



VILNIUSCODINGSCHOOL

Planas



- **Data layer klasių kūrimas**
- **Model layer klasių kūrimas**
- **Service layer klasių kūrimas**
- **Duomenų bazės**
- **PHP ir duomenų bazės**
- **Gautų duomenų grąžinimas JSON formatu**
- **Ajax**

Data layer klasių kūrimas

- User duomenų klasės kūrimas
- Expense duomenų klasės kūrimas

Model layer klasių kūrimas

- User modelio klasės kūrimas
- Expense modelio klasės kūrimas

Service/controller layer klasių kūrimas

- Interface kūrimas
- Service klasės kūrimas pagal interface
 - UserService
 - ExpensesService

Duomenų bazės

- PHPMYAdmin
- Finances duomenų bazės kūrimas
- User lentelės kūrimas
- Expense lentelės kūrimas
-
-

Duomenų bazės

- Prisijungimas prie duomenų bazės naudojant PHP skriptą
-

```
class ConnectionFactory{
    public static function GetConnection(){

        $servername = "localhost";
        $username = "root";
        $password = "";
        $dbname = "phpjs3";

        $conn = new mysqli($servername, $username, $password, $dbname);
        if($conn->connect_error){
            return null;
        }
        return $conn;
    }

    function ValidateConnection($conn){

    }
}
```

PHP ir duomenų bazės

- Duomenų įterpimas į duomenų bazę naudojant PHP skriptą

```
public function create($user){  
  
    require_once("../conn.php");  
    $conn = GetConnection();  
    ValidateConnection($conn);  
  
    if($stmt = $conn->prepare("INSERT INTO User (name, surname, created) VALUES(?, ?, NOW())")){  
  
        $stmt->bind_param("ss", $name, $surname);  
  
        $name = $_POST["name"];  
        $eventdate = $_POST["surname"];  
  
        if($stmt->execute()){  
            return json_encode(new Response("OK", null, $user));  
        }  
        else{  
            return json_encode(new Response("ERROR", "Žiūrėkite logus", null));  
        }  
    }  
    else{  
        return json_encode(new Response("ERROR", "Žiūrėkite logus", null));  
    }  
}
```


PHP ir duomenų bazės

- Duomenų nuskaitymas naudojant php scriptą

```
if($stmt = $conn->prepare("SELECT username, password FROM users WHERE username=?")){

    $stmt->bind_param("s", $username);

    $username = $_POST["username"];

    if($stmt->execute()){
        $stmt->bind_result($username, $password);

        $stmt->fetch();

        if(password_verify($_POST["password"], $password)){
            $_SESSION["username"] = $username;
            header('Location: ../../index.php');
        }
        else{
            $_SESSION["error"] = "Netinkamas vartotojo vardas ir/arba slaptažodis!";
            header('Location: ../../error.php');
        }
    }
    else{
        $_SESSION["error"] = "Problemos su kodu. Ieškokit kažkur klaidos!";
        header('Location: ../../error.php');
    }
}
else{
    echo $conn->error;
}
```

Gautų duomenų grąžinimas JSON formatu

- Duomenų talpinimas į objektų masyvą
- Json masyvo grąžinimas

```
if($stmt = $conn->prepare("SELECT name, surname FROM events")){

    if($stmt->execute()){
        $stmt->bind_result($event, $eventdate);

        // paimame rezultatus visus
        $result = $stmt->get_result();

        $resultToReturn = array();

        while ($row = $result->fetch_assoc()) {
            $event = new stdClass();
            $event->event = $row["name"];
            $event->eventdate = $row["eventdate"];

            array_push($resultToReturn, $event);
        }

        $eventsJSON = json_encode($resultToReturn);

        echo $eventsJSON;
    }
}
else{
    $event = new stdClass();
    $event->error = $conn->error;
    echo json_encode($event);
}
```

Ajax

- Front end funkcijų su ajax rašymas
- Duomenų atvaizdavimas

Klausimai

