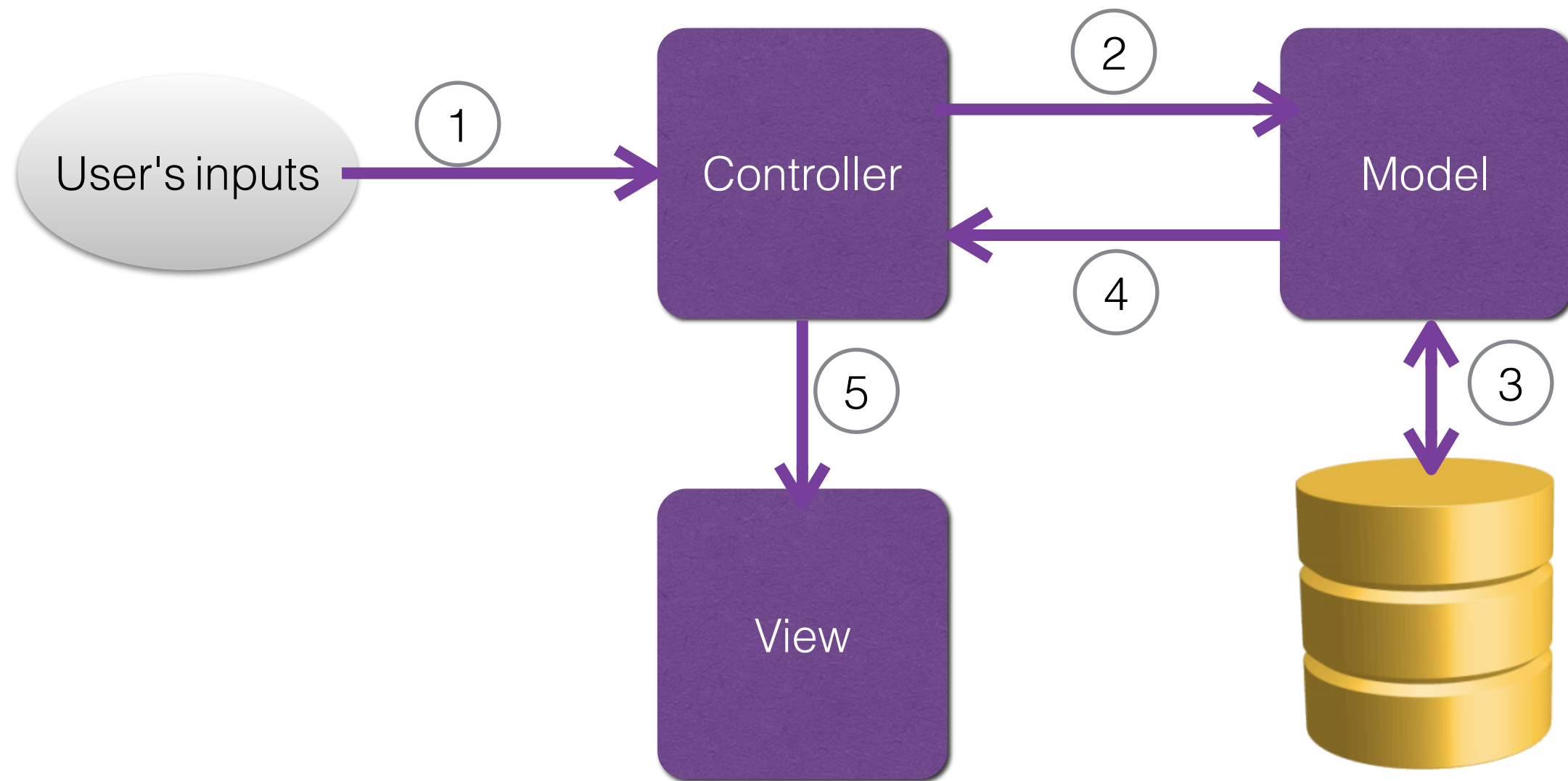


The Model-View- Controller (MVC)

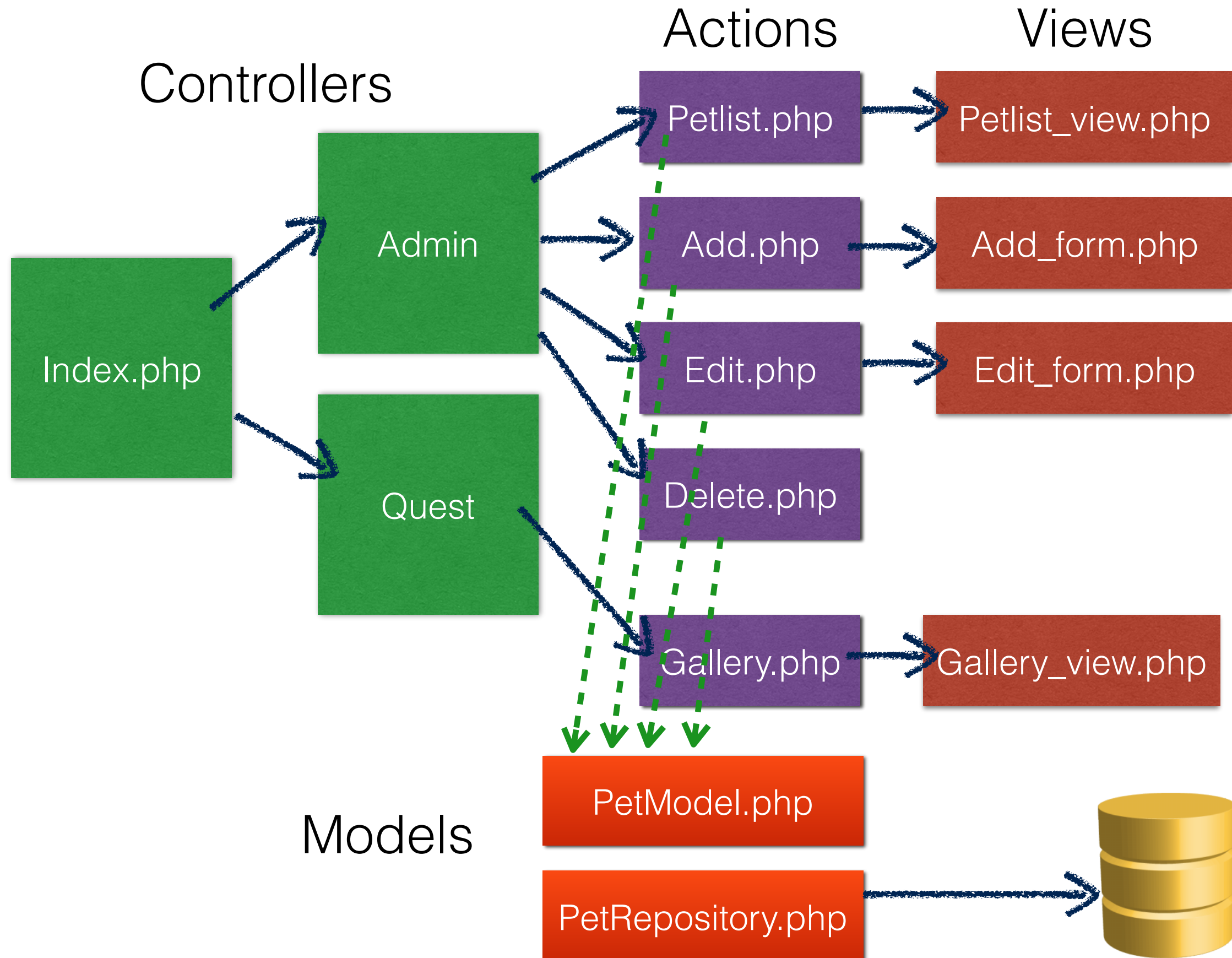
The model-view-controller (MVC)

- The model-view-controller (MVC) design pattern is a common approach to organizing scripts consistently
- MVC separates coding concerns into three categories: models, views and controllers.
 - A **model** that represents data. Your models should also hold most logic involved in the system you're building.
 - A **view** that shows information visually. The information to be displayed by the view is received from a model.
 - A **controller** that retrieves input from users and sends commands to relevant model(s).
- The current project structure:
 - View - The /templates/page.php
 - Model - The /classes/PageData.php
 - Controller - The index.php

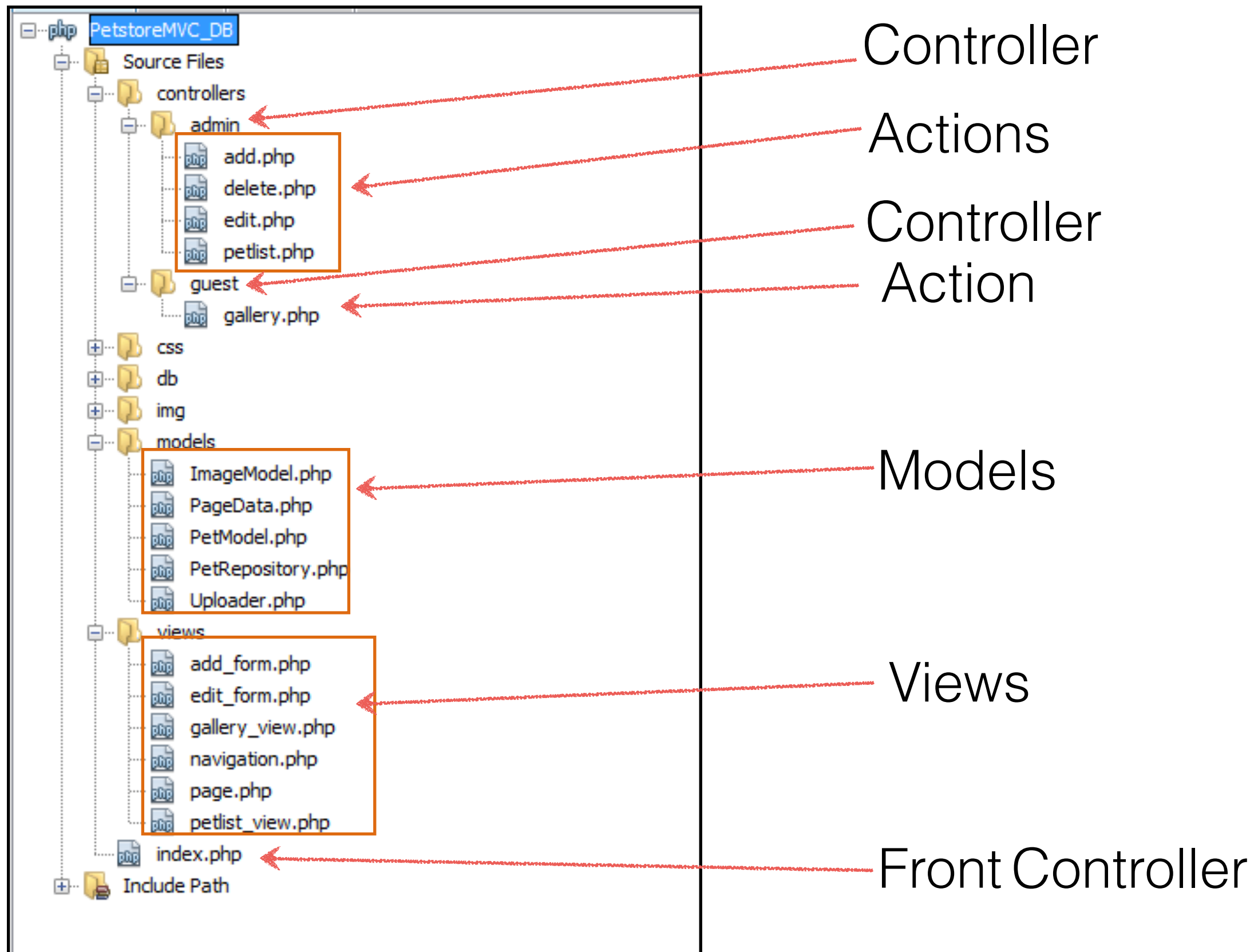
The Interaction in MVC



Creating the Petstore MVC Project



The Pet Store Project Structure



General Steps of Creating a MVC project

- Create a new PHP Project
- Create folder structures as shown in the previous slide.
 - Add basic files
 - PageData.php in models folder
 - navigation.php in views folder
 - page.php in views folder
 - layout.css in css folder
 - navigation.css in css folder
- Create a database script for generating the database, tables, and data. And write PHP scripts for connect to the database.
 - petdb.sql in db folder
 - db_connect.php in db folder
 - db_error.php in db folder
- Create model classes
 - PetMode.php in models folder
 - PetRepository.php
- Create controllers, example
 - petlist.php in controllers/admin folder
- Create views, example
 - petlist_view.php in views folder

petdb.sql

```
DROP DATABASE IF EXISTS petdb;
CREATE DATABASE petdb;USE petdb;

CREATE TABLE pet (
pet_id          INT(11)          NOT NULL      AUTO_INCREMENT,
petName         VARCHAR(255)     NOT NULL,
price           DECIMAL(10,2)    NOT NULL,
imagePath       VARCHAR(255)     NOT NULL,
PRIMARY KEY (pet_id));

INSERT INTO pet VALUES
(1, 'coco', 1000, 'img/coco.jpg'),
(2, 'bobo', 800, 'img/bobo.jpg'),
(3, 'dodo', 700, 'img/dodo.jpg');

GRANT SELECT, INSERT, DELETE, UPDATE
ON petdb.*TO admin@localhost
IDENTIFIED BY 'pass@word';

GRANT SELECTON pet
TO peter@localhost
IDENTIFIED BY 'pass@word';
```

db_connect.php

```
<?php
$dsn = 'mysql:host=localhost;dbname=petdb';
$username = 'admin';
$password = 'pass@word';

try {
    $db = new PDO($dsn, $username,
        $password);    echo 'Connected.';
} catch (PDOException $ex) {
    $error_msg = $ex->getMessage();
    include('db_error.php');    exit();
}
?>
```


PetModel.php

```
<?php
class Pet {
    public $petID;
    public $petName;
    public $price;
    public $imagePath;

    public function __construct($petID, $petName, $price,
                                $imagePath) {
        $this->petID = $petID;
        $this->petName = $petName;
        $this->price = $price;
        $this->imagePath = $imagePath;
    }
}
```

PetRepository.php

```
<?php
include_once 'models/PetModel.php';
class PetRepository {
    //declare a new property
    private $db;

    //declare a constructor method requires a database connection as argument
    public function __construct($dbConnection) {
        //store the received connection in the $this->db property
        $this->db = $dbConnection;
    }

    public function getPets() {
        //the actual SQL statement
        $petList = array();
        $i = 0;
        $sql = "SELECT pet_id, petName, price, imagePath FROM pet";
        $pets = $this->db->query($sql);
        $pet = $pets->fetch();
        while ($pet) {
            $petList[$i++] = new Pet($pet['pet_id'],
                                    $pet['petName'],
                                    $pet['price'],
                                    $pet['imagePath']);
            $pet = $pets->fetch();
        }
        return $petList;
    }
}
```

PetRepository.php

```
public function savePet($pet) {
    $sql = "UPDATE pet SET "
        . "petName = '$pet->petName'"
        . ", price = $pet->price"
        . ", imagePath = '$pet->imagePath' "
        . "WHERE pet_id = $pet->petID";
    $updateCount = $this->db->exec($sql);
    if ($updateCount > 0)
        return "$pet->petName was saved.";
    else
        return "$pet->petName wasn't saved.";
}

public function deletePet($petID) {
    $sql = "DELETE FROM pet "
        . "WHERE pet_id = $petID";
    $deleteCount = $this->db->exec($sql);
    if ($deleteCount > 0)
        return "PetID: $petID was deleted.";
    else
        return "PetID: $petID wasn't deleted.";
}

public function addPet($pet) {
    $sql = "INSERT INTO pet (petName, price, imagePath) VALUES ("
        . "'$pet->petName', "
        . "$pet->price, "
        . "'$pet->imagePath')";
    $insertCount = $this->db->exec($sql);
    if ($insertCount > 0)
        return "$pet->petName was added.";
    else
        return "$pet->petName wasn't added.";
}
```

index.php (The Front Controller)

```
<?php
include_once "models/PageData.php";
$pageData = new PageData();
$pageData->title = "Sunny Pet Store";
$pageData->addCSS('css/layout.css');
$pageData->addCSS('css/navigation.css');

//connect to database
include_once "db/db_connect.php";

$pageData->content .= include_once "views/navigation.php";
$navigationIsClicked = isset($_GET["action"]);
if ($navigationIsClicked) {
    $controller = $_GET["action"];
} else {
    $controller = "guest/gallery";
}
$pageData->content .= include_once "controllers/
$controller.php";
$page = include_once "views/page.php";
echo $page;
```

The Petlist Controller's action (petlist.php)

```
<?php
//complete code listing for controllers/admin/
petlist.php

// Load required model classes
include_once "models/PetModel.php";
include_once "models/PetRepository.php";

// Retrieve the data from the model
$petDB = new PetRepository($db);
$petList = $petDB->getPets();

// Specify which view to use
$petListView = include_once "views/
petlist_view.php";

return $petListView;
```

The Petlist view (petlist_view.php)

```
<?php
$hasPets = isset($petList);
if ($hasPets === false) {
    return 'views/petlist_view.php needs $petList';
}
$output = "
    <table>
        <tr>
            <th>Petname</th>
            <th>Price</th>
            <th>Picture</th>
            <th>Edit</th>
            <th>Delete</th>
        </tr>
";

foreach($petList as $pet) {
    $output .= "<tr>";
    $src = $pet->imagePath;
    $output .= "<td>$pet->petName</td>"
        . "<td>$pet->price</td>"
        . "<td><img src='$src' height=100 width=100 /></td>"
        . "<td><a href='index.php?action=admin/edit&petID=$pet->petID&petname=$pet->petName&price=$pet->price'>Edit</a></td>"
        . "<td><a href='index.php?action=admin/delete&petID=$pet->petID'>Delete</a></td>";
    $output .= "</tr>";
}

$output .= '</table>';
return $output;
```

Connected.
[Gallery](#) [Pet List](#) [Add New Pet](#)

Petname	Price	Picture	Edit	Delete
coco	1000.00		Edit	Delete
bobo	800.00		Edit	Delete
dodo	700.00		Edit	Delete

The Edit Controller's action (edit.php)


```
<?php
// Load required model classes
include_once "models/PetModel.php";
include_once "models/PetRepository.php";

// Check if the form is postback
$editSubmitted = isset($_POST['edit_submitted']);
if ($editSubmitted) {
    $operation = $_POST['edit_submitted'];
    if ($operation == 'Cancel') {
        $output = include_once "controllers/petlist.php";
    } else if ($operation == 'Save') {
        $petID = $_POST['petID'];
        $petname = $_POST['petname'];
        $price = $_POST['price'];
        $pet = new Pet($petID, $petname, $price, "img/$petname.jpg");
        $petDB = new PetRepository($db);
        $output = $petDB->savePet($pet);
    }
} else {
    //this runs if form was NOT submitted
    $petid = $_GET['petID'];
    $petname = $_GET['petname'];
    $price = $_GET['price'];
    // Specify the view
    $output = include_once "views/edit_form.php";
}
return $output;
```

The Edit View (edit_form.php)

```
<?php
return "
<form method='post' action='index.php?action=admin/edit' >

    <h4>Edit a pet information</h4>
    <input type='text' name='petID' value='$petid' hidden />
    Petname: <input type='text' name='petname' readonly
    value='$petname' /><br>
    Price: <input type='text' name='price' value='$price' /
    ><br>
    <input type='submit' value='Save' name='edit_submitted' /
    >
    <input type='submit' value='Cancel' name='edit_submitted' />
</form>
";
```



Connected.

[Gallery](#) [Pet List](#) [Add New Pet](#)

Edit a pet information

Petname:

Price:

Save

Cancel

As we don't want the user view or modify the petID. However the petID is required when updating the record in the database. So, we make it as a hidden field.