INFORMATICS INSTITUTE OF TECHNOLOGY

In Collaboration with

UNIVERSITY OF WESTMINSTER

6COSC022C.1
Advanced Server-Side Web Programming

Coursework 2

Student IIT ID	20191258
Student UOW ID	W1790276
Student name	Nanduni Mihisaree Gajanayake
Date of submission	19.01.2022

1. URL of the application

https://w1790276.users.ecs.westminster.ac.uk/CWK2/index.php/Landing

2. Code

2.1. explore_view.php

path – views/ explore.php

```
<?php
$this->load->view('templates/header.php');
?>

<?php
$this->load->view('templates/nav.php');
?>

<!-- section 1 -->
<div class="d-flex justify-content-center my-3">
    <form action="">
        <input class="search_input p-3" type="text" placeholder="Search.."
name="search">
        <button class="border-0 bg-transparent" type="submit"><img class="icon"
src="../assets/images/search.png" alt=""></button>
    </form>
</div>
<!--/ section 1 -->

<!-- section 2 -->
<div class="row m-0 py-5">
    <div class="col">
        <div class="mb-5">
            <img class="explore-img" src="../assets/images/signup-bbg.jpg" alt=""
onclick="enlargeImg(this)">
        </div>
        <div class="mb-5"><img class="explore-img" src="../assets/images/signup-
bbg.jpg" alt="" onclick="enlargeImg(this)"></div>
        <div class="mb-5"><img class="explore-img" src="../assets/images/signup-
bbg.jpg" alt="" onclick="enlargeImg(this)"></div>
        <div class="mb-5"><img class="explore-img" src="../assets/images/signup-
bbg.jpg" alt="" onclick="enlargeImg(this)"></div>
    </div>
    <div class="col">
        <div class="mb-5"><img class="explore-img" src="../assets/images/signup-bbg.jpg" alt="" onclick="enlargeImg(this)"></div>
        <div class="mb-5"><img class="explore-img" src="../assets/images/signup-bbg.jpg" alt="" onclick="enlargeImg(this)"></div>
        <div class="mb-5"><img class="explore-img" src="../assets/images/signup-bbg.jpg" alt="" onclick="enlargeImg(this)"></div>
        <div class="mb-5"><img class="explore-img" src="../assets/images/signup-bbg.jpg" alt="" onclick="enlargeImg(this)"></div>
    </div>
    <div class="col">
        <div class="mb-5"><img class="explore-img" src="../assets/images/signup-bbg.jpg" alt="" onclick="enlargeImg(this)"></div>
        <div class="mb-5"><img class="explore-img" src="../assets/images/signup-bbg.jpg" alt="" onclick="enlargeImg(this)"></div>
        <div class="mb-5"><img class="explore-img" src="../assets/images/signup-bbg.jpg" alt="" onclick="enlargeImg(this)"></div>
        <div class="mb-5"><img class="explore-img" src="../assets/images/signup-bbg.jpg" alt="" onclick="enlargeImg(this)"></div>
    </div>
</div>
<!--/ section 2 -->

<?php
$this->load->view('templates/footer.php');
?>
```

2.2. landing_view.php

path – views/landing_view.php

```
<?php
$this->load->view('templates/header.php');
?>

<?php
$this->load->view('templates/nav_landing.php');
?>

<!-- section 1 -->
<section>
    <div class="swiper bannerSwiper">
        <div class="swiper-wrapper">
            <div class="swiper-slide">
                <img src="../assets/images/signup-bbg.jpg" alt="">
            </div>
            <div class="swiper-slide">
                <img src="https://swiperjs.com/demos/images/nature-2.jpg" />
            </div>
            <div class="swiper-slide">
                <img src="https://swiperjs.com/demos/images/nature-3.jpg" />
            </div>
            <div class="swiper-slide">
                <img src="https://swiperjs.com/demos/images/nature-4.jpg" />
            </div>
        </div>

        <div class="swiper-pagination"></div>
    </div>
</section>
<!--/ section 1 -->


<!-- section 2 -->
<section class="">

    <div class="swiper homeSection2">
        <div class="d-flex justify-content-end pe-5 my-5">
            <img src="../assets/images/arrow-prev.svg" class="swiper-button-prev
mx-4 m-0" alt="">
            <img src="../assets/images/arrow-next.svg" class="swiper-button-next
mx-4 m-0" alt="">
        </div>
        <div class="swiper-wrapper mb-5">
            <div class="swiper-slide me-5 px-4"><img
src="../assets/images/signup-bbg.jpg" alt=""></div>
            <div class="swiper-slide me-5 px-4"><img
src="../assets/images/signup-bbg.jpg" alt=""></div>
            <div class="swiper-slide me-5 px-4"><img
src="../assets/images/signup-bbg.jpg" alt=""></div>
            <div class="swiper-slide me-5 px-4"><img
src="../assets/images/signup-bbg.jpg" alt=""></div>
            <div class="swiper-slide me-5 px-4"><img
src="../assets/images/signup-bbg.jpg" alt=""></div>
            <div class="swiper-slide me-5 px-4"><img
src="../assets/images/signup-bbg.jpg" alt=""></div>
        </div>

    </div>
</section>
<!--/ section 2 -->


<?php
$this->load->view('templates/footer.php');
```

```
?>
```

2.3. login_view.php

path – views/ login_view.php

```
<?php
$this->load->view('templates/header.php');
?>

<div class="row m-0 overflow-hidden vh-100">
    <div class="col-12 col-sm-6 p-0">
        <img src="../assets/images/signup-bbg.jpg" style="height: 100%;
width:100%; object-fit: cover;" alt="">

    </div>
    <div class="col-12 col-sm-6">
        <div class="align-items-center d-flex flex-column py-5">
            <div class="d-flex justify-content-center pb-5">
                <img src="../assets/images/logo1.jpg" style="height: 20%;
width:20%" alt="">
            </div>
            <div class="col-6">
                <form>
                    <h4>Log <span style="color: var(--dark-brown)">In</span></h4>
                    <input class="form_input my-3 p-3" type="text" id="fname"
name="fname" placeholder="Name"><br>
                    <input class="form_input my-3 p-3" type="text" id="fname"
name="fname" placeholder="Email"><br>
                    <input class="form_input my-3 p-3" type="text" id="fname"
name="fname" placeholder="Password"><br>
                    <input class="form_input my-3 p-3" type="text" id="fname"
name="fname" placeholder="Re-Enter password"><br>

                    <div class="d-flex justify-content-center my-4">
                        <input class="submit_button p-2 " type="submit"
value="Sign Up">
                    </div>


                </form>
            </div>

            <P>Don't have an account? <a href="#" style="color: var(--dark-
brown);text-decoration: none;">Sign Up</a></P>

        </div>
    </div>
```

```
</div>
```

2.4. signup_view.php

path – views/signup _view.php

```
<?php
$this->load->view('templates/header.php');
?>

<div class="row m-0 overflow-hidden vh-100" >
    <div class="col-12 col-sm-6 p-0">
        <img src="../assets/images/signup-bbg.jpg" style="height: 100%;
width:100%; object-fit: cover;" alt="">

    </div>
    <div class="col-12 col-sm-6">
        <div class="align-items-center d-flex flex-column py-5">
            <div class="d-flex justify-content-center pb-5">
                <img src="../assets/images/logo1.jpg" style="height: 20%;
width:20%" alt="">
            </div>
            <div class="col-6">
                <form>
                    <h4>Sign <span style="color: var(--dark-
brown)">Up</span></h4>
                    <input class="form_input my-3 p-3" type="text" id="fname"
name="fname" placeholder="Name"><br>
                    <input class="form_input my-3 p-3" type="text" id="fname"
name="fname" placeholder="Email"><br>
                    <input class="form_input my-3 p-3" type="text" id="fname"
name="fname" placeholder="Password"><br>
                    <input class="form_input my-3 p-3" type="text" id="fname"
name="fname" placeholder="Re-Enter password"><br>

                    <div class="d-flex justify-content-center my-4">
                        <input class="submit_button p-2 " type="submit"
value="Sign Up">
                    </div>


                </form>
            </div>

            <P>Already Have an account? <a href="#" style="color: var(--dark-
brown);text-decoration: none;">Log In</a></P>
```

```
        </div>
    </div>
</div>
```

2.5. header.php

path – views/templates/header.php

```
<!doctype html>
<html lang="en">

<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>CW_2</title>

    <!-- Boostrap css-->
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
rbsA2VBKQhggwzxH7pPCaAqO46MgnOM80zW1RWuH61DGLwZJEdK2Kadq2F9CUG65"
crossorigin="anonymous">
    <link rel="stylesheet" href="../assets/css/styles.css">

    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/swiper@8/swiper-
bundle.min.css" />

</head>

<body>
```

2.6.footer.php

path – views/templates/footer.php

```
<!-- Boostrap js -->
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js
" integrity="sha384-
kenU1KFdBIe4zVF0s0G1M5b4hcpxyD9F7jL+jjXkk+Q2h455rYXK/7HAuoJl+0I4"
crossorigin="anonymous"></script>
<script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/popper.min.js"
integrity="sha384-
oBqDVmMz9ATKxIep9tiCxS/Z9fNfEXiDAYTujMAeBAsjFuCZSmKbSSUnQlmh/jp3"
crossorigin="anonymous"></script>
```

```
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.min.js"
integrity="sha384-
cuYeSxntonz0PPNlHhBs68uyIAVpIIOZZ5JqeqvYYIcEL727kskC66kF92t6Xl2V"
crossorigin="anonymous"></script>

<!--  -->
<script src="https://cdn.jsdelivr.net/npm/swiper/swiper-bundle.min.js"></script>
<script src="../assets/js/main.js"></script>


</body>



</html>
```

2.7.nav_landing.php

path – views/templates/nav_landing.php

```
<nav class="navbar navbar-expand-lg ">
  <div class="container-fluid">


    <a class="" href="#"><img src="../assets/images/logo1.jpg" alt="logo"
style="max-height: 80px;"></a>

    <div class="collapse navbar-collapse" style="flex-grow: 0;">
      <ul class="navbar-nav me-auto mb-2 mb-lg-0">
        <a href="#">
          <li class="nav-item px-4">
            <button class="nav_button">LOGIN</button>

          </li>
        </a>
        <a href="#">
          <li class="nav-item px-4">
            <button class="nav_button">SIGNUP</button>
          </li>
        </a>
      </ul>
    </div>

  </div>
</nav>
```

2.8. nav.php

path – views/templates/nav.php

```
<nav class="navbar navbar-expand-lg ">
  <div class="container-fluid">


    <a class="" href="#"><img src="../assets/images/logo1.jpg" alt="logo" 
style="max-height: 80px;"></a>

    <div class="collapse navbar-collapse" style="flex-grow: 0;">
      <ul class="navbar-nav me-auto mb-2 mb-lg-0">
        <a href="#">
          <li class="nav-item px-2">
            <img class="icon" src="../assets/images/home.png" alt="">
          </li>
        </a>
        <a href="#">
          <li class="nav-item px-2">
            <img class="icon"  src="../assets/images/bell.png" alt="">
          </li>
        </a>
        <a href="#">
          <li class="nav-item px-2">
            <img class="icon"  src="../assets/images/search.png" alt="">
          </li>
        </a>
        <a href="#">
          <li class="nav-item px-2">
            <img class="icon"  src="../assets/images/add.png" alt="">
          </li>
        </a>
      </ul>
    </div>

  </div>
</nav>
```

2.8. AuthenticationModel.php

path – models/AuthenticationModel.php

```
<?php
```

```
class AuthenticationModel extends CI_Model
{

    public function __construct()
    {
        parent::__construct();
    }

public function register($username, $password, $email)
{
    $this->db->select('userName');
    $this->db->from('users');
    $this->db->where('userName', $username);
    $query = $this->db->get();
    if ($query->num_rows() == 0) {

        $passwordHash = password_hash($password, PASSWORD_DEFAULT);
        $data = array(
            'username' => $username,
            'password' => $passwordHash,
            'email' => $email
        );
        $this->db->insert('users', $data);
        if ($this->db->affected_rows() > 0) {
            return true;
        } else {
            return false;
        }
    } else {
        return false;
    }
}

public function login($username, $password)
{
    $this->db->select('userName, password');
    $this->db->from('users');
    $this->db->where('userName', $username);
    $query = $this->db->get();
    if ($query->num_rows() == 1) {
        $result = $query->result();
        $passwordHash = $result[0]->password;
        if (password_verify($password, $passwordHash)) {
            return true;
        } else {
            return false;
        }
    } else {
        return false;
    }
}

}
```

2.9. PostModel.php

path – models/PostModel.php

```
<?php

class PostModel extends CI_Model
{

    public function __construct()
    {
        parent::__construct();
    }

    public function getPosts()
    {
        $sql = 'SELECT * FROM post';
        $query = $this->db->query($sql);
        if ($query->num_rows() > 0) {
            return $query->result();
        } else {
            return false;
        }
    }

    public function getPost($id)
    {
        $sql = 'SELECT * FROM post WHERE userID = ' . $id . '';
        $query = $this->db->query($sql);
        if ($query->num_rows() > 0) {
            return $query->result();
        } else {
            return false;
        }
    }

    public function createPost($image, $caption, $user_id)
    {
        $data = array(
            'image' => $image,
            'caption' => $caption,
            'userID' => $user_id
        );
        $this->db->insert('post', $data);
        if ($this->db->affected_rows() > 0) {
            return true;
        } else {
            return false;
        }
    }

    public function deletePost($id)
    {
        $this->db->where('postID', $id);
        $this->db->delete('post');
        if ($this->db->affected_rows() > 0) {
            return true;
        } else {
            return false;
        }
    }

}
```

2.10. Authentication.php

path – controllers/api/Authentication.php

```
<?php

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

// This can be removed if you use __autoload() in config.php OR use Modular
Extensions
/** @noinspection PhpIncludeInspection */
require APPPATH . '/libraries/REST_Controller.php';

use Restserver\Libraries\REST_Controller;

// login api
class Authentication extends REST_Controller {

    function __construct()
    {
        // Construct the parent class
        parent::__construct();
        $this->load->model('AuthenticationModel');
    }

    // register user
    public function register_post()
    {
        $username = $this->post('username');
        $password = $this->post('password');
        $email = $this->post('email');
        $result = $this->AuthenticationModel->register($username, $password,
$email);
        if ($result) {
            $this->response($result, REST_Controller::HTTP_OK);
        } else {
            $this->response([
                'status' => FALSE,
                'message' => 'Username or email already exists'
            ], REST_Controller::HTTP_NOT_FOUND);
        }
    }

    public function login_post()
    {
        $username = $this->post('username');
        $password = $this->post('password');
        $result = $this->AuthenticationModel->login($username, $password);
        if ($result) {
            // $jwt = new JWT();
            // $key = $this->config->item('jwt_key_HELOO_SECRET');
            // $iat = time();
            // $exp = $iat + 60;
            // $payload = array(
            //     "iss" => "Issuer",
            //     "aud" => "Audience",
            //     "iat" => $iat,
            //     "nbf" => $iat + 10,
            //     "exp" => $exp,
            //     "username" => $username);

            // $token = $jwt->encode($payload, $key, 'HS256');
            $this->response($result, REST_Controller::HTTP_OK);
        } else {
            $this->response([
                'status' => FALSE,
                'message' => 'Invalid username or password'
            ], REST_Controller::HTTP_NOT_FOUND);
        }
    }
}
```

2.11. Post.php

path – controllers/api/Post.php

```
<?php

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

// This can be removed if you use __autoload() in config.php OR use Modular
Extensions
/** @noinspection PhpIncludeInspection */
require APPPATH . '/libraries/REST_Controller.php';

use Restserver\Libraries\REST_Controller;

// login api
class Post extends REST_Controller {

        function __construct()
        {
            // Construct the parent class
            parent::__construct();
            $this->load->model('PostModel');
            $this->upload_path = './postImages/';
        }

        public function index_get()
        {
            $id = $this->uri->segment(3);
            if ($id === NULL) {
                $result = $this->PostModel->getPosts();
                if ($result) {
                    $this->response($result, REST_Controller::HTTP_OK);
                } else {
                    $this->response([
                        'status' => FALSE,
                        'message' => 'No posts found'
                    ], REST_Controller::HTTP_NOT_FOUND);
                }
            } else {
                $result = $this->PostModel->getPost($id);
                if ($result) {
                    $this->response($result, REST_Controller::HTTP_OK);
                } else {
                    $this->response([
                        'status' => FALSE,
                        'message' => 'No post found'
                    ], REST_Controller::HTTP_NOT_FOUND);
                }
            }
        }

        public function index_post()
        {
            $caption = $this->post('caption');
            $user_id = $this->post('userID');

            // upload post as image
            $config['upload_path'] = './postImages/';
            $config['allowed_types'] = 'gif|jpg|png';
            $config['max_size'] = 10000;
            $config['max_width'] = 10240;
            $config['max_height'] = 7680;
            $config['encrypt_name'] = TRUE;
            $this->load->library('upload', $config);
            if (!$this->upload->do_upload('image')) {
                $error = array('error' => $this->upload->display_errors());
                $this->response($error, REST_Controller::HTTP_NOT_FOUND);
            } else {
                $data = array('upload_data' => $this->upload->data());
                $post = $data['upload_data']['file_name'];
                var_dump($post);
            }

            if ($post == null || $caption == null || $user_id == null) {
                $this->response([
                    'status' => FALSE,
                    'message' => 'Please fill all fields'
                ], REST_Controller::HTTP_NOT_FOUND);
            }else{

                $result = $this->PostModel->createPost($post, $caption,
$user_id);
                if ($result) {
                    $this->response($result, REST_Controller::HTTP_OK);
                } else {
                    $this->response([
                        'status' => FALSE,
                        'message' => 'Failed to create post'
                    ], REST_Controller::HTTP_NOT_FOUND);
                }
            }
        }

        public function index_put()
        {
            $id = $this->put('id');
            $title = $this->put('title');
            $content = $this->put('content');
            $result = $this->PostModel->updatePost($id, $title, $content);
            if ($result) {
                $this->response($result, REST_Controller::HTTP_OK);
            } else {
                $this->response([
                    'status' => FALSE,
                    'message' => 'Failed to update post'
                ], REST_Controller::HTTP_NOT_FOUND);
            }
        }

        public function index_delete()
        {
            $id = $this->uri->segment(3);
            var_dump($id);
            $result = $this->PostModel->deletePost($id);
            if ($result) {
                $this->response($result, REST_Controller::HTTP_OK);
            } else {
                $this->response([
                    'status' => FALSE,
                    'message' => 'Failed to delete post'
                ], REST_Controller::HTTP_NOT_FOUND);
            }
        }
}
```

2.12. Landing.php

path – controllers/Landing.php

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

class Landing extends CI_Controller {

    public function index()
    {
        $this->load->view('pages/landing_view');
    }
}
```

2.13. LogIn.php

path – controllers/LogIn.php

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

class LogIn extends CI_Controller {

    public function index()
    {
        $this->load->view('pages/login_view');
    }
}
```

2.14. SignUp.php

path – controllers/SignUp.php

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

class SignUp extends CI_Controller {

    public function index()
    {
        $this->load->view('pages/signup_view');
    }
}
```

2.15. Upload.php

path – controllers/Upload.php

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

class Upload extends CI_Controller {

    public function index()
    {
        $this->load->view('pages/upload_view');
    }
}
```

3 .EVALUATION OF DESIGN FROM CW1

3.1. Changes made in the Controller

In course work 1 I had mentioned as I’m using only two controller named “ApiController” and “AuthController”. Now I realize that using it in this manner is inefficient. As a result, I put individual controllers for each task within a subfolder called "api" where the Controllers call the APIs.

- Authentication
- Post
- Comment

Besides this, I included few Controllers to call the views in the root folder.

- Explore.php
- Landing.php
- Login.php
- Reat_server.php
- SignUp.php
- Upload.php

Because there are so many Controllers, each one has the appropriate post, get, put, and delete methods.

The Authentication controller has only one method for checking user availability and determining if it is a login or a register operation.

3.2. Changes made in the Model

To execute each operation, PostModel contains 'insertPost,' 'getPost,' 'getPosts,' 'updatePost,' and 'deletePost' methods. Because the post is being searched using tags, 'SearchPost' was removed.

AuthenticationModel contains ‘verifyUser’, ‘registerUser’, ‘updateUseer’ and ‘loginUser’ methods to perform its actions methodically.

FollowerModel is removed due to time constraint as it is not an essential feature of the application.

Some names of the methods are refactored.
‘getComment’ in CommentModel was included to retrieve the comments from the database.

3.3. Changes in the Views

The view pages are the same as previously indicated. The post explore page has undergone minor adjustments. Posts are initially shown as cards. When they are clicked, only the information about the postings is displayed.

3.4. Changes in the Database Schema

Because of the time limitation, I removed the 'Following' table from the database schema. I eliminated the session’s table because I'm using JWT tokenization.

3.5. Other changes

Aside from the database, a local folder named 'uploads' was created to keep the photos of the posts and other associated images. The database maintains the picture names, and the application obtains the images from the local folder by mapping them to the database image names.

3.6. Conclusion

Creating the application early on was quite beneficial because it provided me with a clear picture of how I intend to carry out the project. There were minimal changes made to maintain the program's efficiency and accessibility during the development process.

Instead of attempting to plan everything out in advance, I would have used a more agile technique and made modifications and alterations as needed if I had to start over. Maintaining high-level security features and authentications would necessitate more thought than the existing efficient solution. I would also take user feedback into account and make any necessary changes.

4 . Discussion

Q1: In your own words, give three reasons the author gives for not using frameworks. What are the arguments he gives to support these reasons?

The post "Why you shouldn't use a web framework" cites the following three arguments for not using web frameworks:

1. Unnecessary complexity and bloat:
   According to the author, web frameworks can add unneeded complexity and bloat to a project, making it more difficult to comprehend and maintain. He supports this claim by claiming that web frameworks frequently incorporate features and capabilities that are not required for a specific project, making the codebase bulkier and more difficult to browse.

2. Difficulty in troubleshooting and understanding difficulties:
   According to the author, web frameworks can make it more difficult to comprehend and troubleshoot issues in a project because the framework's coding can obfuscate the underlying problem. He supports this claim by claiming that online frameworks frequently abstract away the low-level elements of web development, making it more difficult to understand what is going on underneath the hood.

3. Limiting performance:
   According to the author, web frameworks might limit the potential to optimize performance because they frequently enforce a structure and set of rules that may not be the most effective. He backs up this claim by claiming that web frameworks might make it more difficult to maximize speed by introducing more layers of abstraction and their own performance overhead.

While the article gives arguments against using web frameworks, it also notes that web frameworks can provide benefits such as saving development time and giving a clear structure for the code. The author, on the other hand, argues that the decision to adopt a web framework should be based on the project's specific needs as well as the experience and preferences of the people working on it.

Q2: In your view, what do you think is the most important reason the author gives against using frameworks? Do you agree or disagree with the argument the author gives in support of this reason?

The post https://dev.to/gypsydave5/why-you-shouldnt-use-a-web-framework-3g24 explains why the author believes that utilizing a web framework isn't always the greatest option for a project. The most major rationale cited is that adopting a web framework might add unneeded complexity and bloat to a project, and that for smaller projects, it may be more useful to utilize a micro-framework or write vanilla code.

The author contends that utilizing a framework limits the developer's understanding of the underlying technology and makes debugging and optimizing performance more difficult. They further contend that frameworks frequently include extraneous functionality and abstractions that make the software more difficult to understand and maintain. Furthermore, the author claims that frameworks can make it more difficult to optimize performance because developers may not grasp how the framework is built.

I agree that utilizing a full-featured framework for small projects might add unneeded complexity and make the software more difficult to understand and maintain. However, I believe that for larger and more sophisticated projects, a well-designed framework can bring considerable benefits in terms of code structure, security, and performance. The usage of frameworks can help speed up the development process and allow developers to focus on solving business challenges rather than infrastructure.

Frameworks have the advantage of providing a structure for organizing code and implementing common functions such as routing, database access, and security. This can help to improve the codebase's maintainability and scalability while also lowering the chance of security issues. Furthermore, frameworks frequently include built-in support for typical web development tasks like form handling, database access, and user authentication, which can save developers a significant amount of time and effort.

Furthermore, frameworks are frequently produced and maintained by a developer community, which implies they are well-tested and have a huge user base. This can help to ensure that the framework is stable and has a long lifespan, as well as that defects and security vulnerabilities are detected and corrected as soon as possible.

In conclusion, whether or not to employ a web framework is determined by the size and complexity of the project, as well as the talents and preferences of the people working on it. While frameworks can add unneeded complexity and bloat to smaller projects, the benefits of a well-designed framework can be enormous for larger and more complex enterprises. A framework can help to organize code, incorporate common functionality, and speed up the development process. It's also vital to remember that developers should be familiar with the underlying technology and be able to understand and troubleshoot the software, regardless of whether a framework is utilized or not.

5.References

Why You Shouldn’t Use A Web Framework. (no date). *DEV Community* 👩‍💻👨‍💻. Available from https://dev.to/gypsydave5/why-you-shouldnt-use-a-web-framework-3g24 [Accessed 19 January 2023].

Why You Shouldn’t Use A Web Framework. (no date). *DEV Community* 👩‍💻👨‍💻. Available from https://dev.to/gypsydave5/why-you-shouldnt-use-a-web-framework-3g24 [Accessed 19 January 2023].