Creating an **MVC (Model-View-Controller) application** using **plain PHP and MySQL** involves structuring your project properly and separating concerns. Below is a step-by-step guide to building a basic MVC app.

# Step 1: Create the Folder Structure

Create a project folder, e.g., `php-mvc`, and inside it, organize the following directories:

```
php-mvc/
│── app/
│   ├── controllers/
│   │   ├── HomeController.php
│   ├── models/
│   │   ├── User.php
│   ├── views/
│   │   ├── home.php
│── config/
│   ├── database.php
│── public/
│   ├── index.php
│── core/
│   ├── Controller.php
│   ├── Model.php
│   ├── App.php
│── .htaccess
│── composer.json
```

---

# Step 2: Configure Database Connection (`config/database.php`)

Create a **database connection file**.

```
<?php
class Database {
    private static $host = "localhost";
    private static $dbname = "mvc_app";
    private static $username = "root";
    private static $password = "";
    public static $conn;

    public static function connect() {
        if (!self::$conn) {
            try {
                self::$conn = new PDO("mysql:host=" . self::$host .
";dbname=" . self::$dbname, self::$username, self::$password);
                self::$conn->setAttribute(PDO::ATTR_ERRMODE,
PDO::ERRMODE_EXCEPTION);
            } catch (PDOException $e) {
                die("Database connection failed: " . $e->getMessage());
            }
        }
        return self::$conn;
    }
}
?>
```

---

# Step 3: Set Up the Core Files

### 1. `core/App.php` - Handle URL Routing

```
<?php
class App {
    protected $controller = "HomeController";
    protected $method = "index";
    protected $params = [];

    public function __construct() {
        $url = $this->parseUrl();

        if (file_exists("../app/controllers/" . $url[0] . ".php")) {
            $this->controller = $url[0];
            unset($url[0]);
        }

        require_once "../app/controllers/" . $this->controller . ".php";
        $this->controller = new $this->controller;

        if (isset($url[1]) && method_exists($this->controller, $url[1])) {
            $this->method = $url[1];
            unset($url[1]);
        }

        $this->params = $url ? array_values($url) : [];
        call_user_func_array([$this->controller, $this->method], $this-
>params);
    }

    public function parseUrl() {
        if (isset($_GET['url'])) {
            return explode("/", filter_var(rtrim($_GET['url'], "/"),
FILTER_SANITIZE_URL));
        }
        return ["HomeController"];
    }
}
?>
```

### 2. `core/Controller.php` - Base Controller

```
<?php
class Controller {
    public function view($view, $data = []) {
        require_once "../app/views/" . $view . ".php";
    }

    public function model($model) {
        require_once "../app/models/" . $model . ".php";
        return new $model();
    }
}
?>
```

---

# Step 4: Create a Model

**`app/models/User.php`**

```
<?php
require_once "../config/database.php";

class User {
    private $db;

    public function __construct() {
        $this->db = Database::connect();
    }

    public function getUsers() {
        $stmt = $this->db->prepare("SELECT * FROM users");
        $stmt->execute();
        return $stmt->fetchAll(PDO::FETCH_ASSOC);
    }
}
?>
```

---

# Step 5: Create a Controller

**`app/controllers/HomeController.php`**

```
<?php
require_once "../core/Controller.php";

class HomeController extends Controller {
    public function index() {
        $userModel = $this->model("User");
        $users = $userModel->getUsers();
        $this->view("home", ["users" => $users]);
    }
}
?>
```

---

## Step 6: Create a View

**`app/views/home.php`**

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>MVC App</title>
</head>
<body>
    <h1>Users List</h1>
    <ul>
        <?php foreach ($data["users"] as $user): ?>
            <li><?php echo $user["name"]; ?></li>
        <?php endforeach; ?>
    </ul>
</body>
</html>
```

---

## Step 7: Configure `.htaccess` for Pretty URLs

Create an `.htaccess` file inside the **`public/`** directory.

```
RewriteEngine On
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule ^(.+)$ index.php?url=$1 [QSA,L]
```

---

## Step 8: Set Up `public/index.php`

```
<?php
require_once "../core/App.php";
$app = new App();
?>
```

---

## Step 9: Run the Application

1. Start your local server (XAMPP or PHP built-in server).

2. Navigate to the **`public/`** directory and run:
3. `php -S localhost:8000`
4. Open in the browser:
5. `http://localhost:8000/`

---

# Conclusion

- **Model (`User.php`)** → Handles database interactions.
- **View (`home.php`)** → Displays data.
- **Controller (`HomeController.php`)** → Manages business logic and passes data to the view.
- **App (`core/App.php`)** → Handles URL routing.
- **Link: https://chatgpt.com/c/67d99684-76d0-8005-a183-e46379c2d529**