
SERVER-SIDE WEB DEVELOPMENT

Lecture 9

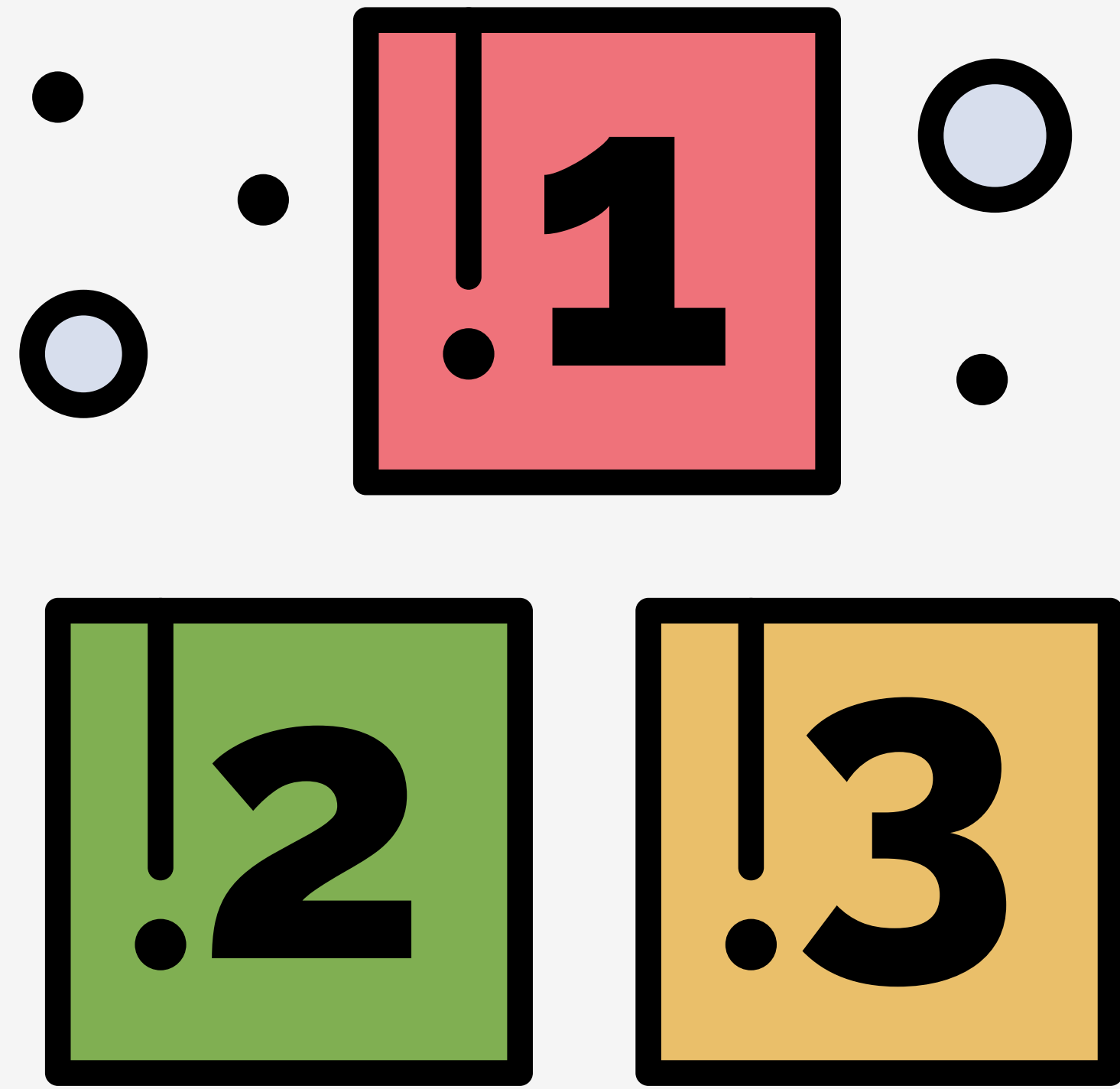
TODAY'S TOPICS



- Restful Controllers
- Restful Routes
- Forms

RESTFUL CONTROLLERS

RESTFUL CONTROLLERS



- **index** - displays all items
- **show** - display one item
- **create** - displays form to create
- **store** - creates a new item
- **edit** - displays form to edit an item
- **update** - updates an item
- **destroy** - deletes an item

INDEX

```
public function index () {  
    // retrieve all movies  
    $movies = Movie::all();  
  
    return view('movie', ['movies' => $movies]);  
}
```

SHOW

```
public function show ($id) {  
    // retrieve a movie  
    $movie = Movie::find($id);  
  
    return view('details', ['movie' => $movie]);  
}
```

CREATE

```
public function create () {  
    // display new movie form  
    return view('create');  
}
```

STORE

```
public function store () {  
    // add a new movie  
    $movie = new Movie;  
    $movie->movie_title = request('movie_title');  
    $movie->save();  
  
    // returns to movies page  
    return redirect('/');  
}
```


EDIT

```
public function edit ($id) {  
    // display edit movie form  
    $movie = Movie::find($id);  
    return view('edit', ['movie' => $movie]);  
}
```

EDIT

```
public function edit ($id) {  
    // display edit movie form  
    $movie = Movie::find($id);  
    return view('edit', ['movie' => $movie]);  
}
```

UPDATE

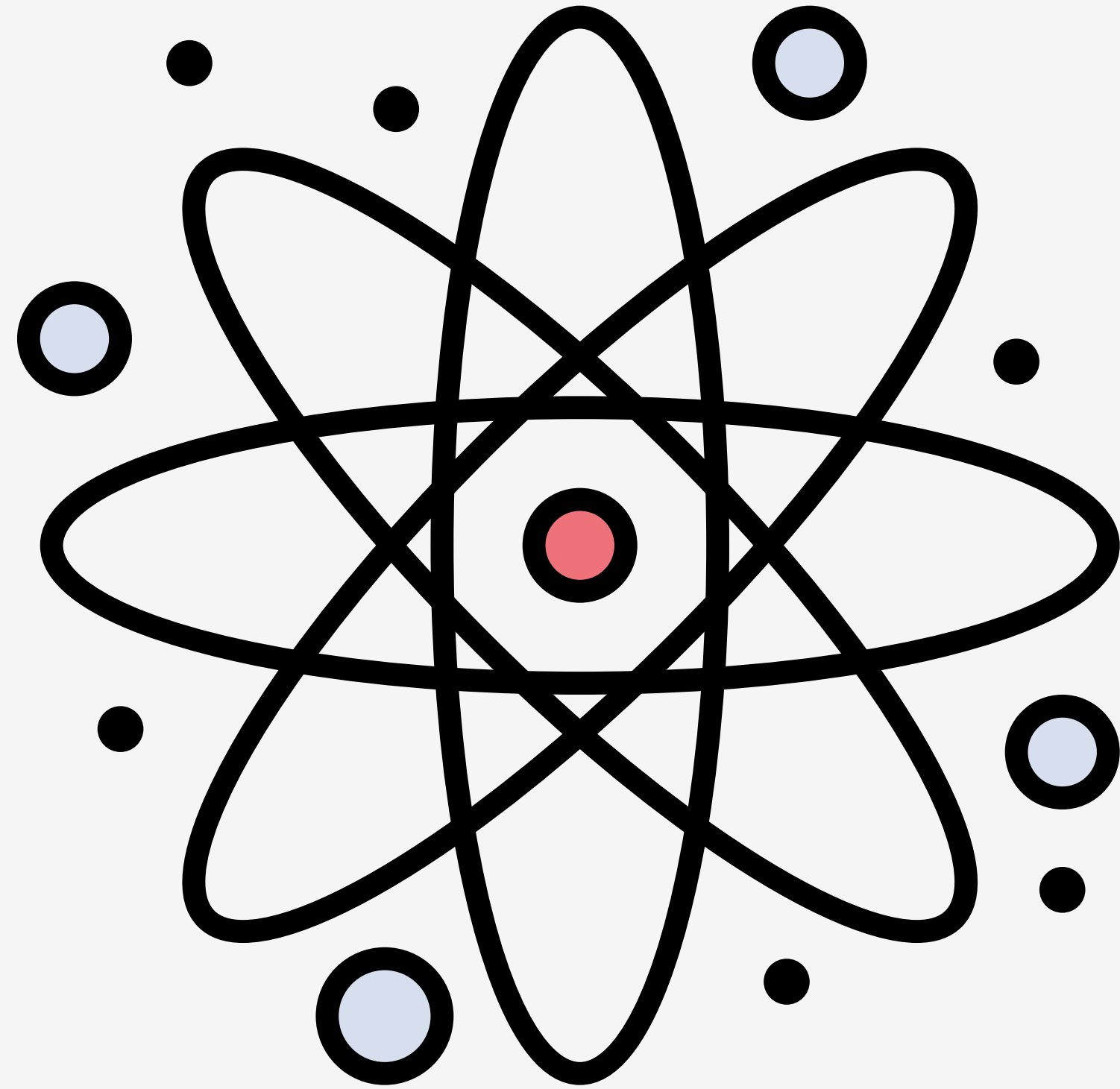
```
public function destroy ($id) {  
    // retrieves movie and deletes it  
    $movie = Movie::find($id);  
    $movie->delete();  
  
    // redirects to movies page  
    return redirect('/');  
}
```

DESTROY

```
public function destroy ($id) {  
    // retrieves movie and deletes it  
    $movie = Movie::find($id);  
    $movie->delete();  
  
    // redirects to movies page  
    return redirect('/');  
}
```

RESTFUL ROUTES

RESTFUL ROUTES



- **get** - displaying data
- **post** - creating data
- **put** - updating data
- **delete** - deleting data

ROUTES

```
Route::get('/', 'MoviesController@index');
```

```
Route::post('/', 'MoviesController@store');
```

```
Route::get('/create', 'MoviesController@create');
```

```
Route::get('/{movie}/edit', 'MoviesController@edit');
```

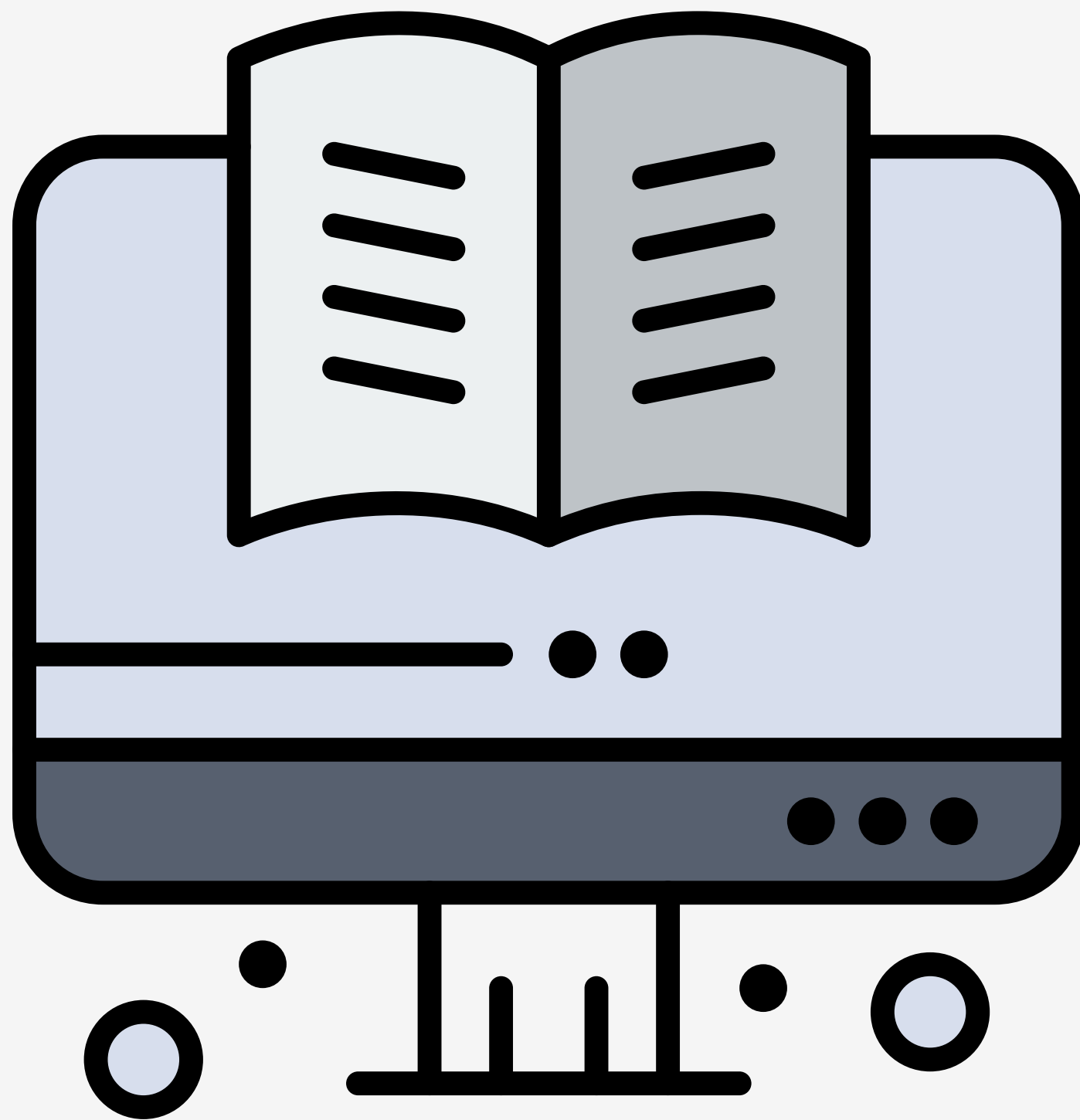
```
Route::get('/{movie}', 'MoviesController@show');
```

```
Route::put('/{movie}', 'MoviesController@update');
```

```
Route::delete('/{movie}',  
    'MoviesController@destroy');
```

FORMS IN LARAVEL

FORMS IN LARAVEL



- Laravel has incorporated many features when working with forms
- Cross-Site Request Forgery
- Method Spoofing
- Form Validation

CSRF

```
<form method="POST" action="/movies">  
  @csrf  
  <input type="text" name="movie_title">  
</form>
```

METHOD SPOOFING

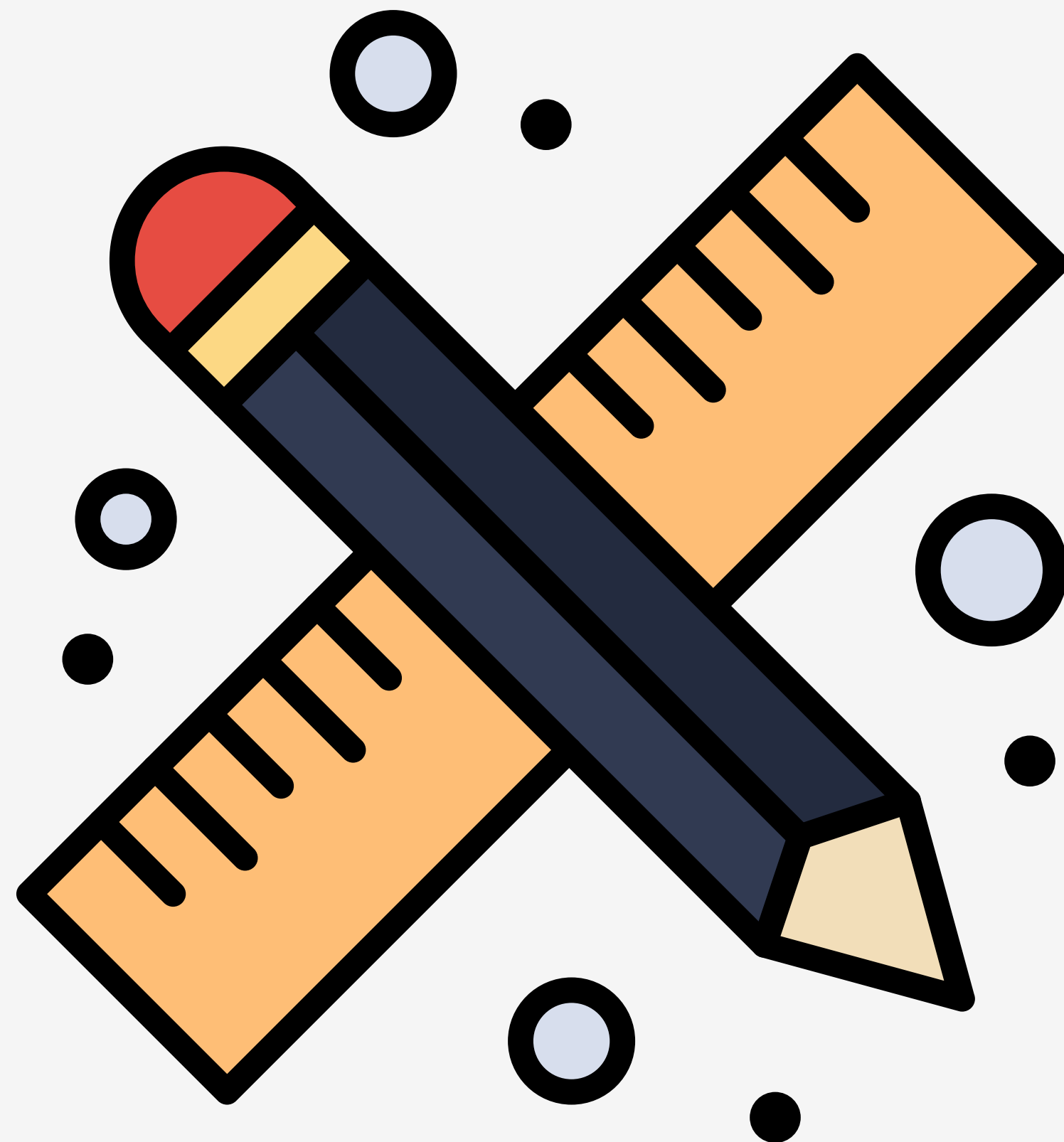
```
<form method="POST"
      action="/movies/{{ $movie->movie_id }}">
  @csrf
  @method( 'PUT' )
  <input type="text" name="movie_title">
</form>
```

STORE

```
public function store () {  
    request()->validate( [  
        'movie_title' => 'required',  
        'director' => 'required',  
        'year' => 'required'  
    ] )  
    // add a new movie  
    $movie = new Movie;  
    $movie->movie_title = request('movie_title');  
    $movie->director = request('director');  
    $movie->year = request('year');  
    $movie->save();  
  
    // returns to movies page  
    return redirect(' / ');  
}
```

HANDS-ON

TASK MANAGER - FINAL



- *GITHUB CLASSROOM ASSIGNMENT*
- Create a Task Manager web application
- Users will need to register and sign in to use application
- Users will be able to create, view, edit, and delete groups (list) and items
- Migration files have been provided
- *DUE:* Wed. Aug. 12 @ 11:59 PM

NEXT TIME...

- Authentication

