**1. Postavljanje baze podataka**

Prvo, kreirajte bazu podataka i tablice:

CREATE DATABASE inventory\_management;

USE inventory\_management;

CREATE TABLE users (

id INT AUTO\_INCREMENT PRIMARY KEY,

username VARCHAR(255) NOT NULL UNIQUE,

password VARCHAR(255) NOT NULL,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

CREATE TABLE products (

id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(255) NOT NULL,

quantity INT NOT NULL,

price DECIMAL(10, 2) NOT NULL,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP

);

**2. Modeli**

**models/Product.php**

<?php

class Product {

private $db;

public function \_\_construct($db) {

$this->db = $db;

}

public function getAllProducts() {

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

$stmt->execute();

return $stmt->fetchAll(PDO::FETCH\_ASSOC);

}

public function addProduct($name, $quantity, $price) {

$stmt = $this->db->prepare("INSERT INTO products (name, quantity, price) VALUES (:name, :quantity, :price)");

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

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

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

return $stmt->execute();

}

public function editProduct($id, $name, $quantity, $price) {

$stmt = $this->db->prepare("UPDATE products SET name = :name, quantity = :quantity, price = :price WHERE id = :id");

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

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

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

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

return $stmt->execute();

}

public function deleteProduct($id) {

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

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

return $stmt->execute();

}

}

?>

models/User.php

<?php

class User {

private $db;

public function \_\_construct($db) {

$this->db = $db;

}

public function register($username, $password) {

$hashedPassword = password\_hash($password, PASSWORD\_DEFAULT);

$stmt = $this->db->prepare("INSERT INTO users (username, password) VALUES (:username, :password)");

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

$stmt->bindParam(':password', $hashedPassword);

return $stmt->execute();

}

public function login($username, $password) {

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

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

$stmt->execute();

$user = $stmt->fetch(PDO::FETCH\_ASSOC);

if ($user && password\_verify($password, $user['password'])) {

return $user;

}

return false;

}

}

?>

**3. Kontroleri**

**controllers/ProductController.php**

<?php

class ProductController {

private $productModel;

public function \_\_construct($db) {

$this->productModel = new Product($db);

}

public function listProducts() {

$products = $this->productModel->getAllProducts();

include 'views/product/list.php';

}

public function showAddProductForm() {

include 'views/product/add.php';

}

public function processAddProduct() {

if ($\_SERVER['REQUEST\_METHOD'] == 'POST') {

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

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

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

$this->productModel->addProduct($name, $quantity, $price);

header('Location: index.php?action=listProducts');

}

}

public function showEditProductForm($id) {

$product = $this->productModel->getProductById($id);

include 'views/product/edit.php';

}

public function processEditProduct() {

if ($\_SERVER['REQUEST\_METHOD'] == 'POST') {

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

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

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

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

$this->productModel->editProduct($id, $name, $quantity, $price);

header('Location: index.php?action=listProducts');

}

}

public function deleteProduct($id) {

$this->productModel->deleteProduct($id);

header('Location: index.php?action=listProducts');

}

}

?>

controllers/UserController.php

<?php

class UserController {

private $userModel;

public function \_\_construct($db) {

$this->userModel = new User($db);

}

public function showLoginForm() {

include 'views/user/login.php';

}

public function processLogin() {

if ($\_SERVER['REQUEST\_METHOD'] == 'POST') {

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

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

$user = $this->userModel->login($username, $password);

if ($user) {

$\_SESSION['user\_id'] = $user['id'];

header('Location: index.php');

} else {

$error = "Invalid credentials";

include 'views/user/login.php';

}

}

}

public function showRegisterForm() {

include 'views/user/register.php';

}

public function processRegister() {

if ($\_SERVER['REQUEST\_METHOD'] == 'POST') {

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

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

$this->userModel->register($username, $password);

header('Location: index.php?action=login');

}

}

public function logout() {

session\_destroy();

header('Location: index.php');

}

}

?>

**4. Pogledi (Views)**

**views/product/list.php**

<!DOCTYPE html>

<html>

<head>

<title>Products List</title>

</head>

<body>

<h1>Products</h1>

<a href="index.php?action=showAddProductForm">Add Product</a>

<table>

<tr>

<th>ID</th>

<th>Name</th>

<th>Quantity</th>

<th>Price</th>

<th>Actions</th>

</tr>

<?php foreach ($products as $product): ?>

<tr>

<td><?= $product['id'] ?></td>

<td><?= $product['name'] ?></td>

<td><?= $product['quantity'] ?></td>

<td><?= $product['price'] ?></td>

<td>

<a href="index.php?action=showEditProductForm&id=<?= $product['id'] ?>">Edit</a>

<a href="index.php?action=deleteProduct&id=<?= $product['id'] ?>">Delete</a>

</td>

</tr>

<?php endforeach; ?>

</table>

</body>

</html>

views/product/add.php

<!DOCTYPE html>

<html>

<head>

<title>Add Product</title>

</head>

<body>

<h1>Add Product</h1>

<form action="index.php?action=processAddProduct" method="post">

<label for="name">Name:</label>

<input type="text" id="name" name="name" required>

<label for="quantity">Quantity:</label>

<input type="number" id="quantity" name="quantity" required>

<label for="price">Price:</label>

<input type="number" step="0.01" id="price" name="price" required>

<input type="submit" value="Add Product">

</form>

</body>

</html>

views/product/edit.php

<!DOCTYPE html>

<html>

<head>

<title>Edit Product</title>

</head>

<body>

<h1>Edit Product</h1>

<form action="index.php?action=processEditProduct" method="post">

<input type="hidden" name="id" value="<?= $product['id'] ?>">

<label for="name">Name:</label>

<input type="text" id="name" name="name" value="<?= $product['name'] ?>" required>

<label for="quantity">Quantity:</label>

<input type="number" id="quantity" name="quantity" value="<?= $product['quantity'] ?>" required>

<label for="price">Price:</label>

<input type="number" step="0.01" id="price" name="price" value="<?= $product['price'] ?>" required>

<input type="submit" value="Edit Product">

</form>

</body>

</html>

views/user/login.php

<!DOCTYPE html>

<html>

<head>

<title>Login</title>

</head>

<body>

<h1>Login</h1>

<form action="index.php?action=processLogin" method="post">

<label for="username">Username:</label>

<input type="text" id="username" name="username" required>

<label for="password">Password:</label>

<input type="password" id="password" name="password" required>

<input type="submit" value="Login">

</form>

<?php if (isset($error)): ?>

<p style="color: red;"><?= $error ?></p>

<?php endif; ?>

</body>

</html>

views/user/register.php

<!DOCTYPE html>

<html>

<head>

<title>Register</title>

</head>

<body>

<h1>Register</h1>

<form action="index.php?action=processRegister" method="post">

<label for="username">Username:</label>

<input type="text" id="username" name="username" required>

<label for="password">Password:</label>

<input type="password" id="password" name="password" required>

<input type="submit" value="Register">

</form>

</body>

</html>

**5. Glavna aplikacija**

**index.php**

<?php

session\_start();

require 'config/database.php';

require 'models/Product.php';

require 'models/User.php';

require 'controllers/ProductController.php';

require 'controllers/UserController.php';

// Povezivanje s bazom podataka

try {

$db = new PDO('mysql:host=localhost;dbname=inventory\_management;charset=utf8', 'root', '');

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

} catch (PDOException $e) {

die("Connection failed: " . $e->getMessage());

}

$productController = new ProductController($db);

$userController = new UserController($db);

$action = isset($\_GET['action']) ? $\_GET['action'] : '';

switch ($action) {

case 'listProducts':

$productController->listProducts();

break;

case 'showAddProductForm':

$productController->showAddProductForm();

break;

case 'processAddProduct':

$productController->processAddProduct();

break;

case 'showEditProductForm':

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

$productController->showEditProductForm($id);

break;

case 'processEditProduct':

$productController->processEditProduct();

break;

case 'deleteProduct':

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

$productController->deleteProduct($id);

break;

case 'showLoginForm':

$userController->showLoginForm();

break;

case 'processLogin':

$userController->processLogin();

break;

case 'showRegisterForm':

$userController->showRegisterForm();

break;

case 'processRegister':

$userController->processRegister();

break;

case 'logout':

$userController->logout();

break;

default:

if (isset($\_SESSION['user\_id'])) {

$productController->listProducts();

} else {

$userController->showLoginForm();

}

break;

}

?>

config/database.php

<?php

define('DB\_HOST', 'localhost');

define('DB\_NAME', 'inventory\_management');

define('DB\_USER', 'root');

define('DB\_PASS', '');

?>