: translateY(-1.5rem);
}

.action {
    width: 50%;
    margin: 2rem auto 0 auto;
}

.signupbtn, .loginbtn{
    font: inherit;
    font-size: 1.25rem;
    padding: 0.5rem;
    width: 100%;
    font-weight: 500;
    border-radius: 25px;
    color: #fff;
    border: 0;
    transition: transform .4s;
}

.signupbtn:enabled:hover, .loginbtn:enabled:hover{
    transform: scale(1.05);
}

.signupbtn:disabled{
    opacity: 0.4;
    background-color: #f27b7b;
}

.loginbtn:disabled{
    opacity: 0.4;
    background-color: #a178f9;
}

.signupbtn:enabled{
    background-color: #ed91c5;
}

.loginbtn:enabled {
    background-color: #a076fb;
}

.signspandiv, .loginspandiv {
    padding: 1rem;
    text-align: center;
    color: #0e0e0e;
}

.signspandiv a {
    color: #018df8;
    text-decoration: none;
}
```



```
.signspandiv a:hover {
  color: #0490fb;
  text-decoration:underline;
}
.loginspandiv a {
  color: #03d7f8;
  text-decoration: none;
}
.loginspandiv a:hover {
  color: #02d6f7;
  text-decoration:underline;
}
.logheading, .pwresetheading{
  font-size: 2.5rem;
  font-weight: 550;
  letter-spacing: 1px;
  color: #8249fb;
  text-align: center;
}
.errormsg{
  color: #f80404;
}

footer {
  /* background-color: #a36dd3;
  color: rgb(1, 1, 1);
  padding: 10px;
  text-align: center; */
  position: fixed;
  left: 0;
  bottom: 0;
  width: 100%;
  background-color: #8f85eb;
  color: rgb(1, 1, 1);
  padding: 10px;
  text-align: center;
  text-decoration: solid;
}
```

### navigation page.css

```
body {
  font-family: "DM Sans", sans-serif;
  line-height: 1.5;
  background-color: #ffffff;
  margin: 0px;
  background-color: var(--background-color);
  color: var(--text-color);
}
```



```
.navigationdiv{
    max-height: 85px;
    height: 85px;
    margin: 0 0 20px 0;
    position: sticky;
    top: 0px;
    box-shadow: 0px 5px 5px -3px rgba(82, 178, 207, 0.3) ;
    background-color: #ffffff;
    z-index: 2;
}
.logodiv{
    float: left;
    height: 100%;
    position: relative;
    margin: 0 0 0 2%;
    width: 18%;
}
.logoimage{
    height: auto;
    width: 100%;
    position: absolute;
    top: 50%;
    transform: translateY(-50%)
}
.searchdiv{
    float: left;
    width: 25%;
    height: 100%;
    position: relative;
    margin: 0 0 0 23%;
}
.search{
    width: 95%;
    position: absolute;
    top: 50%;
    transform: translateY(-50%);
    padding: 3% 0 3% 3%;
    border-radius: 50px;
    font-size: 1rem;
    border: 2px solid #000000;
    color: #000000;
}
.search:focus, .search:hover{
    outline: none !important;
    border: 2px solid #04c2fc;
}
.linkdiv{
    float: left;
    height: 100%;
```



```
    position: relative;
    width: 15%;
    margin: 0 0 0 2%;
}
.linkelement{
    height: 100%;
    width: 30%;
    float: left;
}
.homeimage{
    height: 40%;
    width: auto;
    position: absolute;
    top: 50%;
    transform: translateY(-50%);
    width: auto;
}
.bellimage{
    height: 50%;
    width: auto;
    position: absolute;
    top: 50%;
    transform: translateY(-50%);
    width: auto;
}
.profilediv{
    float: right;
    height: 100%;
    width: 13%;
    max-width: 13%;
    margin: 0 2% 0 0;
    position: relative;
}
.userlink{
    position: absolute;
    top: 50%;
    transform: translateY(-50%);
    font-size: 1.25rem;
    height: auto;
    width: 100%;
    text-align: right;
}
.profilelink{
    color: #05a7d9;
    text-decoration: none;
}
.profilelink:hover{
    color: #019ccb;
    text-decoration: underline;
}
```



```
}
.searchresults, .notifications{
    position: fixed;
    display: none;
    height: fit-content;
    top: 70px;
    right: 0;
    bottom: 0;
    background-color: white;
    box-shadow: 0 0 10px 4px rgba(21, 176, 200, 0.25);
    z-index: 2;
    border-radius: 10px;
}
.searchresults{
    width: 30%;
    max-width: 30%;
    max-height: 400px;
    left: 40%;
}
.notifications{
    width: 30%;
    max-width: 30%;
    max-height: 500px;
    left: 60%;
}
.user{
    height: 56px;
    padding: 0 2%;
    box-shadow: 0px 0px 4px 0px rgba(7, 160, 207, 0.25)
}
.userlinks{
    color: #000000;
    text-decoration: none;
}
.seauserimagediv{
    padding: 3px 0 0 2px;
    width: 50px;
    height: 50px;
    float: left;
}
.seauserimage{
    height: 50px;
    width: 50px;
    border-radius: 50%;
    object-fit: cover;
}
.searuserdeet{
    padding: 3px 0 0 65px;
    box-sizing: border-box;
```





```
}
.noresult{
    text-align: center;
    color: #040404;
    padding-top: 15px;
    box-sizing: border-box;
}
.notifications div{
    width: 100%;
    box-sizing: border-box;
    padding: 10px 0 10px 5%;
    height: fit-content;
    box-shadow: 0px 0px 4px 0px rgba(82, 178, 207, 0.25)
}
.notifications a{
    color: #000000;
    text-decoration: none;
}
.notifications a:hover{
    color: #00bffa;
}
.commentspam{
    font-size: 14px;
    color: #000000;
    padding-left: 4px;
    font-weight: 600;
}

.logoutlink{
    color: #EB9494;
    margin-left: 80px;
    font-size: 16px;
    font-weight: 550;
    letter-spacing: 1px;
    text-decoration: underline;
    height: 35%;
    width: auto;
    position: absolute;
    top: 50%;
    left: 70%;
    transform: translateY(-50%);
    width: auto;
}

.logoutimage{
    height: 40%;
    width: auto;
    position: absolute;
    top: 50%;
```



```
    left: 65%;
    transform: translateY(-50%);
    width: auto;
}

footer {
    /* background-color: #a36dd3;
    color: rgb(1, 1, 1);
    padding: 10px;
    text-align: center; */
    position: fixed;
    left: 0;
    bottom: 0;
    width: 100%;
    background-color: #8f85eb;
    color: rgb(1, 1, 1);
    padding: 10px;
    text-align: center;
    text-decoration: solid;
}

.searchicon{
    height: 40%;
    width: auto;
    position: absolute;
    top: 50%;
    left: 85%;
    transform: translateY(-50%);
    width: auto;
}

:root {
    --background-color: #ffffff;
    --text-color: #000000;
}

.dark-mode {
    --background-color: #333333;
    --text-color: #ffffff;
}

.container {
    padding: 20px;
}

button {
    padding: 10px 20px;
    background-color: #007bff;
    color: #ffffff;
```



```

    border: none;
    cursor: pointer;
}

.lightimage{
    height: 50%;
    width: auto;
    position: absolute;
    top: 50%;
    right: 300%;
    transform: translateY(-50%);
    width: auto;
}

```

### home.css

```

.feedcontainer{
    width: 100%;
    display: block;
    overflow: auto;
}

.postdiv{
    width: 30%;
    aspect-ratio : 1 / 1.5;
    margin-left: 2.5%;
    margin-bottom: 50px;
    background-color: white;
    float: left;
    border-radius: 30px; /* Add curved margin */
    position: relative;
}

.postdiv::before {
    content: '';
    position: absolute;
    top: -10px;
    left: -10px;
    width: calc(100% + 20px);
    height: calc(100% + 20px);
    background: linear-gradient(45deg, #fff, #725eda, #33edce, #fff);
    background-size: 300% 300%;
    border-radius: 20px; /* Match the border-radius of the postdiv */
    z-index: -1;
    animation: glitter 1.5s ease infinite;
}

@keyframes glitter {
    0% {

```



```
        background-position: 0% 0%;
    }
    50% {
        background-position: 100% 100%;
    }
    100% {
        background-position: 0% 0%;
    }
}

.postimage{
    width: 90%;
    aspect-ratio : 1 / 1;
    border-radius: 10px;
    object-fit: cover;
    margin: 0 auto;
    display: block;
}

.locationdiv{
    margin: 1% 5%;
    font-size: 16px;
    letter-spacing: 0.5px;
}

}

.locationdiv a{
    color: #000000;
    text-decoration: none;
    transition: color .4s;
}

.locationdiv a:hover{
    color: #05a0cf;
    text-decoration: none;
}

.fa-solid, .commuserlink{
    margin-right: 3%;
}

.userlikediv{
    width: 100%;
    margin: 0 5%;
    height: 30px;
}

}

.usernamediv{
    width: 75%;
    float: left;
    font-size: 18px;
    letter-spacing: 0.5px;
    margin-top: 5px;
}

}

.usernamediv a{
```



```
        color: #04a0d0;
        text-decoration: none;
        transition: color .4s;
    }
    .username div a: hover{
        color: #00a0d1;
        text-decoration: none;
    }
    .likediv{
        width: 15%;
        float: left;
        text-align: right;
        font-size: 28px;
        font-weight: 900;
    }
    .commentsdiv{
        margin: 0 5%;
        font-size: 13px;
        font-weight: 600;
        color: #040404;
    }
    .commentsdiv a{
        color: #9e3a88;
        text-decoration: none;
    }
    .captiondiv{
        margin: 0 5% 5px 5%;
        font-size: 14px;
        text-align: justify;
        color: #030303;
    }
    .fa-heart{
        cursor: pointer;
        transition: transform .4s;
    }
    .fa-heart: hover{
        transform: scale(1.05);
    }
    .locationcontainer{
        width: 100%;
        display: block;
        overflow: auto;
    }
    .locationlistdiv{
        width: 20%;
        float: left;
        position: fixed;
        display: flex;
        flex-flow: column;
```



```
}
.postlocadiv{
    box-shadow: 0px 0px 10px 0px rgba(44, 70, 175, 0.25) ;
    width: 80%;
    float: right;
}
.imagelocdiv{
    width: 28%;
    aspect-ratio : 1 / 1;
    margin-left: 4%;
    float: left;
}
.locpostimage{
    width: 96%;
    aspect-ratio : 1 / 1;
    margin-top: 20px;
    border-radius: 10px;
    object-fit: cover;
    transition: transform .4s;
}
.locpostimage:hover{
    transform: scale(1.03);
}
#locationname{
    font-size: 24px;
    font-weight: 700;
    letter-spacing: 1px;
    margin: 20px 0 0 10%;
    color: #02a9dc;
}
#locationlist{
    font-size: 16px;
    letter-spacing: 1px;
    margin: 20px 0 0 10%;
}
#locationlist a{
    color: #030303;
    text-decoration: none;
}
#locationlist a:hover{
    color: #059ecd;
    text-decoration: none;
}
.notfollowing{
    width: 60%;
    margin: auto;
    display: block;
    overflow: hidden;
}
```



```
.heading{
    width: 100%;
    text-align: center;
    font-size: 30px;
    padding: 10px 0;
    letter-spacing: 1px;
    color: #030303;
    font-weight: 600;
}
.userlisting{
    width: 45%;
    margin: auto;
    height: fit-content;
}
.users{
    width: 100%;
    float: left;
    padding: 8px 0;
    box-sizing: border-box;
    box-shadow: 0px 0px 4px 0px rgba(82, 178, 207, 0.25)
}
.profilepicdiv{
    width: 50px;
    height: 50px;
    float: left;
    padding: 0 8px;
}
.profilepic{
    height: 50px;
    width: 50px;
    border-radius: 50%;
    object-fit: cover;
}
.userlisting a{
    color: #050505;
    text-decoration: none;
}
.userlisting a:hover{
    color: #06a4d5;
}

footer {
    /* background-color: #a36dd3;
    color: rgb(1, 1, 1);
    padding: 10px;
    text-align: center; */
    position: fixed;
    left: 0;
    bottom: 0;
```



```
width: 100%;
background-color: #8f85eb;
color: rgb(1, 1, 1);
padding: 10px;
text-align: center;
text-decoration: solid;
}
```

### profile.css

```
body {
  background-color: #ffffff;
  background-color: var(--background-color);
  color: var(--text-color);
}
.profilecontainer{
  width: 100%;
  display: block;
  overflow: auto;
}
.profiledeetdiv{
  width: 34%;
  max-width: 34%;
  float: left;
  margin-left: 2%;
  padding: 2%;
  position: fixed;
  box-shadow: 0px 0px 10px 0px rgba(152, 162, 203, 0.25) ;
  border-radius: 10px;
  display: flex;
  flex-flow: column;
  height: 73%;
}
.postsdiv{
  width: 56%;
  float: right;
  max-width: 60%;
  display: block;
  overflow: auto;
  margin-left: 4%;
}
.postimagediv{
  width: 30%;
  aspect-ratio : 1 / 1;
  float: left;
  margin: 2% 2% 2% 1%;
}
.postimage{
```





```
    width: 96%;
    aspect-ratio : 1 / 1;
    border-radius: 10px;
    object-fit: cover;
    transition: transform .4s;
}
.postimage:hover{
    transform: scale(1.05);
}
.topdiv{
    width: 100%;
}
.profpicdiv{
    width: 62%;
    height: 250px;
    float: left;
    align-items: center;
}
.profileimage{
    width: 86%;
    aspect-ratio : 1 / 1;
    object-fit: cover;
    border-radius: 50%;
}
.followdiv{
    float: left;
    height: 250px;
    width: 38%;
    text-align: center;
}
.flabel{
    margin-top: 50px;
    font-weight: 600;
    font-size: 18px;
    letter-spacing: 1px;
    color: #666666;
}
.fcount{
    font-weight: 600;
    font-size: 28px;
    letter-spacing: 1px;
    color: #90CDD6;
}
.usernameidiv{
    font-weight: 500;
    font-size: 20px;
    letter-spacing: 1px;
    color: #8EA0E3;
    margin-top: 20px;
```



```
}
.namediv{
    font-size: 20px;
    font-weight: 600;
    letter-spacing: 4px;
    color: #666666;
    right: 55%;
}
.biodiv{
    font-size: 16px;
    letter-spacing: 1px;
    line-height: normal;
    color: #666666;
    margin: 5px 0 40px 0;
}
.profbottomdiv{
    width: 100%;
}
a{
    text-decoration: none;
}
#followbutton{
    padding: 10px;
    background-color: #9270b6;
    width: 40%;
    letter-spacing: 1px;
    margin: auto;
    font-weight: 500;
    font-size: 20px;
    text-align: center;
    color: #ffffff;
    border-radius: 20px;
    transition: transform .4s;
}
#followbutton:hover{
    transform: scale(1.05);
}
.logoutlink, .editprlink{
    font-size: 16px;
    font-weight: 550;
    letter-spacing: 1px;
    text-decoration: underline;
}

.editprlink{
    color: #7EC4CF;
}

.editimage{
```



```
    height: 5%;
    width: 5%;
    position: absolute;
    top: 77%;
    transform: translateY(-50%);
    width: auto;
}

footer {
    /* background-color: #a36dd3;
    color: rgb(1, 1, 1);
    padding: 10px;
    text-align: center; */
    position: fixed;
    left: 0;
    bottom: 0;
    width: 100%;
    background-color: #8f85eb;
    color: rgb(1, 1, 1);
    padding: 10px;
    text-align: center;
    text-decoration: solid;
}

#nametag{
    font-size: 20px;
}

.addlinkbutton{
    height: 15%;
    width: 20%;
    position: fixed;
    right: 35%;
    bottom: 0;
    margin: 0%;
    transform: translateX(60%);
}

.linkimage{
    height: 50%;
    width: 20%;
    position: fixed;
    right: 20%;
    bottom: 50;
    margin: 0%;
    transform: translateX(50%);
}
```



### viewpost.css

```
.postcontainer{
    width: 100%;
    display: block;
    overflow: auto;
}
.leftdiv{
    margin-top: 10px;
    width: 55%;
    max-width: 55%;
    float: left;
    position: fixed;
}
.rightdiv{
    width: 45%;
    float: right;
    max-width: 45%;
    display: block;
    overflow: auto;
}
.postimagediv, .locationlikediv{
    margin: auto;
    display: block;
    overflow: auto;
}
.postimagediv{
    width: 60%;
}
.locationlikediv{
    width: 58%;
}
.postimage{
    width: 100%;
    aspect-ratio : 1 / 1;
    border-radius: 10px;
    object-fit: cover;
}
.locationdiv{
    width: 60%;
    float: right;
    box-sizing: border-box;
    text-align: right;
    font-size: 25px;
    font-weight: 600;
    letter-spacing: 1px;
    padding-top: 4px;
}
.likediv{
```



```
    width: fit-content;
    float: left;
    font-size: 30px;
}
.likecount{
    width: 20%;
    float: left;
    font-size: 17px;
    font-weight: 550;
    margin: 8px 0 0 10px;
    color: #666666;
}
.fa-location-dot{
    margin-right: 3%;
}
.locationdiv a{
    color: #666666;
    text-decoration: none;
    transition: color .4s;
}
.locationdiv a:hover{
    color: #52B2CF;
    text-decoration: none;
}
.fa-heart{
    padding: 0;
    cursor: pointer;
    transition: transform .4s;
}
.fa-heart:hover{
    transform: scale(1.03);
}

.usernameimgdiv{
    width: 100%;
    display: block;
    overflow: auto;
}
.userimagediv{
    width: 12%;
    aspect-ratio : 1 / 1;
    float: left;
}
.userimage{
    width: 100%;
    aspect-ratio : 1 / 1;
    border-radius: 50%;
    object-fit: cover;
```



```
}
.usernamediv{
    width: 50%;
    float: left;
    padding-left: 3%;
    font-size: 24px;
    letter-spacing: 0.5px;
    margin-top: 1%;
}
.usernamediv a{
    color: #52B2CF;
    text-decoration: none;
    transition: color .4s;
}
.usernamediv a:hover{
    color: #32AFD5;
    text-decoration: none;
}
.captiondiv{
    width: 90%;
    font-size: 18px;
    color: #666666;
}
.commentareadiv{
    margin: 15px 0;
    width: 95%;
    display: block;
    overflow: auto;
}
.commentsdiv{
    width: 90%;
    font-size: 15px;
    color: #666666;
}
.commentsdiv a{
    color: #9A618E;
    text-decoration: none;
    padding-right: 2%;
}
.commentsdiv a:hover{
    color: #9A618E;
    text-decoration: underline;
}
.comments{
    padding: 15px 0 12px 0;
    border-bottom: 1px solid #52B2CF;
}
textarea{
    width: 60%;
```



```
height: 50px;
padding: 8px;
border: 2px solid #cccccc;
border-radius: 8px;
font-size: 16px;
letter-spacing: 1px;
line-height: normal;
color: #666666;
resize: none;
float: left;
outline: none;
font-family: "DM Sans", sans-serif;
}
textarea:focus{
    border: 2px solid #8FD6C2;
}
button{
    height: 40px;
    float: left;
    width: 25%;
    border-radius: 20px;
    margin-left: 2%;
    margin-top: 15px;
    border: none;
    font-size: 16px;
    transition: background-color .4s;
    cursor: pointer;
}
button:disabled{
    background-color: #cccccc;
    color: #7C7D7C;
}
button:enabled{
    background-color: #52B2CF;
    color: white;
}
footer {
    /* background-color: #a36dd3;
    color: rgb(1, 1, 1);
    padding: 10px;
    text-align: center; */
    position: fixed;
    left: 0;
    bottom: 0;
    width: 100%;
    background-color: #8f85eb;
    color: rgb(1, 1, 1);
    padding: 10px;
    text-align: center;
```



```
    text-decoration: solid;
}
```

### **edituserprofile.css**

```
.editprofcontainer{
    width: 100%;
    display: block;
    overflow: auto;
}
.texteditdiv{
    width: 50%;
    float: right;
    display: block;
    overflow: auto;
    padding-top: 20px;
}
.piceditdiv{
    width: 50%;
    float: left;
    display: block;
    overflow: auto;
    position: relative;
}
.editprofdiv{
    width: 90%;
    float: left;
}
#editprofile{
    cursor: pointer;
    height: 50px;
    text-align: center;
    background-color: #1bb9d2;
    width: 22%;
    margin: 10px 0 0 17%;
    padding-top: 10px;
    color: white;
    font-size: 20px;
    letter-spacing: 1px;
    box-sizing: border-box;
    border-radius: 20px;
    transform: background-color 4s;
}
#editprofile:hover{
    background-color: #8FD6C2;
}
.profigamediv{
    width: 100%;
```





```
}
#profpicid{
    width: 50%;
    aspect-ratio : 1 / 1;
    border-radius: 10px;
    object-fit: cover;
    border-radius: 50%;
    margin: auto;
    display: block;
}
input[type=file] {
    margin: 20px 0 0 25%;
    width: 50%;
    height: 50px;
    opacity: 0%;
    cursor: pointer;
    position: absolute;
}
.dummy {
    width: 50%;
    margin: 20px 0 0 25%;
    background-color: #744fc4;
    text-align: center;
    border-radius: 20px;
    height: 50px;
    padding-top: 8px;
    box-sizing: border-box;
    font-size: 20px;
    letter-spacing: 1px;
    color: #fbfafd;
    transform:background-color 4s;
}
.dummy:hover{
    background-color: #585eb1;
}
.errormsg{
    color: #EB9494;
    width: 100%;
    text-align: center;
    font-size: 18px;
    padding-bottom: 10px;
}
.namechangediv, .emailchangediv, .biochangediv{
    width: 100%;
    margin-bottom: 20px;
    display: block;
    overflow: auto;
}
.labeldiv{
```



```
    font-weight: 600;
    font-size: 24px;
    letter-spacing: 1px;
    color: #007cdb;
    width: 18%;
    margin-left: 20%;
    float: left;
}
.inputdiv{
    font-size: 16px;
    color: #666666;
    width: 62%;
    float: right;
}

.inputedit{
    width: 90%;
    height: 30px;
    outline: none;
    border: none;
    font-size: 16px;
    letter-spacing: 1px;
    line-height: normal;
    border-bottom: 2px solid #C7B6ED;
    color: #666666;
}
.inputedit:focus{
    border-bottom: 2px solid #A57DFF;
}
textarea{
    width: 80%;
    height: 100px;
    padding: 12px 20px;
    border: 2px solid #C7B6ED;
    border-radius: 8px;
    font-size: 16px;
    letter-spacing: 1px;
    line-height: normal;
    color: #666666;
    resize: none;
    outline: none;
    font-family: "DM Sans", sans-serif;
}
textarea:focus{
    border: 2px solid #A57DFF;
}
.changepwddiv{
    padding-top: 10px;
    margin-left: 20%;
```



```
    font-size: 20px;
    letter-spacing: 0.5px;
    line-height: normal;
    color: #666666;
}

.changepwddiv a{
    color: #1a029e;
}

.changepwddiv a:hover{
    color: #8b75d6;
}

footer {
    /* background-color: #a36dd3;
    color: rgb(1, 1, 1);
    padding: 10px;
    text-align: center; */
    position: fixed;
    left: 0;
    bottom: 0;
    width: 100%;
    background-color: #8f85eb;
    color: rgb(1, 1, 1);
    padding: 10px;
    text-align: center;
    text-decoration: solid;
}

.editimage{
    height: 5%;
    width: 5%;
    position: absolute;
    top: 58%;
    transform: translateY(-50%);
    width: auto;
    left: 57%;
}
```

### **createnewpost.css**

```
.uppostcontainer{
    width: 100%;
    display: block;
    overflow: auto;
}

.filediv {
```



```
    width: 55%;
    float: left;
    position: relative;
  }
  .captiondiv{
    width: 45%;
    float: right;
  }
  input[type=file] {
    position: absolute;
    margin: 0 35%;
    width: 30%;
    height: 40px;
    opacity: 0%;
    cursor: pointer;
  }
  .dummy {
    width: 30%;
    margin: 0 35%;
    background-color: #e80f0f;
    text-align: center;
    border-radius: 55px;
    height: 45px;
  }
  .dummy p{
    color: #F9F5F5;
    font-size: 18px;
    letter-spacing: 1px;
    margin: 0;
    padding-top: 5px;
  }
  input[type=file]:valid + .dummy {
    background-color: #03abde;
  }
  .postpicdiv{
    width: 40%;
    margin: 0 auto 10px auto;
  }
  #postpicid{
    width: 100%;
    aspect-ratio : 1 / 1;
    object-fit: cover;
  }
  .errormsg{
    margin: 20px auto;
    width: 40%;
    color: #cb0a0a;
    text-align: center;
  }
```



```
.postsubmitdiv{
  margin-top: 40px;
  width: 100%;
  float: left;
}
#uploadpost{
  width: 15%;
  background-color: #09d299;
  font-size: 1.3rem;
  padding: 0.8rem 0.5rem;
  border-radius: 55px;
  color: #fff;
  text-align: center;
  margin: auto;
  cursor: pointer;
  transition: background-color .4s;
}
#uploadpost:hover{
  background-color: #08b8d4
}
.captlabel, .loclabel{
  margin-top: 20px;
  font-weight: 600;
  font-size: 20px;
  letter-spacing: 1px;
  color: #647bcd;
}
textarea{
  width: 60%;
  height: 100px;
  padding: 12px 20px;
  border: 2px solid #cccccc;
  border-radius: 8px;
  font-size: 16px;
  letter-spacing: 1px;
  line-height: normal;
  color: #0b0b0b;
  resize: none;
  outline: none;
  font-family: "DM Sans", sans-serif;
}
textarea:focus{
  border: 2px solid #06c992;
}
select{
  width: 50%;
  height: 30px;
  border-radius: 4px;
  outline: none;
```



```

    border: 2px solid #696767;
    font-size: 16px;
    letter-spacing: 0.5px;
    line-height: normal;
    color: #666666;
  }
  select:focus{
    border: 2px solid #0bc792;
  }
  footer {
    /* background-color: #a36dd3;
    color: rgb(1, 1, 1);
    padding: 10px;
    text-align: center; */
    position: fixed;
    left: 0;
    bottom: 0;
    width: 100%;
    background-color: #6860b5;
    color: rgb(1, 1, 1);
    padding: 10px;
    text-align: center;
    text-decoration: solid;
  }

```

## 2.5 SQL Codes for Holiday Connect Database

### Table structure for table `comments`

```

CREATE TABLE `comments` (
  `CommentId` int(11) NOT NULL,
  `UserId` int(11) NOT NULL,
  `PostId` int(11) NOT NULL,
  `CommentBody` varchar(50) NOT NULL,
  `Timestamp` timestamp NOT NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

```

### Dumping data for table `comments`



```
INSERT INTO `comments` (`CommentId`, `UserId`, `PostId`, `CommentBody`, `Timestamp`)
VALUES
```

```
(1, 2, 5, 'Wow nccc !', '2024-05-11 08:13:10'),
(2, 2, 5, 'woo', '2024-05-11 08:15:36'),
(3, 3, 10, 'image', '2024-05-11 14:01:35'),
(4, 2, 11, 'wow', '2024-05-11 14:39:35'),
(5, 2, 10, 'ncc', '2024-05-11 14:39:56'),
(6, 7, 11, 'ncc', '2024-05-11 21:35:22');
```

### **Table structure for table `following`**

```
CREATE TABLE `following` (
  `FollowId` int(11) NOT NULL,
  `UserId` int(11) NOT NULL,
  `IsFollowing` int(11) NOT NULL,
  `Timestamp` timestamp NOT NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

### **Dumping data for table `following`**

```
INSERT INTO `following` (`FollowId`, `UserId`, `IsFollowing`, `Timestamp`) VALUES
(8, 2, 1, '2024-05-11 08:11:34'),
(13, 1, 2, '2024-05-11 11:18:48'),
(15, 1, 3, '2024-05-11 11:32:49'),
(16, 1, 1, '2024-05-11 15:26:33'),
(17, 4, 2, '2024-05-11 15:27:33'),
(18, 4, 1, '2024-05-11 15:28:56'),
(20, 3, 1, '2024-05-11 20:48:33'),
(21, 3, 3, '2024-05-11 20:49:00'),
(22, 6, 2, '2024-05-11 21:07:28'),
(24, 7, 1, '2024-05-11 21:35:11'),
```



```
(25, 7, 2, '2024-05-11 21:35:27'),  
(27, 6, 1, '2024-05-11 21:39:00');
```

### Table structure for table `likes`

```
CREATE TABLE `likes` (  
  `LikeId` int(11) NOT NULL,  
  `UserId` int(11) NOT NULL,  
  `PostId` int(11) NOT NULL,  
  `Timestamp` timestamp NOT NULL DEFAULT current_timestamp()  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

### Dumping data for table `likes`

```
INSERT INTO `likes` (`LikeId`, `UserId`, `PostId`, `Timestamp`) VALUES  
(2, 2, 5, '2024-05-11 08:11:38'),  
(3, 2, 6, '2024-05-11 08:12:26'),  
(6, 1, 6, '2024-05-11 08:22:12'),  
(8, 1, 7, '2024-05-11 09:36:28'),  
(9, 1, 11, '2024-05-11 10:04:59'),  
(10, 1, 10, '2024-05-11 10:05:00'),  
(11, 1, 9, '2024-05-11 10:05:01'),  
(12, 1, 8, '2024-05-11 10:05:04'),  
(13, 1, 5, '2024-05-11 10:05:06'),  
(15, 3, 10, '2024-05-11 11:32:14'),  
(16, 3, 9, '2024-05-11 11:32:15'),  
(17, 3, 8, '2024-05-11 11:32:17'),  
(18, 3, 7, '2024-05-11 11:32:18'),  
(19, 3, 5, '2024-05-11 11:32:19'),  
(20, 1, 12, '2024-05-11 11:32:58'),  
(21, 3, 11, '2024-05-11 14:01:40'),
```





```
(23, 2, 11, '2024-05-11 14:38:36'),  
(24, 2, 10, '2024-05-11 14:38:37'),  
(25, 2, 9, '2024-05-11 14:38:38'),  
(26, 2, 14, '2024-05-11 14:39:20'),  
(27, 1, 14, '2024-05-11 14:50:31'),  
(28, 1, 15, '2024-05-11 18:24:26'),  
(29, 4, 14, '2024-05-11 19:44:47'),  
(30, 4, 11, '2024-05-11 19:44:48'),  
(31, 4, 10, '2024-05-11 19:44:49'),  
(32, 4, 9, '2024-05-11 19:44:56'),  
(33, 4, 8, '2024-05-11 19:44:57'),  
(34, 4, 7, '2024-05-11 19:44:58'),  
(35, 4, 16, '2024-05-11 19:49:49'),  
(36, 3, 14, '2024-05-11 20:33:45'),  
(37, 3, 6, '2024-05-11 20:33:52'),  
(38, 3, 15, '2024-05-11 20:48:49'),  
(39, 3, 13, '2024-05-11 20:49:04'),  
(40, 3, 12, '2024-05-11 20:49:05'),  
(41, 6, 14, '2024-05-11 21:07:43'),  
(42, 6, 11, '2024-05-11 21:07:44'),  
(43, 6, 10, '2024-05-11 21:07:47'),  
(44, 6, 18, '2024-05-11 21:08:52'),  
(45, 7, 19, '2024-05-11 21:35:05'),  
(46, 7, 11, '2024-05-11 21:35:16'),  
(47, 7, 14, '2024-05-11 21:36:11'),  
(48, 7, 10, '2024-05-11 21:36:12');
```

#### **Table structure for table `location`**

```
CREATE TABLE `location` (  
  `LocationId` int(11) NOT NULL,  
  `LocationName` varchar(255) NOT NULL
```



```
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

### **Dumping data for table `location`**

```
INSERT INTO `location` (`LocationId`, `LocationName`) VALUES
```

```
(1, ''),  
(2, 'Afghanistan'),  
(3, 'Albania'),  
(4, 'Algeria'),  
(5, 'American Samoa'),  
(6, 'Andorra'),  
(7, 'Angola'),  
(8, 'Anguilla'),  
(9, 'Antarctica'),  
(10, 'Antigua and Barbuda'),  
(11, 'Argentina'),  
(12, 'Armenia'),  
(13, 'Aruba'),  
(14, 'Australia'),  
(15, 'Austria'),  
(16, 'Azerbaijan'),  
(17, 'Bahamas'),  
(18, 'Bahrain'),  
(19, 'Bangladesh'),  
(20, 'Barbados'),  
(21, 'Belarus'),  
(22, 'Belgium'),  
(23, 'Belize'),  
(24, 'Benin'),  
(25, 'Bermuda'),  
(26, 'Bhutan'),  
(27, 'Bosnia and Herzegovina'),
```



(28, 'Botswana'),  
(29, 'Bouvet Island'),  
(30, 'Brazil'),  
(31, 'British Indian Ocean Territory'),  
(32, 'Brunei Darussalam'),  
(33, 'Bulgaria'),  
(34, 'Burkina Faso'),  
(35, 'Burundi'),  
(36, 'Cambodia'),  
(37, 'Cameroon'),  
(38, 'Canada'),  
(39, 'Cape Verde'),  
(40, 'Cayman Islands'),  
(41, 'Central African Republic'),  
(42, 'Chad'),  
(43, 'Chile'),  
(44, 'China'),  
(45, 'Christmas Island'),  
(46, 'Cocos (Keeling) Islands'),  
(47, 'Colombia'),  
(48, 'Comoros'),  
(49, 'Congo'),  
(50, 'Cook Islands'),  
(51, 'Costa Rica'),  
(52, 'Croatia'),  
(53, 'Cuba'),  
(54, 'Cyprus'),  
(55, 'Czech Republic'),  
(56, 'Denmark'),  
(57, 'Djibouti'),  
(58, 'Dominica'),  
(59, 'Dominican Republic'),



(60, 'Ecuador'),  
(61, 'Egypt'),  
(62, 'El Salvador'),  
(63, 'Equatorial Guinea'),  
(64, 'Eritrea'),  
(65, 'Estonia'),  
(66, 'Ethiopia'),  
(67, 'Falkland Islands (Malvinas)'),  
(68, 'Faroe Islands'),  
(69, 'Fiji'),  
(70, 'Finland'),  
(71, 'France'),  
(72, 'French Guiana'),  
(73, 'French Polynesia'),  
(74, 'French Southern Territories'),  
(75, 'Gabon'),  
(76, 'Gambia'),  
(77, 'Georgia'),  
(78, 'Germany'),  
(79, 'Ghana'),  
(80, 'Gibraltar'),  
(81, 'Greece'),  
(82, 'Greenland'),  
(83, 'Grenada'),  
(84, 'Guadeloupe'),  
(85, 'Guam'),  
(86, 'Guatemala'),  
(87, 'Guernsey'),  
(88, 'Guinea'),  
(89, 'Guinea-Bissau'),  
(90, 'Guyana'),  
(91, 'Haiti'),



(92, 'Heard Island and McDonald Islands'),  
(93, 'Holy See (Vatican City State)'),  
(94, 'Honduras'),  
(95, 'Hong Kong'),  
(96, 'Hungary'),  
(97, 'Iceland'),  
(98, 'India'),  
(99, 'Indonesia'),  
(100, 'Iraq'),  
(101, 'Ireland'),  
(102, 'Isle of Man'),  
(103, 'Israel'),  
(104, 'Italy'),  
(105, 'Jamaica'),  
(106, 'Japan'),  
(107, 'Jersey'),  
(108, 'Jordan'),  
(109, 'Kazakhstan'),  
(110, 'Kenya'),  
(111, 'Kiribati'),  
(112, 'Kuwait'),  
(113, 'Kyrgyzstan'),  
(114, 'Lao Peoples Democratic Republic'),  
(115, 'Latvia'),  
(116, 'Lebanon'),  
(117, 'Lesotho'),  
(118, 'Liberia'),  
(119, 'Libya'),  
(120, 'Liechtenstein'),  
(121, 'Lithuania'),  
(122, 'Luxembourg'),  
(123, 'Macao'),



(124, 'Madagascar'),  
(125, 'Malawi'),  
(126, 'Malaysia'),  
(127, 'Maldives'),  
(128, 'Mali'),  
(129, 'Malta'),  
(130, 'Marshall Islands'),  
(131, 'Martinique'),  
(132, 'Mauritania'),  
(133, 'Mauritius'),  
(134, 'Mayotte'),  
(135, 'Mexico'),  
(136, 'Monaco'),  
(137, 'Mongolia'),  
(138, 'Montenegro'),  
(139, 'Montserrat'),  
(140, 'Morocco'),  
(141, 'Mozambique'),  
(142, 'Myanmar'),  
(143, 'Namibia'),  
(144, 'Nauru'),  
(145, 'Nepal'),  
(146, 'Netherlands'),  
(147, 'New Caledonia'),  
(148, 'New Zealand'),  
(149, 'Nicaragua'),  
(150, 'Niger'),  
(151, 'Nigeria'),  
(152, 'Niue'),  
(153, 'Norfolk Island'),  
(154, 'Northern Mariana Islands'),  
(155, 'Norway'),



(156, 'Oman'),  
(157, 'Pakistan'),  
(158, 'Palau'),  
(159, 'Panama'),  
(160, 'Papua New Guinea'),  
(161, 'Paraguay'),  
(162, 'Peru'),  
(163, 'Philippines'),  
(164, 'Pitcairn'),  
(165, 'Poland'),  
(166, 'Portugal'),  
(167, 'Puerto Rico'),  
(168, 'Qatar'),  
(169, 'Romania'),  
(170, 'Russian Federation'),  
(171, 'Rwanda'),  
(172, 'Saint Kitts and Nevis'),  
(173, 'Saint Lucia'),  
(174, 'Saint Martin (French part)'),  
(175, 'Saint Pierre and Miquelon'),  
(176, 'Saint Vincent and the Grenadines'),  
(177, 'Samoa'),  
(178, 'San Marino'),  
(179, 'Sao Tome and Principe'),  
(180, 'Saudi Arabia'),  
(181, 'Senegal'),  
(182, 'Serbia'),  
(183, 'Seychelles'),  
(184, 'Sierra Leone'),  
(185, 'Singapore'),  
(186, 'Sint Maarten (Dutch part)'),  
(187, 'Slovakia'),



(188, 'Slovenia'),  
(189, 'Solomon Islands'),  
(190, 'Somalia'),  
(191, 'South Africa'),  
(192, 'South Georgia and the South Sandwich Islands'),  
(193, 'South Sudan'),  
(194, 'Spain'),  
(195, 'Sri Lanka'),  
(196, 'Sudan'),  
(197, 'SuriLocationName'),  
(198, 'Svalbard and Jan Mayen'),  
(199, 'Swaziland'),  
(200, 'Sweden'),  
(201, 'Switzerland'),  
(202, 'Syrian Arab Republic'),  
(203, 'Tajikistan'),  
(204, 'Thailand'),  
(205, 'Timor-Leste'),  
(206, 'Togo'),  
(207, 'Tokelau'),  
(208, 'Tonga'),  
(209, 'Trinidad and Tobago'),  
(210, 'Tunisia'),  
(211, 'Turkey'),  
(212, 'Turkmenistan'),  
(213, 'Turks and Caicos Islands'),  
(214, 'Tuvalu'),  
(215, 'Uganda'),  
(216, 'Ukraine'),  
(217, 'United Arab Emirates'),  
(218, 'United Kingdom'),  
(219, 'United States'),





```
(220, 'United States Minor Outlying Islands'),  
(221, 'Uruguay'),  
(222, 'Uzbekistan'),  
(223, 'Vanuatu'),  
(224, 'Viet Nam'),  
(225, 'Wallis and Futuna'),  
(226, 'Western Sahara'),  
(227, 'Yemen'),  
(228, 'Zambia'),  
(229, 'Zimbabwe');
```

#### **Table structure for table `notification`**

```
CREATE TABLE `notification` (  
  `NotifId` int(11) NOT NULL,  
  `UserId` int(11) NOT NULL,  
  `FromUser` int(11) NOT NULL,  
  `PostId` int(11) DEFAULT NULL,  
  `CommentBody` varchar(50) DEFAULT NULL,  
  `Notification` varchar(100) NOT NULL,  
  `Timestamp` timestamp NOT NULL DEFAULT current_timestamp()  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

#### **Dumping data for table `notification`**

```
INSERT INTO `notification` (`NotifId`, `UserId`, `FromUser`, `PostId`, `CommentBody`,  
  `Notification`, `Timestamp`) VALUES  
(9, 1, 2, NULL, NULL, 'Followed you!', '2024-05-11 08:11:34'),  
(10, 1, 2, 5, NULL, 'Liked your post!', '2024-05-11 08:11:38'),  
(11, 2, 2, 6, NULL, 'Liked your post!', '2024-05-11 08:12:26'),  
(12, 1, 2, 5, 'Wow nccc !', 'Commented on your post!', '2024-05-11 08:13:10'),
```



(13, 1, 2, 5, 'woo', 'Commented on your post!', '2024-05-11 08:15:36'),  
(17, 2, 1, 6, NULL, 'Liked your post!', '2024-05-11 08:22:12'),  
(20, 1, 1, 7, NULL, 'Liked your post!', '2024-05-11 09:36:28'),  
(23, 1, 1, 11, NULL, 'Liked your post!', '2024-05-11 10:04:59'),  
(24, 1, 1, 10, NULL, 'Liked your post!', '2024-05-11 10:05:00'),  
(25, 1, 1, 9, NULL, 'Liked your post!', '2024-05-11 10:05:01'),  
(26, 1, 1, 8, NULL, 'Liked your post!', '2024-05-11 10:05:04'),  
(27, 1, 1, 5, NULL, 'Liked your post!', '2024-05-11 10:05:06'),  
(28, 2, 1, NULL, NULL, 'Followed you!', '2024-05-11 11:18:48'),  
(31, 1, 3, 10, NULL, 'Liked your post!', '2024-05-11 11:32:14'),  
(32, 1, 3, 9, NULL, 'Liked your post!', '2024-05-11 11:32:15'),  
(33, 1, 3, 8, NULL, 'Liked your post!', '2024-05-11 11:32:17'),  
(34, 1, 3, 7, NULL, 'Liked your post!', '2024-05-11 11:32:18'),  
(35, 1, 3, 5, NULL, 'Liked your post!', '2024-05-11 11:32:19'),  
(36, 3, 1, NULL, NULL, 'Followed you!', '2024-05-11 11:32:49'),  
(37, 3, 1, 12, NULL, 'Liked your post!', '2024-05-11 11:32:58'),  
(38, 1, 3, 10, 'image', 'Commented on your post!', '2024-05-11 14:01:35'),  
(39, 1, 3, 11, NULL, 'Liked your post!', '2024-05-11 14:01:40'),  
(41, 1, 2, 11, NULL, 'Liked your post!', '2024-05-11 14:38:36'),  
(42, 1, 2, 10, NULL, 'Liked your post!', '2024-05-11 14:38:37'),  
(43, 1, 2, 9, NULL, 'Liked your post!', '2024-05-11 14:38:38'),  
(44, 2, 2, 14, NULL, 'Liked your post!', '2024-05-11 14:39:20'),  
(45, 1, 2, 11, 'wow', 'Commented on your post!', '2024-05-11 14:39:35'),  
(46, 1, 2, 10, 'ncc', 'Commented on your post!', '2024-05-11 14:39:56'),  
(47, 2, 1, 14, NULL, 'Liked your post!', '2024-05-11 14:50:31'),  
(48, 1, 1, NULL, NULL, 'Followed you!', '2024-05-11 15:26:33'),  
(49, 2, 4, NULL, NULL, 'Followed you!', '2024-05-11 15:27:33'),  
(50, 1, 4, NULL, NULL, 'Followed you!', '2024-05-11 15:28:56'),  
(51, 3, 1, 15, NULL, 'Liked your post!', '2024-05-11 18:24:26'),  
(52, 2, 4, 14, NULL, 'Liked your post!', '2024-05-11 19:44:47'),  
(53, 1, 4, 11, NULL, 'Liked your post!', '2024-05-11 19:44:48'),  
(54, 1, 4, 10, NULL, 'Liked your post!', '2024-05-11 19:44:49'),



```
(55, 1, 4, 9, NULL, 'Liked your post!', '2024-05-11 19:44:56'),
(56, 1, 4, 8, NULL, 'Liked your post!', '2024-05-11 19:44:57'),
(57, 1, 4, 7, NULL, 'Liked your post!', '2024-05-11 19:44:58'),
(58, 4, 4, 16, NULL, 'Liked your post!', '2024-05-11 19:49:49'),
(60, 2, 3, 14, NULL, 'Liked your post!', '2024-05-11 20:33:45'),
(61, 2, 3, 6, NULL, 'Liked your post!', '2024-05-11 20:33:52'),
(62, 1, 3, NULL, NULL, 'Followed you!', '2024-05-11 20:48:33'),
(63, 3, 3, 15, NULL, 'Liked your post!', '2024-05-11 20:48:49'),
(64, 3, 3, NULL, NULL, 'Followed you!', '2024-05-11 20:49:00'),
(65, 3, 3, 13, NULL, 'Liked your post!', '2024-05-11 20:49:04'),
(66, 3, 3, 12, NULL, 'Liked your post!', '2024-05-11 20:49:05'),
(67, 2, 6, NULL, NULL, 'Followed you!', '2024-05-11 21:07:28'),
(69, 2, 6, 14, NULL, 'Liked your post!', '2024-05-11 21:07:43'),
(70, 1, 6, 11, NULL, 'Liked your post!', '2024-05-11 21:07:44'),
(71, 1, 6, 10, NULL, 'Liked your post!', '2024-05-11 21:07:47'),
(72, 6, 6, 18, NULL, 'Liked your post!', '2024-05-11 21:08:52'),
(73, 7, 7, 19, NULL, 'Liked your post!', '2024-05-11 21:35:05'),
(74, 1, 7, NULL, NULL, 'Followed you!', '2024-05-11 21:35:11'),
(75, 1, 7, 11, NULL, 'Liked your post!', '2024-05-11 21:35:16'),
(76, 1, 7, 11, 'ncc', 'Commented on your post!', '2024-05-11 21:35:22'),
(77, 2, 7, NULL, NULL, 'Followed you!', '2024-05-11 21:35:27'),
(78, 2, 7, 14, NULL, 'Liked your post!', '2024-05-11 21:36:11'),
(79, 1, 7, 10, NULL, 'Liked your post!', '2024-05-11 21:36:12'),
(81, 1, 6, NULL, NULL, 'Followed you!', '2024-05-11 21:39:00');
```

### Table structure for table `posts`

```
CREATE TABLE `posts` (
  `PostId` int(11) NOT NULL,
  `UserId` int(11) NOT NULL,
  `LocationId` int(11) NOT NULL DEFAULT 1,
```



```
`PostImage` varchar(255) NOT NULL DEFAULT 'default.jpg',
`Caption` varchar(100) DEFAULT "",
`Timestamp` timestamp NOT NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

### Dumping data for table `posts`

```
INSERT INTO `posts` (`PostId`, `UserId`, `LocationId`, `PostImage`, `Caption`, `Timestamp`)
VALUES
(5, 1, 195, '9901b9cf-8e58-4c3a-8fea-e2ed5c0f3f954.JPG', 'Gaming Logo', '2024-05-11
08:10:48'),
(6, 2, 195, 'download_(5).png', 'I have started my university life at IIT.', '2024-05-11 08:12:21'),
(7, 1, 1, '68D561B5B7CA417181FDABBA65D0E10F.jpg', "", '2024-05-11 09:36:22'),
(8, 1, 1, 'jonatan-pie-EvKBHBGgaUo-unsplash.jpg', "", '2024-05-11 09:40:30'),
(9,
1,
219,
'MV5BMjMxNjY2MDU1OV5BMl5BanBnXkFtZTgwNzY1MTUwNTM@__V1_FMjpg_U
X1000_.jpg', 'Avengers', '2024-05-11 09:55:15'),
(10, 1, 1, 'WallpaperDog-5487207.jpg', "", '2024-05-11 09:58:32'),
(11, 1, 2, 'IMG_1879.jpg', 'Selfie', '2024-05-11 10:02:50'),
(12, 3, 1, 'Black_White_Minimalist_Business_Logo.png', "", '2024-05-11 11:31:53'),
(13, 3, 1, 'Untitled_Project.jpg', "", '2024-05-11 14:00:46'),
(14, 2, 219, 'download_(4).png', 'github', '2024-05-11 14:39:11'),
(15, 3, 1, 'Black_White_Minimalist_Business_Logo1.png', "", '2024-05-11 16:18:42'),
(16, 4, 2, 'Black_White_Minimalist_Business_Logo2.png', 'holiday team', '2024-05-11
19:47:27'),
(17, 3, 195, 'Untitled_Project_(1).jpg', "", '2024-05-11 20:49:59'),
(18, 6, 195, 'Untitled_Project1.jpg', 'gaming', '2024-05-11 21:08:48'),
(19, 7, 5, '9901b9cf-8e58-4c3a-8fea-e2ed5c0f3f955.JPG', 'Comment', '2024-05-11 21:35:02');
```

### Table structure for table `users`



```
CREATE TABLE `users` (
  `UserId` int(11) NOT NULL,
  `Username` varchar(255) NOT NULL,
  `Name` varchar(255) NOT NULL,
  `Email` varchar(255) NOT NULL,
  `Password` varchar(255) NOT NULL,
  `UserBio` varchar(120) DEFAULT "",
  `UserImage` varchar(255) NOT NULL DEFAULT 'default.jpg'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

### Dumping data for table `users`

```
INSERT INTO `users` (`UserId`, `Username`, `Name`, `Email`, `Password`, `UserBio`,
`UserImage`) VALUES
(1, '@chirath', 'Chirath Thusaraka Sirimanna', 'chirath.20191168@iit.ac.lk',
'$2y$10$TORos4I0gsfPox1qOkPNxe9g15eMKBKqhCmlD2t1mAXhE9w8i/sWC', 'From Sri
Lanka.\n23 Y Gamer.', 'IMG_2766_(1).jpg'),
(2, '@chirath1', 'Chirath Thusaraka Sirimanna', 'chirath.20191168@iit.ac.lk',
'$2y$10$fjyqwQT6wCS8rcOTUGPy.ejE2fJNICWpT5UIVuwRgpiFgk1CsYlK', 'Hi im
chirath', 'IMG_1879-removebg-preview.png'),
(3, '@chirath2', 'Chirath Thusaraka Sirimanna', 'chirath.20191168@iit.ac.lk',
'$2y$10$Yyt5YSLsy4PT2PuMizuOBVo3UsqDEixwVWk2a14RTGRqEbsBGRW', '',
'default.jpg'),
(4, '@chirath4', 'Chirath Thusaraka Sirimanna', 'chirath.20191168@iit.ac.lk',
'$2y$10$1e4wV4wKMODEVnU7qNJLT.Kl6RK.pm/aKyo6c4cmlZbpP6PuNV1HW', '',
'default.jpg'),
(6, '@chirath3', 'Chirath Thusaraka Sirimanna', 'chirath.20191168@iit.ac.lk',
'$2y$10$PKk5Y1U/U9KBXDj8ti.7AeBof36God.2W1HoI6k3uV1UuMnR.V5wW', 'Hi 19Y.',
'default.jpg'),
(7, '@chirath6', 'Chirath Thusaraka Sirimanna', 'chirath.20191168@iit.ac.lk',
'$2y$10$5qOGMBctGB7c9h5xFKmWfOU15XIGYGCr95Pia7NGr5waM1GIP2.fO', '',
'default.jpg');
```



## Indexes for dumped tables

### Indexes for table `comments`

```
ALTER TABLE `comments`  
  ADD PRIMARY KEY (`CommentId`),  
  ADD KEY `FK_CommentUser` (`UserId`),  
  ADD KEY `FK_CommentPost` (`PostId`);
```

### Indexes for table `following`

```
ALTER TABLE `following`  
  ADD PRIMARY KEY (`FollowId`),  
  ADD KEY `FK_FollowUser` (`UserId`),  
  ADD KEY `FK_FollowIsFollowing` (`IsFollowing`);
```

### Indexes for table `likes`

```
ALTER TABLE `likes`  
  ADD PRIMARY KEY (`LikeId`),  
  ADD KEY `FK_LikeUser` (`UserId`),  
  ADD KEY `FK_LikePost` (`PostId`);
```

### Indexes for table `location`

```
ALTER TABLE `location`  
  ADD PRIMARY KEY (`LocationId`);
```

**Indexes for table `notification`**

```
ALTER TABLE `notification`  
  ADD PRIMARY KEY (`NotifId`),  
  ADD KEY `FK_NotifUser` (`UserId`),  
  ADD KEY `FK_NotifFromUser` (`FromUser`);
```

**Indexes for table `posts`**

```
ALTER TABLE `posts`  
  ADD PRIMARY KEY (`PostId`),  
  ADD KEY `FK_PostUser` (`UserId`),  
  ADD KEY `FK_PostLocation` (`LocationId`);
```

**Indexes for table `users`**

```
ALTER TABLE `users`  
  ADD PRIMARY KEY (`UserId`),  
  ADD UNIQUE KEY `Username` (`Username`);
```

**AUTO\_INCREMENT for dumped tables****AUTO\_INCREMENT for table `comments`**

```
ALTER TABLE `comments`  
  MODIFY `CommentId` int(11) NOT NULL AUTO_INCREMENT,  
  AUTO_INCREMENT=7;
```

**AUTO\_INCREMENT for table `following`**

```
ALTER TABLE `following`  
  MODIFY `FollowId` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=28;
```

**AUTO\_INCREMENT for table `likes`**

```
ALTER TABLE `likes`  
  MODIFY `LikeId` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=49;
```

**AUTO\_INCREMENT for table `notification`**

```
ALTER TABLE `notification`  
  MODIFY `NotifId` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=82;
```

**AUTO\_INCREMENT for table `posts`**

```
ALTER TABLE `posts`  
  MODIFY `PostId` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=20;
```

**AUTO\_INCREMENT for table `users`**

```
ALTER TABLE `users`  
  MODIFY `UserId` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=8;
```

**Constraints for dumped tables**



**Constraints for table `comments`**

```
ALTER TABLE `comments`  
  ADD CONSTRAINT `FK_CommentPost` FOREIGN KEY (`PostId`) REFERENCES `posts`  
  (`PostId`),  
  ADD CONSTRAINT `FK_CommentUser` FOREIGN KEY (`UserId`) REFERENCES  
  `users` (`UserId`);
```

**Constraints for table `following`**

```
ALTER TABLE `following`  
  ADD CONSTRAINT `FK_FollowIsFollowing` FOREIGN KEY (`IsFollowing`)  
  REFERENCES `users` (`UserId`),  
  ADD CONSTRAINT `FK_FollowUser` FOREIGN KEY (`UserId`) REFERENCES `users`  
  (`UserId`);
```

**Constraints for table `likes`**

```
ALTER TABLE `likes`  
  ADD CONSTRAINT `FK_LikePost` FOREIGN KEY (`PostId`) REFERENCES `posts`  
  (`PostId`),  
  ADD CONSTRAINT `FK_LikeUser` FOREIGN KEY (`UserId`) REFERENCES `users`  
  (`UserId`);
```

**Constraints for table `notification`**

```
ALTER TABLE `notification`  
  ADD CONSTRAINT `FK_NotifFromUser` FOREIGN KEY (`FromUser`) REFERENCES  
  `users` (`UserId`),
```



```
ADD CONSTRAINT `FK_NotifUser` FOREIGN KEY (`UserId`) REFERENCES `users`  
(`UserId`);
```

### **Constraints for table `posts`**

```
ALTER TABLE `posts`  
  ADD CONSTRAINT `FK_PostLocation` FOREIGN KEY (`LocationId`) REFERENCES  
  `location` (`LocationId`),  
  ADD CONSTRAINT `FK_PostUser` FOREIGN KEY (`UserId`) REFERENCES `users`  
  (`UserId`);  
COMMIT;
```



### 3. Evaluation of Design from Coursework 1

#### 3.1 Changes to requirements from Coursework 1

- **Additional luxury items included** - implemented night mode for easier visualization of the web application to the users since this is a social media application.
- **Additionally added five different language files** – adding additional languages it will be easy to users that are using different languages instead of english since this is a social media application.
- **There are no alterations to the other requirements from Coursework 1.**

#### 3.2 Database Script Changes

- **Columns FromUser, PostId, and CommentBody have been added to the notifications table to differentiate between users and types of notifications.**

#### 3.3 Class Model Changes

- The Posts model was renamed to PostModelFile and additional methods added to the PostModelFile model included
- The Users model was renamed to UserModelFile and additional methods added to the UserModelFile model included.
- Changes in methods within the Home Controller involve moving **profile, create post, notifications, and location tags to other controllers**, while **following posts, comments, and checklikes get methods were added**, and **like and comments post methods** were added to it.
- The MyProfile controller was renamed to ProfileCon, and changes in methods within the ProfileCon controller include the logout method being relocated to another controller and additional methods added to this controller.



- The CreatePost controller include methods such as **profile picture post, create post, and user posts, location, post, posts location, and like count get** methods being incorporated.
- The Login controller was renamed to LoginCon, and additional methods like **logout, user profile, user get, user post, password reset, userdetails get methods, and edit profile post** methods were included.
- Extra views added to the application include **edit profile, navigation, reset password, post, sign up, and user profile**, to showcase required functionalities.

### 3.4 Discussion Paragraph

I found it incredibly beneficial that the application's design was carefully developed in advance before start of the actual implementation. This approach was particularly helpful me in ensuring the accurate tracking of essential requirements. If I were to start over, I would like to dedicate more time to design and add more functionalities to comprehensive class diagram, outlining all necessary classes and methods with utmost clarity.



## 4. Discussion of Article

### 4.1 Question 1

Web frameworks are software tools designed to assist developers in creating web applications in a standardized manner, abstracting away some of the complexities of the underlying technologies. This article delves into the adverse effects that frameworks can have on developers.

The article suggests that beginners who rely solely on frameworks without understanding the fundamentals of technologies like HTTP or the structure of a POST request may struggle in the long term. This lack of foundational knowledge to web application development leaves them dependent on shifting from one trending framework to another, often in pursuit of features that newer frameworks offer. Moreover, developers operating within this framework-centric approach may experience problems, feeling uncertain about their understanding of the underlying technologies beyond the framework's interface. They risk becoming categorized as specialists in specific frameworks, such as React software developers and Angular software developers.

The article presents a second reason the developers often opt for frameworks because working without them can be comparatively challenging. The author contends that with the expensive resources available on the web today. Including documentation and technology tutorials, navigating without frameworks is more feasible than in the past. For instance, the author suggests using native CSS features like variables instead of relying on pre-processors like Sass and utilizing flexbox and grid layouts instead of frameworks like Bootstrap for client-side styling.

Furthermore, the article addresses a third reason the importance of mastering the basics rather than relying solely on someone else's abstraction. Frequently developers encounter situations where they seek solutions within a new framework only to find that their problem cannot be resolved within its confines or requires a vastly different approach. Thus, the author advocates for prioritizing understanding fundamental concepts such as HTTP, HTML, CSS, and JavaScript over diving directly into advanced frameworks like React with Redux and



Bootstrap, as well as embracing foundational technologies like SQL instead of other framework-based solutions.

## 4.2 Question 2

In my view, the author highlights a critical aspect regarding how frameworks can negatively impact developers. It is emphasized that when newcomers utilize frameworks without a foundational understanding of web development, their learning process becomes interrupt. Lacking this fundamental knowledge, transitioning to a newer, trendy framework can result in confusion, as one may struggle to comprehend the underlying processes. This problem up arises because the developer is simply familiar with the properties of the framework they used previously. Additionally, the author points out that accessing documentation and tutorials for coding has become increasingly convenient in today's digital life.

Furthermore, I have a different opinion regarding the ease of development without the use of web frameworks. In my experience, having a solid understanding of the basics allows developers to harness frameworks more effectively. With this foundational knowledge, navigating through frameworks becomes much smoother, facilitating the achievement of desired outcomes in a more efficient manner.



## 5. References

**Source - Article Why You Shouldn't Use A Web Framework**

Link:- <https://dev.to/gypsydave5/why-you-shouldnt-use-a-web-framework-3g24>.