UE21CS341A-SoftwareEngineering

Deliverable 2

“Online Motorcycle Bike Rental System”

“Implementation Details Document”

Team:

Preetham V Divakara --PES1UG22CS448

Prithvi Raj B L --PES1UG22CS451

Implementation Details :

Introduction:

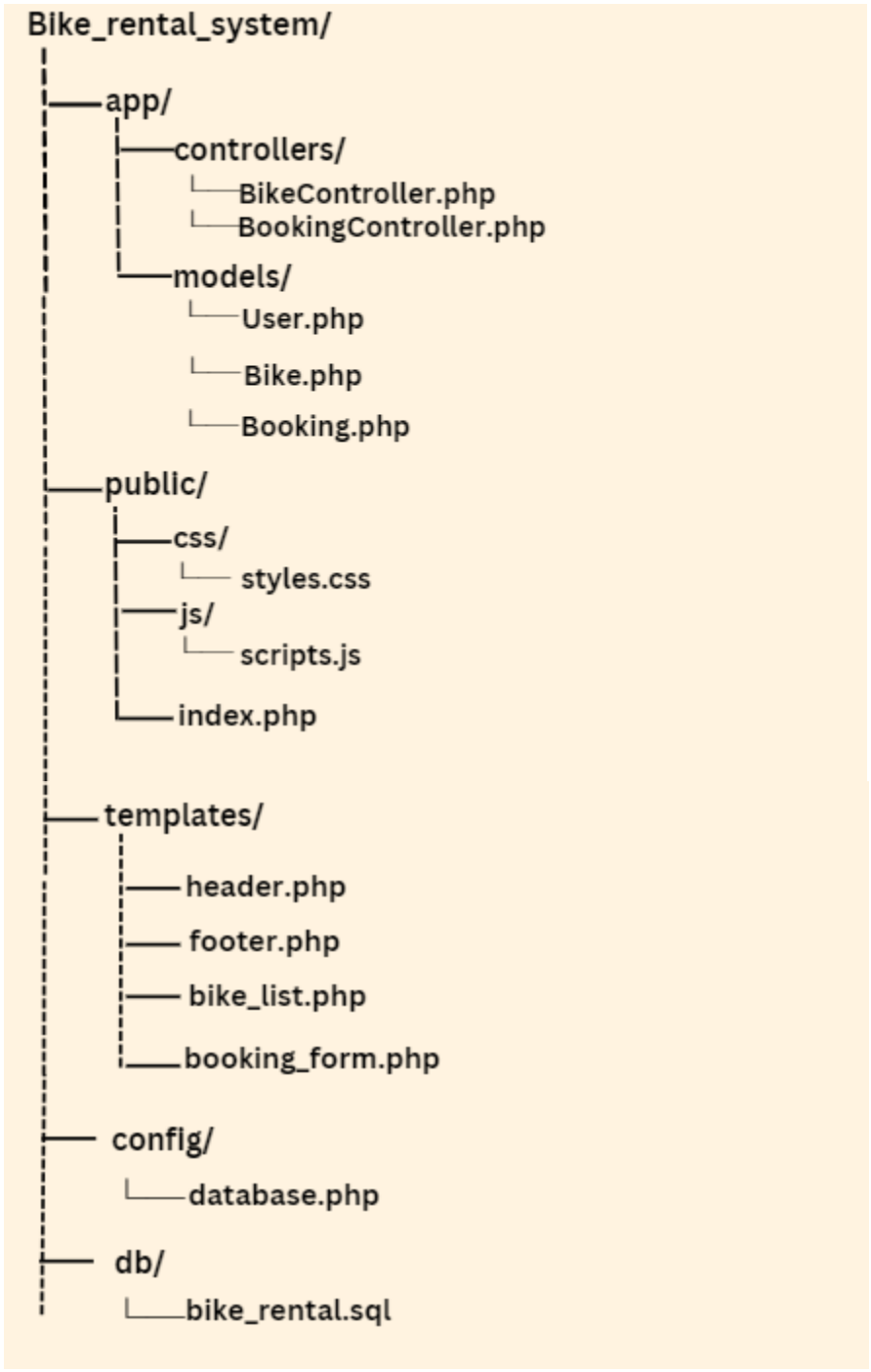
This code document provides an overview of the structure, dependencies, and key modules of the **Online Motorcycle Bike Rental System**. It explains the organization of files and directories, major code functionalities, and key components that contribute to the development of this system.

Dependencies:

The Bike Rental System is built using the following technologies:

- **PHP**: Used for server-side logic and handling user requests.
- **MySQL**: Database management for storing user, bike, and booking data.
- **HTML/CSS/JavaScript**: Front-end technologies for designing the user interface.
- **Apache**: Web server for running the application.

Project Structure:



Main Application File:

index.php:

The index.php file in the public/ directory is the entry point for the web application. It initializes the app and loads necessary components such as controllers, views, and models.

```
<?php

require_once '../config/database.php';

require_once '../app/controllers/BikeController.php';

require_once '../app/controllers/BookingController.php';


if ($_SERVER['REQUEST_METHOD'] === 'GET') {

    $controller = new BikeController();

    $controller->showBikes();

} elseif ($_SERVER['REQUEST_METHOD'] === 'POST') {

    $controller = new BookingController();

    $controller->bookBike($_POST);

}

?>
```

Database Configuration:

config/database.php:

This file contains the configuration for connecting to the MySQL database.

```php
<?php

$host = 'localhost';
$db = 'bike_rental';
$user = 'root';
$pass = '';

try {

    $pdo = new PDO("mysql:host=$host;dbname=$db", $user, $pass);

    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

} catch (PDOException $e) {

    echo 'Connection failed: ' . $e->getMessage();
}
?>
```

Controller:

BikeController.php:

The BikeController handles user requests related to bike listings, such as showing available bikes and displaying bike details.

```php
<?php

class BikeController {

    public function showBikes() {

        // Fetch bike data from the database
        $pdo = $this->getDbConnection();

        $stmt = $pdo->query('SELECT * FROM bikes WHERE available = 1');

        $bikes = $stmt->fetchAll();

        // Load the bike listing view
        require '../templates/bike_list.php';
    }

    private function getDbConnection() {
```

```
        require '../config/database.php';

        return $pdo;
    }
}
?>
```

Model:

Bike.php:

The Bike model represents the bike entity and handles data interactions such as retrieving bike information or updating availability.

```
<?php
class Bike {
    public function getAllBikes() {
        global $pdo;

        $query = 'SELECT * FROM bikes';

        return $pdo->query($query)->fetchAll();
    }

    public function updateAvailability($bikeId, $status) {
        global $pdo;
        $query = 'UPDATE bikes SET available = ? WHERE id = ?';

        $stmt = $pdo->prepare($query);

        return $stmt->execute([$status, $bikeId]);
    }
}
?>
```

View:

bike_list.php:

This view displays the list of available bikes for users to browse. It fetches data from the BikeController and presents it in a user-friendly format.

```
<?php require 'header.php'; ?>
<h1>Available Bikes</h1>
<ul>
    <?php foreach ($bikes as $bike): ?>
        <li>
            <h3><?php echo $bike['model']; ?></h3>

            <p>Price per day: <?php echo $bike['price_per_day']; ?> USD</p>

            <a href="booking_form.php?bike_id=<?php echo $bike['id']; ?>">Book Now</a>
        </li>
    <?php endforeach; ?>
</ul>
<?php require 'footer.php'; ?>
```

Booking Functionality:

BookingController.php:

The BookingController handles the booking process. When a user submits the booking form, this controller manages the booking request and updates the database accordingly.

```
<?php

class BookingController {

    public function bookBike($postData) {

        $pdo = $this->getDbConnection();

        $stmt = $pdo->prepare('INSERT INTO bookings (user_id, bike_id, rental_start, rental_end)
VALUES (?, ?, ?, ?)');



        $stmt->execute([$postData['user_id'], $postData['bike_id'], $postData['rental_start'],
$postData['rental_end']]);



        // Redirect to a confirmation page

        header('Location: booking_confirmation.php');
```

```
    }

    private function getDbConnection() {
        require '../config/database.php';
        return $pdo;
    }
}
?>
```

Static Files:

styles.css and scripts.js:

- **styles.css**: Contains the CSS for styling the front-end components like the bike list and booking form.
- **scripts.js**: JavaScript functionalities for validating forms or enhancing the user experience (e.g., date picker for rental dates).

SQL File:

bike_rental.sql:

This SQL file would contain the SQL commands to create your database tables and insert any necessary default data.

```
CREATE DATABASE IF NOT EXISTS bike_rental;

USE bike_rental;

-- Table for storing user information

CREATE TABLE users (
```

```
    user_id INT AUTO_INCREMENT PRIMARY KEY,
    username VARCHAR(50) NOT NULL,
    email VARCHAR(100) UNIQUE NOT NULL,
    password VARCHAR(255) NOT NULL
);
-- Table for storing bike information
CREATE TABLE bikes (
    bike_id INT AUTO_INCREMENT PRIMARY KEY,
    model VARCHAR(100) NOT NULL,
    brand VARCHAR(50) NOT NULL,
    price_per_day DECIMAL(10, 2) NOT NULL,
    available BOOLEAN DEFAULT TRUE
);
-- Table for managing bookings
CREATE TABLE bookings (
    booking_id INT AUTO_INCREMENT PRIMARY KEY,
    user_id INT,
    bike_id INT,
    rental_start DATE,
    rental_end DATE,
    FOREIGN KEY (user_id) REFERENCES users(user_id),
    FOREIGN KEY (bike_id) REFERENCES bikes(bike_id)
);
```