**🧱 1. Postavi osnovnu strukturu projekta**

bash

CopyEdit

/project-root

├── /app

│ ├── /Controllers

│ ├── /Models

│ └── /Views

├── /core

├── /public

│ └── index.php ← ulazna točka (public root)

├── /config

├── composer.json

**⚙️ 2. Inicijaliziraj Composer i PSR-4 autoload**

U terminalu u rootu:

bash

CopyEdit

composer init

Kad pita ime paketa, možeš staviti npr. ime/projekt.

Zatim **dodaj PSR-4 autoloading ručno** u composer.json:

json

CopyEdit

"autoload": {

"psr-4": {

"App\\": "app/",

"Core\\": "core/"

}

}

Zatim:

bash

CopyEdit

composer dump-autoload

**🧪 3. Napravi testni kontroler**

**app/Controllers/HomeController.php**

php

CopyEdit

<?php

namespace App\Controllers;

class HomeController {

public function index() {

echo "Pozdrav iz HomeController::index!";

}

}

**🚪 4. Postavi index.php kao ulaznu točku**

**public/index.php**

php

CopyEdit

<?php

require\_once \_\_DIR\_\_ . '/../vendor/autoload.php';

use App\Controllers\HomeController;

$controller = new HomeController();

$controller->index();

➡️ Sad možeš pokrenuti server:

bash

CopyEdit

php -S localhost:8000 -t public

I otići na http://localhost:8000 → ispisat će Pozdrav iz HomeController::index!

**🔢 5. Database.php (singleton, po profesoru)**

**core/Database.php**

php

CopyEdit

<?php

namespace Core;

use PDO;

use PDOException;

class Database {

private static $instance;

private $connection;

private function \_\_construct() {

try {

$this->connection = new PDO(

"mysql:host=localhost;dbname=blogdb",

"root",

"Lozinka123." // prilagodi po potrebi

);

$this->connection->setAttribute(PDO::ATTR\_DEFAULT\_FETCH\_MODE, PDO::FETCH\_ASSOC);

$this->connection->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

} catch (PDOException $e) {

die("Greška kod povezivanja: " . $e->getMessage());

}

}

public static function getInstance() {

if (!self::$instance) {

self::$instance = new self();

}

return self::$instance;

}

public function getConnection() {

return $this->connection;

}

}

**🔢 6. Model – Article.php**

**app/Models/Article.php**

php

CopyEdit

<?php

namespace App\Models;

use Core\Database;

class Article {

private $db;

public function \_\_construct() {

$this->db = Database::getInstance()->getConnection();

}

public function getAll() {

$stmt = $this->db->prepare("SELECT \* FROM clanci");

$stmt->execute();

return $stmt->fetchAll();

}

public function getById($id) {

$stmt = $this->db->prepare("SELECT \* FROM clanci WHERE id = :id");

$stmt->bindParam(':id', $id);

$stmt->execute();

return $stmt->fetch();

}

public function create($naslov, $tijelo) {

$stmt = $this->db->prepare("INSERT INTO clanci (korisnikId, naslov, tijelo) VALUES (1, :naslov, :tijelo)");

$stmt->bindParam(':naslov', $naslov);

$stmt->bindParam(':tijelo', $tijelo);

$stmt->execute();

}

public function update($id, $naslov, $tijelo) {

$stmt = $this->db->prepare("UPDATE clanci SET naslov = :naslov, tijelo = :tijelo WHERE id = :id");

$stmt->bindParam(':id', $id);

$stmt->bindParam(':naslov', $naslov);

$stmt->bindParam(':tijelo', $tijelo);

$stmt->execute();

}

public function delete($id) {

$stmt = $this->db->prepare("DELETE FROM clanci WHERE id = :id");

$stmt->bindParam(':id', $id);

$stmt->execute();

}

}

**🔢 7. Controller – ArticlesController.php**

**app/Controllers/ArticlesController.php**

php

CopyEdit

<?php

namespace App\Controllers;

use App\Models\Article;

class ArticlesController {

public function index() {

$articles = (new Article())->getAll();

view('articles/index.view.php', [

'articles' => $articles

]);

}

public function create() {

view('articles/create.view.php');

}

public function store() {

$naslov = $\_POST['naslov'];

$tijelo = $\_POST['tijelo'];

(new Article())->create($naslov, $tijelo);

redirect('/articles');

}

public function edit() {

$article = (new Article())->getById($\_GET['id']);

view('articles/edit.view.php', [

'article' => $article

]);

}

public function update() {

$id = $\_POST['id'];

$naslov = $\_POST['naslov'];

$tijelo = $\_POST['tijelo'];

(new Article())->update($id, $naslov, $tijelo);

redirect('/articles');

}

public function destroy() {

$id = $\_POST['id'];

(new Article())->delete($id);

redirect('/articles');

}

}

**🔢 8. Router – moderniziran s autoload podrškom**

**core/Router.php**

php

CopyEdit

<?php

namespace Core;

class Router {

private $routes = [];

public function addRoute($method, $path, $controller, $action) {

$this->routes[] = compact('method', 'path', 'controller', 'action');

}

public function get($path, $controller, $action) {

$this->addRoute('GET', $path, $controller, $action);

}

public function post($path, $controller, $action) {

$this->addRoute('POST', $path, $controller, $action);

}

public function dispatch() {

$currentUri = parse\_url($\_SERVER['REQUEST\_URI'], PHP\_URL\_PATH);

$method = $\_SERVER['REQUEST\_METHOD'];

foreach ($this->routes as $route) {

if ($route['method'] === $method && $route['path'] === $currentUri) {

$controllerClass = 'App\\Controllers\\' . $route['controller'];

$controller = new $controllerClass();

return $controller->{$route['action']}();

}

}

http\_response\_code(404);

echo "404 - Stranica nije pronađena.";

}

}

**🔢 9. Helper funkcije**

**core/functions.php**

php

CopyEdit

<?php

function dd($value) {

echo '<pre>';

var\_dump($value);

echo '</pre>';

die();

}

function view($pathToView, $data = []) {

extract($data);

require \_\_DIR\_\_ . '/../app/Views/' . $pathToView;

}

function redirect($path) {

header('Location: ' . $path);

exit;

}

**🔢 10. index.php (ulazna točka)**

**public/index.php**

php

CopyEdit

<?php

require\_once \_\_DIR\_\_ . '/../vendor/autoload.php';

require\_once \_\_DIR\_\_ . '/../core/functions.php';

use Core\Router;

$router = new Router();

require\_once \_\_DIR\_\_ . '/../routes.php';

$router->dispatch();

**🔢 11. routes.php**

**routes.php**

php

CopyEdit

<?php

$router->get('/articles', 'ArticlesController', 'index');

$router->get('/articles/create', 'ArticlesController', 'create');

$router->post('/articles/store', 'ArticlesController', 'store');

$router->get('/articles/edit', 'ArticlesController', 'edit');

$router->post('/articles/update', 'ArticlesController', 'update');

$router->post('/articles/delete', 'ArticlesController', 'destroy');

**🔢 12. Views (po potrebi)**

* app/Views/articles/index.view.php
* app/Views/articles/create.view.php
* app/Views/articles/edit.view.php

Želiš li da ti odmah napišem HTML za te fajlove?

**🔚 REZIME**

✅ Singleton baza kao kod profesora  
✅ Composer + namespace organizacija  
✅ PSR-4 kompatibilno  
✅ Moderni MVC stil kompatibilan s Laravel logikom  
✅ CRUD funkcionalnost gotova