



PROJECT MVC MODEL-VIEW-CONTROLLER



ANGGOTA KELOMPOK

1. 22081070100 / Rafli Afriza Nugraha
2. 22081070100 / Glen Hakim
3. 2208107010090 / T. Farhan Syah
4. 22081070100 / T.M. Fadlul Ihsan
5. 2208107010026 / Indriani Miza Alfiyanti
6. 2409103622220021 / Saif Syahsyah
Hizbullah



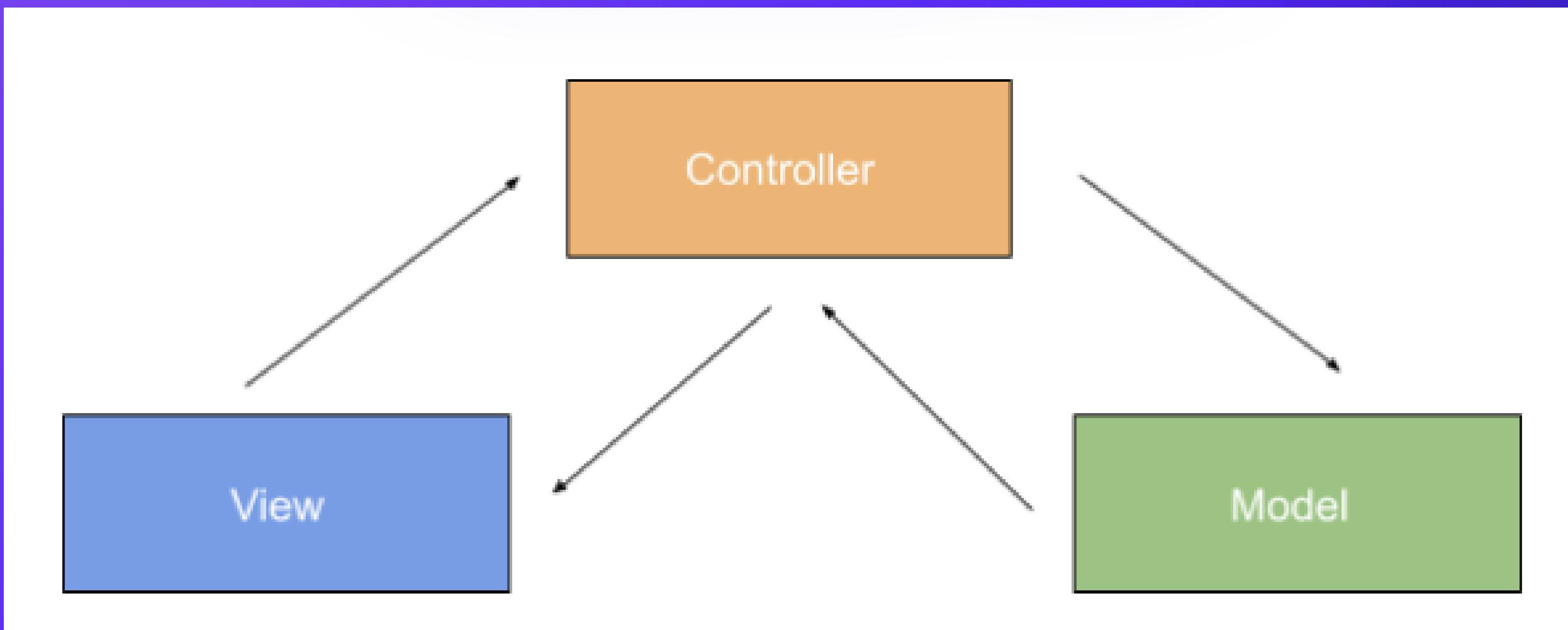
DEFINISI MVC

MVC atau *Model View Controller* adalah sebuah pola desain arsitektur dalam sistem pengembangan website yang terdiri dari tiga bagian, yaitu:

- Model, bagian yang mengelola dan berhubungan langsung dengan database;
- View, bagian yang akan menyajikan tampilan informasi kepada pengguna;
- Controller, bagian yang menghubungkan model dan view dalam setiap proses request dari user.



ILUSTRASI MVC



PROJECT 1

BANK LOCKER MANAGEMENT SYSTEM USING PHP AND MYSQL

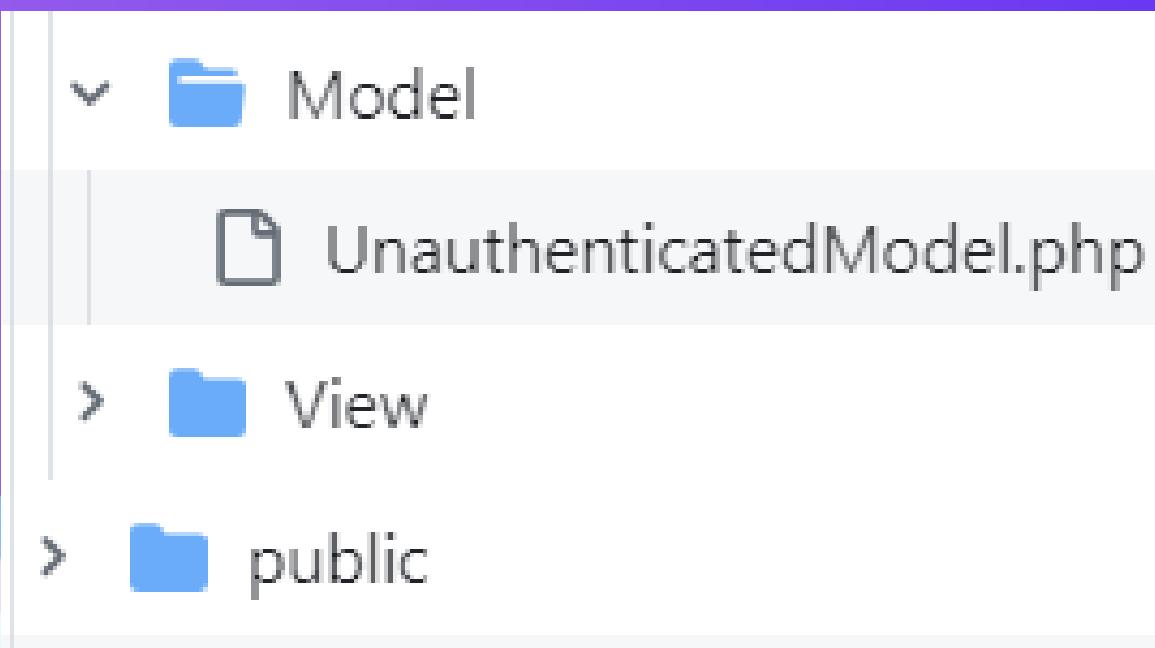
SISTEM MANAJEMEN LOKER BANK INI BERTUJUAN UNTUK MENGELOLA PENGGUNAAN LOKER SECARA EFISIEN DAN AMAN. PADA SISTEMINI PENGGUNA HANYA MEMPUNYAI AKSES UNTUK MENGHUBUNGI ADMIN DAN MENCARI KODE LOKER YANG TELAH DIBERIKAN KEPADA MEREKA. SISTEM PENGELOLAAN LOKER BANK INI DIRANCANG AGAR PENGGUNA CUKUP MENGHUBUNGI ADMIN DAN MENCARI KODE LOKERNYA. HANYA ADMIN YANG MEMPUNYAI IZIN UNTUK MENDAFTARKAN LOKER BARU. INTINYA, SISTEMINI DIGUNAKAN UNTUK MENGELOLA LOKER DI BANK.

DEFINISI MODEL

Dalam konteks pola desain MVC, "Model" mengacu pada bagian dari aplikasi yang bertanggung jawab atas pengelolaan data, logika bisnis, dan aturan aplikasi. Model dapat merujuk kepada konsep atau representasi abstrak dari suatu sistem, objek, atau fenomena di dunia nyata.



SOURCE CODE



```
1  <?php
2
3  namespace Kelompok1\MVC\Model;
4
5  use Kelompok1\MVC\Database\Connection;
6
7  class UnauthenticatedModel {
8      private $conn;
9
10     public function __construct()
11     {
12         $db = new Connection();
13         $this->conn = $db->getConnection();
14     }
15
16     public function getAllAboutUsData(){
17         $query = "SELECT * FROM tblpage where PageType='aboutus' ";
18         $result = $this->conn->query($query);
19
20         return $result->fetch_all(MYSQLI_ASSOC);
21     }
22
23     public function getAllContactUsData(){
24         $query = "SELECT * FROM tblpage where PageType='contactus' ";
25         $result = $this->conn->query($query);
26
27         return $result->fetch_all(MYSQLI_ASSOC);
28     }
29
30     public function getSearchLocker($searchby){
31         $query = "select * from tblassignlocker where LockerNumber='$searchby' || KeyNumber='$searchby' ";
32         $result = $this->conn->query($query);
33
34         return $result->fetch_all(MYSQLI_ASSOC);
```

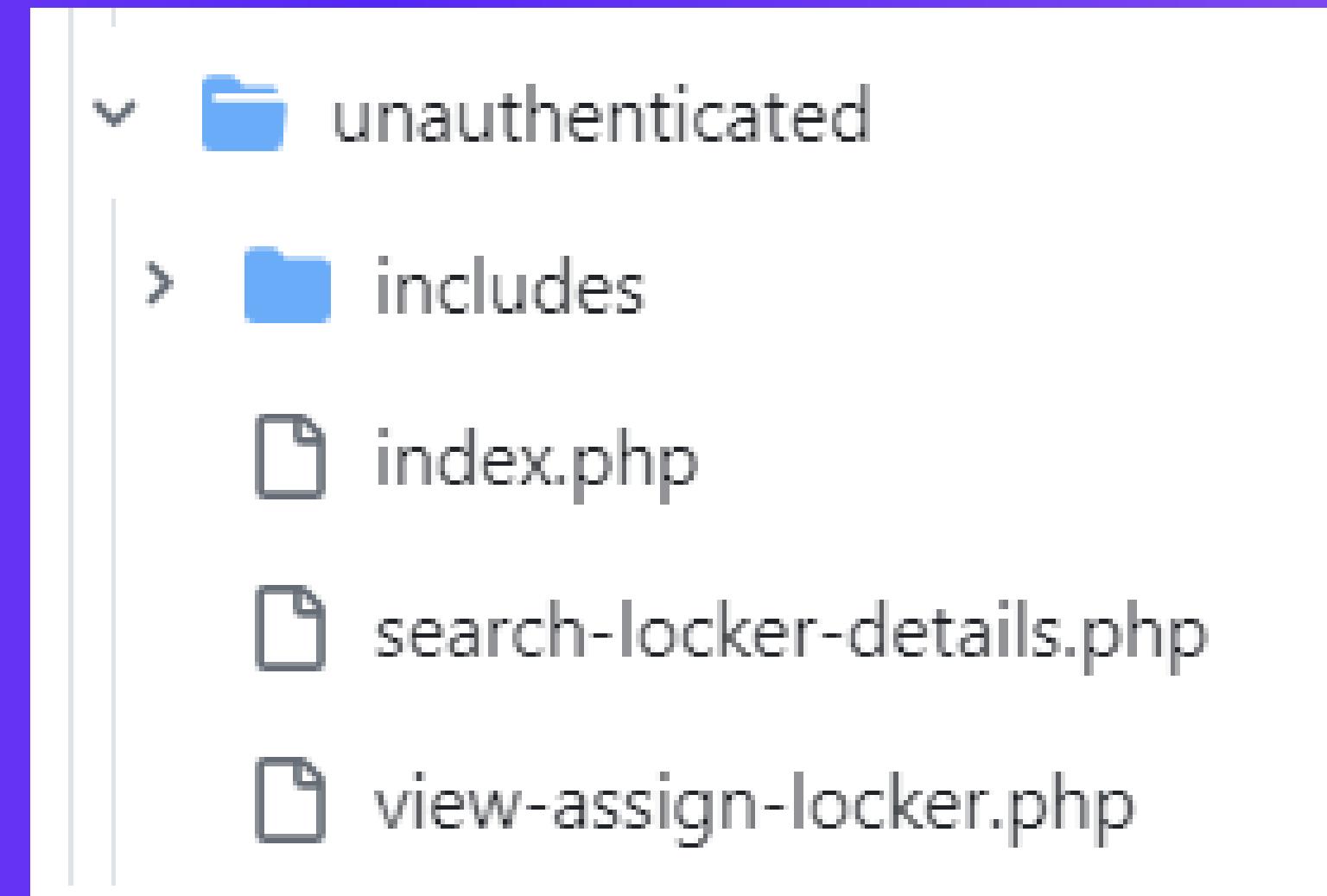
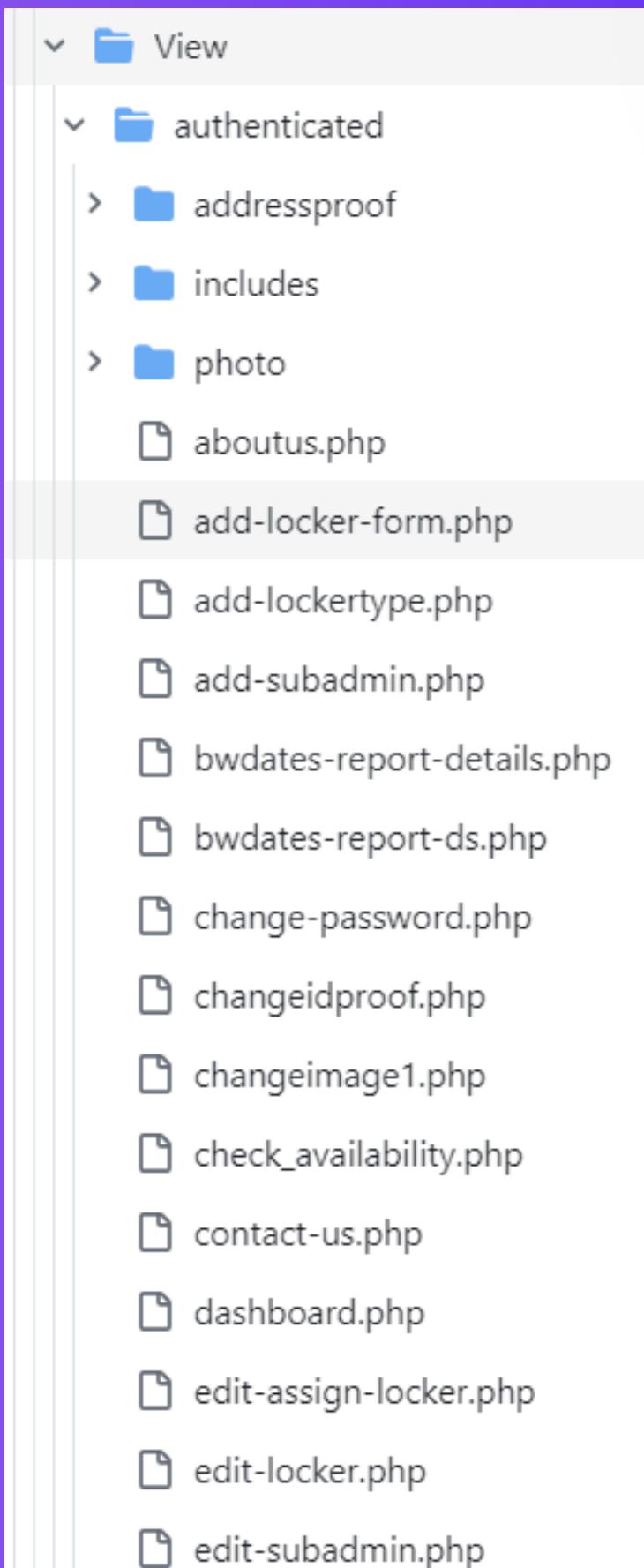


DEFINISI VIEW

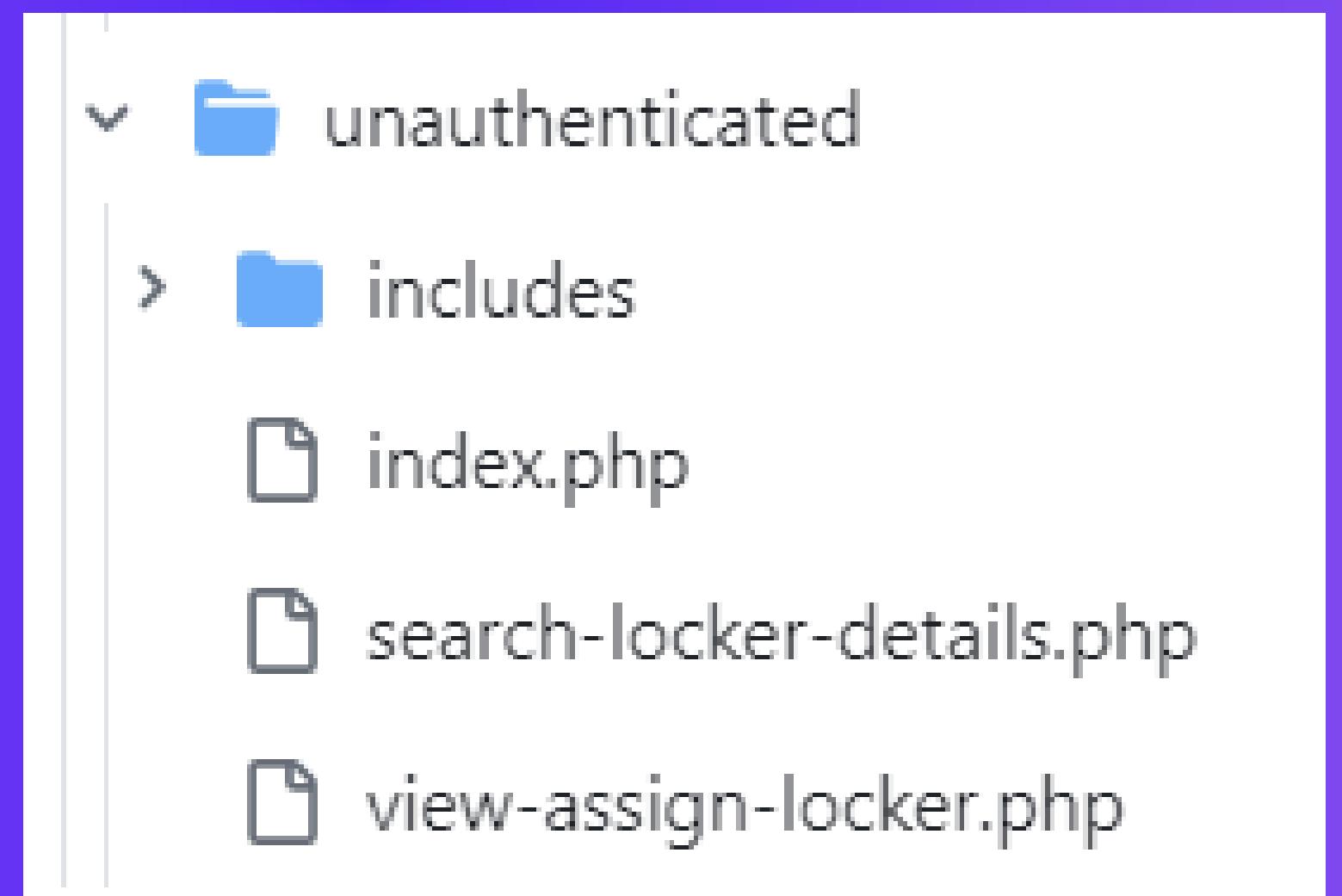
Dalam konteks pengembangan perangkat lunak, terutama ketika menggunakan pola desain Model-View-Controller (MVC), "View" merujuk kepada bagian dari aplikasi yang bertanggung jawab untuk menampilkan informasi kepada pengguna dan menerima input dari mereka.

View adalah komponen penting dalam pola desain MVC yang bertanggung jawab atas presentasi data kepada pengguna dan memungkinkan interaksi antara pengguna dan aplikasi.

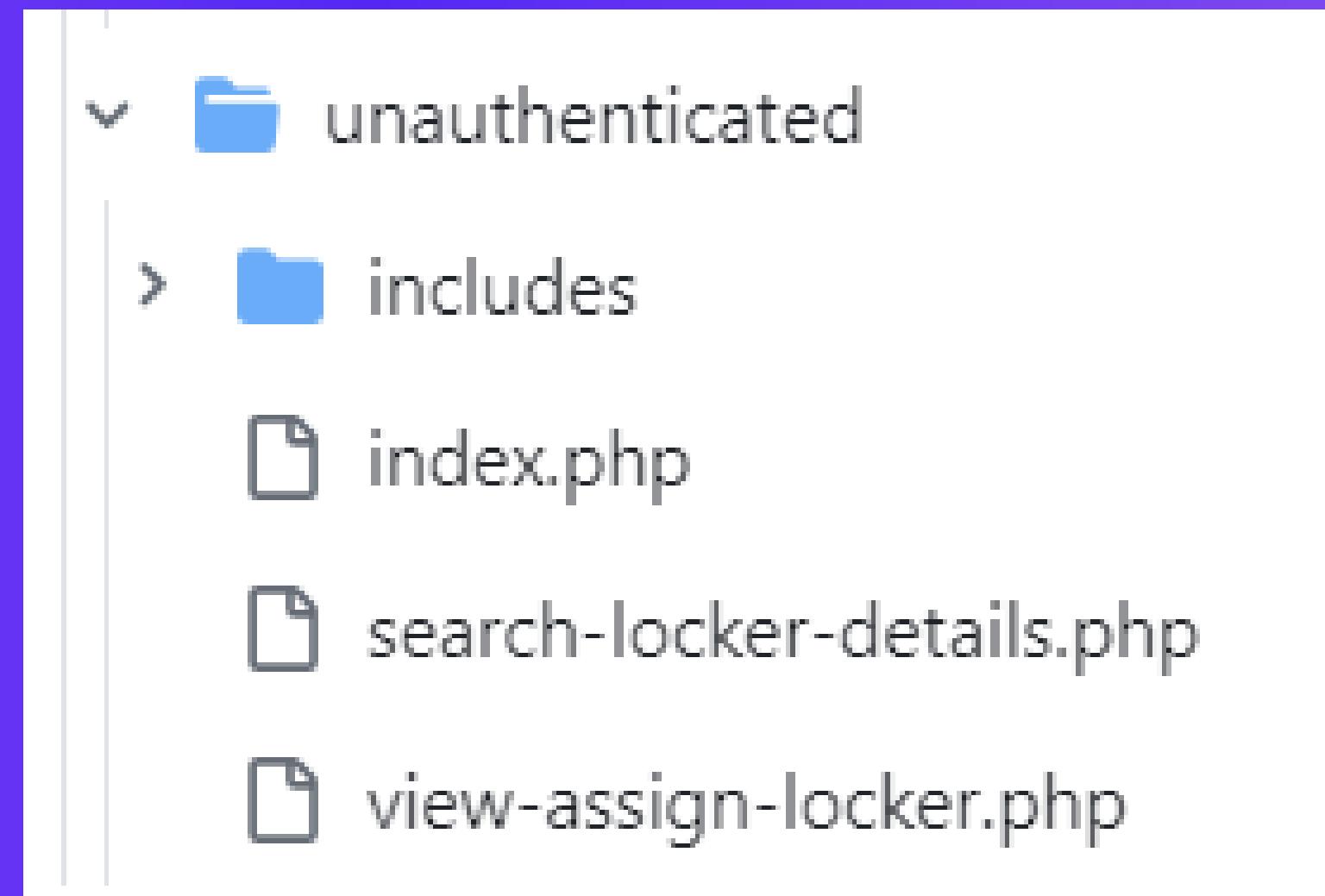
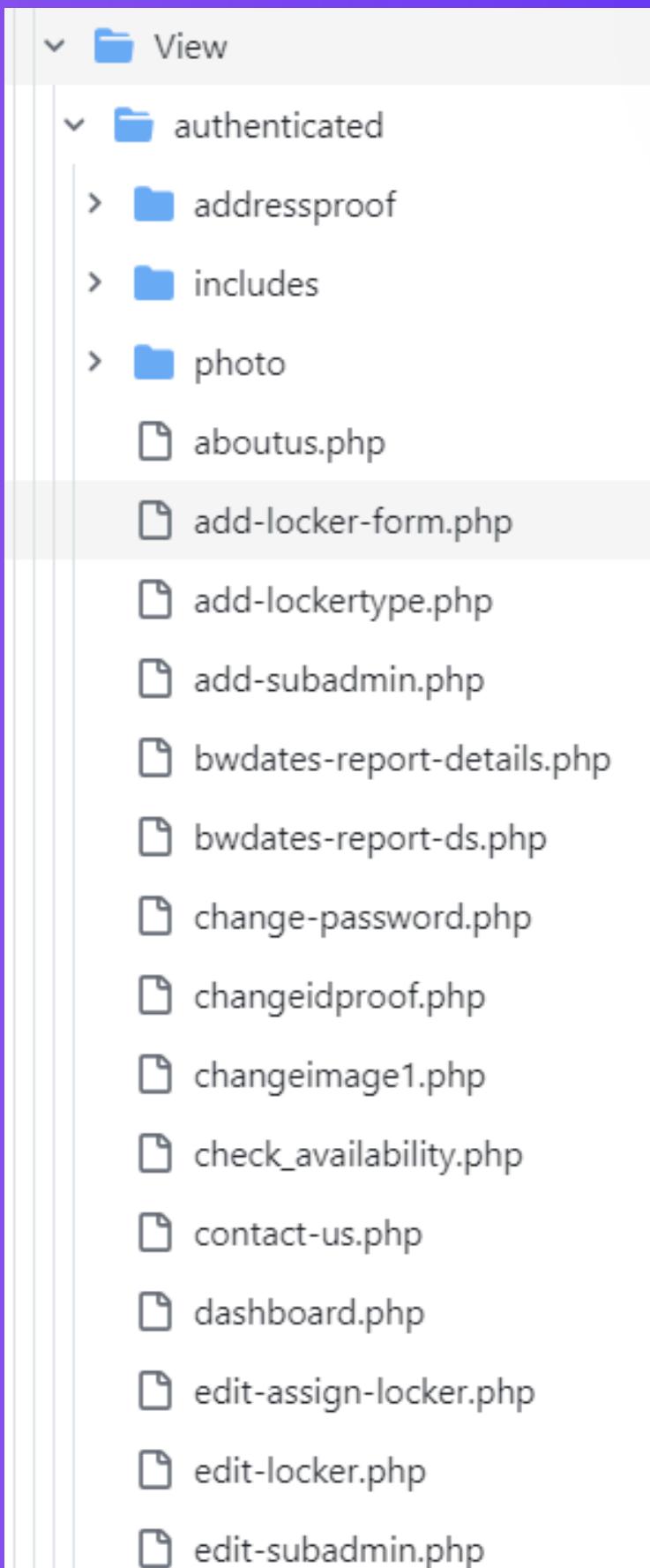
IMPLEMENTASI VIEW



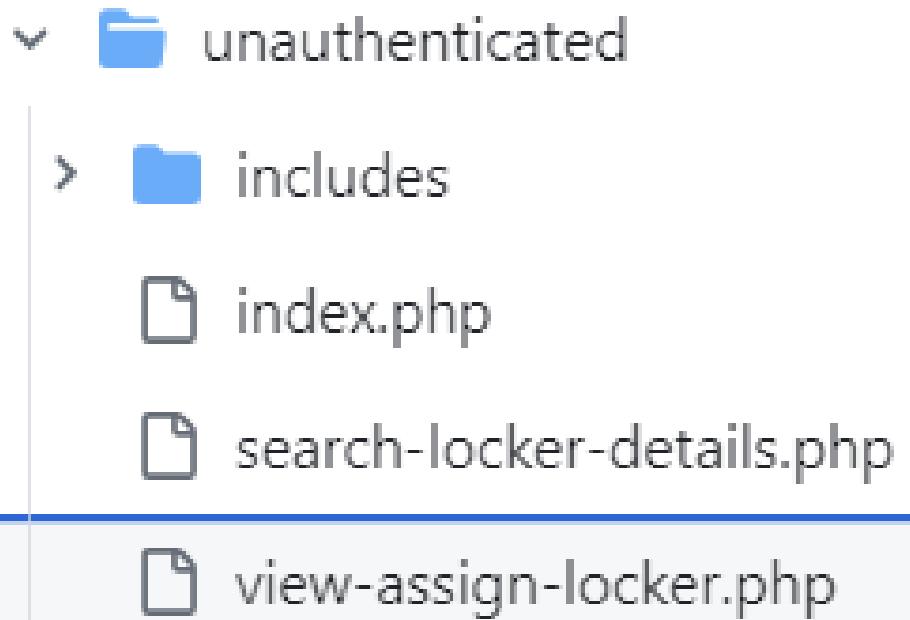
IMPLEMENTASI VIEW



IMPLEMENTASI VIEW



IMPLEMENTASI VIEW



```
1  <?php
2  include('banker/includes/config.php');
3  session_start();
4  error_reporting(0);
5
6  ?>
7
8  <!DOCTYPE HTML>
9  <html lang="zxx">
10 <head>
11     <title>Bank Locker Management System | Home</title>
12     <script type="application/x-javascript">
13         addEventListener("load", function() {
14             setTimeout(hideURLbar, 0);
15         }, false);
16
17         function hideURLbar() {
18             window.scrollTo(0, 1);
19         }
20     </script>
21     <!-- Meta tag Keywords -->
22     <!-- css files -->
23     <link href="css1/bootstrap.css" rel="stylesheet" type="text/css" media="all" /> <!-- Bootstrap-Core-CSS -->
24     <link href="css1/style.css" rel="stylesheet" type="text/css" media="all" /> <!-- Style-CSS -->
25     <link rel="stylesheet" href="css1/font-awesome.css"> <!-- Font-Awesome-Icons-CSS -->
26     <link rel="stylesheet" href="css1/flexslider.css" type="text/css" media="all" /> <!-- Banner-Slider-CSS -->
27     <!-- //css files -->
28     <link rel="stylesheet" type="text/css" href="css1/demo.css" />
29     <!-- online-fonts -->
30     <link href="//fonts.googleapis.com/css?family=Dosis:200,300,400,500,600,700,800&subset=latin-ext" rel="stylesheet">
31     <link href="//fonts.googleapis.com/css?family=Ubuntu+Condensed&subset=cyrillic,cyrillic-ext,greek,greek-ext,latin-ext" rel="stylesheet">
```

DEFINISI CONTROLLER

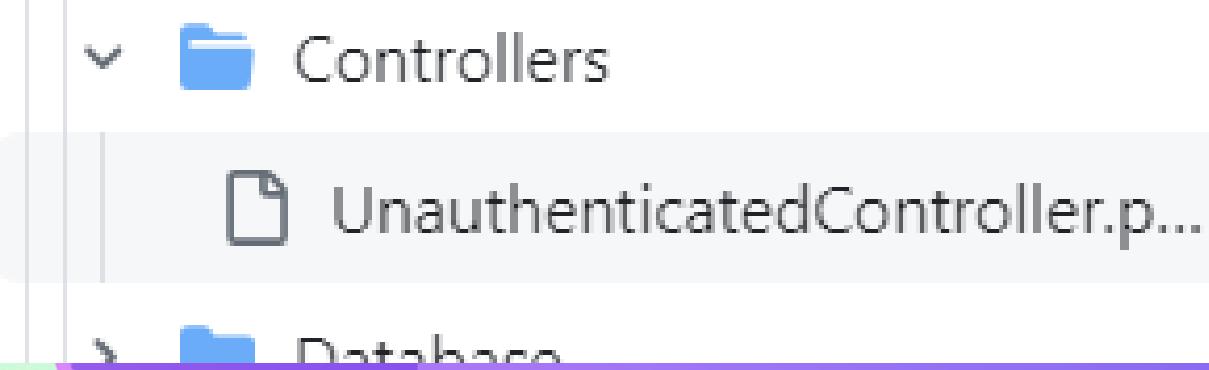


Dalam konteks pola desain Model-View-Controller (MVC), "Controller" merujuk kepada bagian dari aplikasi yang bertanggung jawab untuk menerima input dari pengguna, memprosesnya, dan mengirimkan instruksi kepada Model atau View untuk melakukan tindakan yang sesuai.

Secara keseluruhan, Controller adalah komponen penting dalam pola desain MVC yang bertanggung jawab atas pengelolaan input pengguna, pemrosesan data, dan pengiriman instruksi kepada Model dan View untuk menjalankan logika aplikasi.

SOURCE CODE

```
1  <?php
2
3  namespace Kelompok1\MVC\Controllers;
4
5  use Kelompok1\MVC\App\View;
6  use Kelompok1\MVC\Model\UnauthenticatedModel;
7
8  class UnauthenticatedController
9  {
10     function index(): void
11     {
12         $unauthenticatedModelData = new UnauthenticatedModel();
13         $aboutus = $unauthenticatedModelData->getAllAboutUsData();
14         $contactus = $unauthenticatedModelData->getAllContactUsData();
15
16         include_once __DIR__ . '/../View/unauthenticated/index.php';
17     }
18
19     function searchLocker($searchinput = null): void
20     {
21         $unauthenticatedModelData = new UnauthenticatedModel();
22         $searchResult = $unauthenticatedModelData->getSearchLocker($searchinput);
23         $cnt = 1;
24         $count = count($searchResult);
25
26
27         include_once __DIR__ . '/../View/unauthenticated/search-locker-details.php';
28     }
29
30     function showLockerDetail($ltid = null): void
31     {
```



PROJECT 2

DIRECTORY MANAGEMENT SYSTEM USING PHP AND MYSQL

SISTEM MANAJEMEN DIREKTORI INI DIRANCANG KHUSUS UNTUK MENGELOLA KARTU NAMA SECARA EFISIEN. PADA SISTEM INI PENGGUNA HANYA DIBERIKAN AKSES UNTUK MELIHAT DAN MENCARI KARTU NAMA YANG SUDAH ADA PADA DATABASE. SISTEM MANAJEMEN DIREKTORI INI DIRANCANG UNTUK MENGELOLA KARTU NAMA. PENGGUNA HANYA DAPAT MELIHAT DAN MENCARI KARTU NAMA YANG TERSEDIA, SEDANGKAN ADMIN MEMPUNYAI KEMAMPUAN UNTUK MENAMBAH, MELIHAT, MENGEDIT DAN MENGHAPUS KARTU NAMA. SISTEM INI MEMASTIKAN INFORMASI KARTU NAMA DIKELOLA DENGAN BAIK DAN SELALU DIPERBARUI SESUAI KEBUTUHAN.

DEFINISI MODEL

Model dalam Directory Management System bertanggung jawab atas pengelolaan dan pemrosesan data yang terkait dengan direktori atau entitas-entitas dalam sistem. Ini mencakup operasi-operasi seperti penyimpanan data tentang pengguna, direktori, file, dan metadata terkait dalam database MySQL, serta pelaksanaan logika bisnis yang diperlukan untuk operasi-operasi tersebut menggunakan PHP.



SOURCE CODE

```
1  <?php
2
3  namespace App\Models;
4
5  require_once __DIR__ . '../../../../../config/database.php';
6
7  class Directory {
8      private $con;
9
10     public function __construct() {
11         $this->con = $GLOBALS['con'];
12     }
13
14     public function searchDirectories($sdata) {
15         $query = "SELECT * FROM tbldirectory WHERE (FullName LIKE ? OR MobileNumber LIKE ?) AND (Status=1) ORDER BY id DESC LIMIT 12";
16         $stmt = $this->con->prepare($query);
17         $searchTerm = $sdata . '%';
18         $stmt->bind_param('ss', $searchTerm, $searchTerm);
19         $stmt->execute();
20         return $stmt->get_result();
21     }
22 }
```

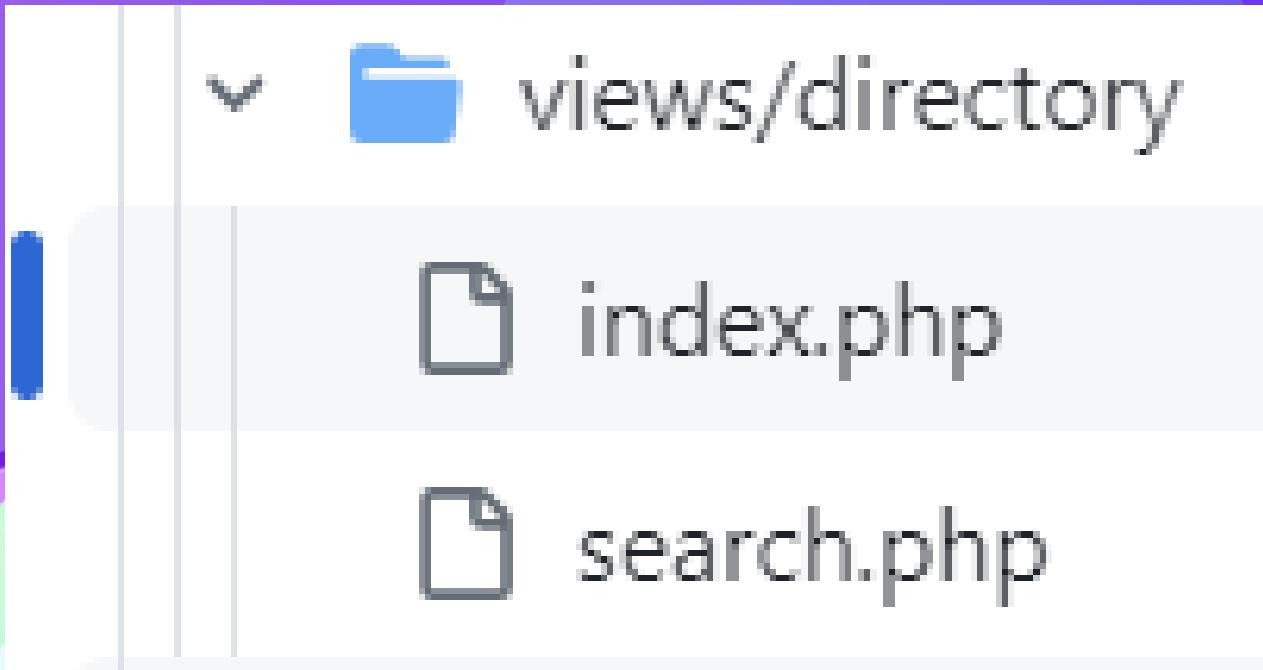


DEFINISI VIEW

Dalam konteks "Directory Management System using PHP and MySQL", View merujuk kepada bagian dari aplikasi yang bertanggung jawab untuk menampilkan informasi kepada pengguna terkait direktori dan file yang tersimpan dalam sistem.

View dalam Directory Management System menggunakan PHP untuk menghasilkan tampilan berbasis web yang memperlihatkan direktori, file, dan informasi terkait kepada pengguna.

SOURCE CODE



```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="utf-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
6      <title>Directory Management System</title>
7      <link href="../../vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">
8      <link href="../../css/heroic-features.css" rel="stylesheet">
9  </head>
10 <body>
11     <?php include __DIR__ . '/../../includes/header.php'; ?>
12
13     <div class="container">
14         <form name="search" method="post" action="searchdata.php">
15             <div class="card my-4">
16                 <h5 class="card-header">Search By Name / Contact Number</h5>
17                 <div class="card-body">
18                     <div class="input-group">
19                         <input type="text" class="form-control" name="searchdata" placeholder="Search for..." required="true">
20                         <span class="input-group-btn">
21                             <button class="btn btn-secondary" type="submit">Go!</button>
22                         </span>
23                     </div>
24                 </div>
25             </div>
26         </form>
27     </div>
28
29     <?php include __DIR__ . '/../../includes/footer.php'; ?>
30     <script src="../../vendor/jquery/jquery.min.js"></script>
31     <script src="../../vendor/bootstrap/js/bootstrap.bundle.min.js"></script>
32 </body>
33 </html>
```

DEFINISI CONTROLLER



Dalam konteks "Directory Management System using PHP and MySQL", Controller merujuk kepada bagian dari aplikasi yang bertanggung jawab atas pengelolaan alur logika dan pengaturan interaksi antara pengguna, tampilan (View), dan data (Model).

Controller dalam Directory Management System menggunakan PHP untuk memproses permintaan pengguna yang diterima melalui antarmuka pengguna (UI) dan mengarahkan aksi yang sesuai ke Model atau View yang terkait.

SOURCE CODE

```
1  <?php
2
3  namespace App\Controllers;
4
5  use App\Models\Directory;
6
7  class DirectoryController {
8      private $model;
9
10     public function __construct() {
11         $this->model = new Directory();
12     }
13
14     public function index() {
15         include __DIR__ . '/../views/directory/index.php';
16     }
17
18     public function search() {
19         if (isset($_POST['searchdata'])) {
20             $sdata = $_POST['searchdata'];
21             $directories = $this->model->searchDirectories($sdata);
22             include __DIR__ . '/../views/directory/search.php';
23         } else {
24             header('Location: /');
25         }
26     }
27 }
```

controllers

DirectoryController.php