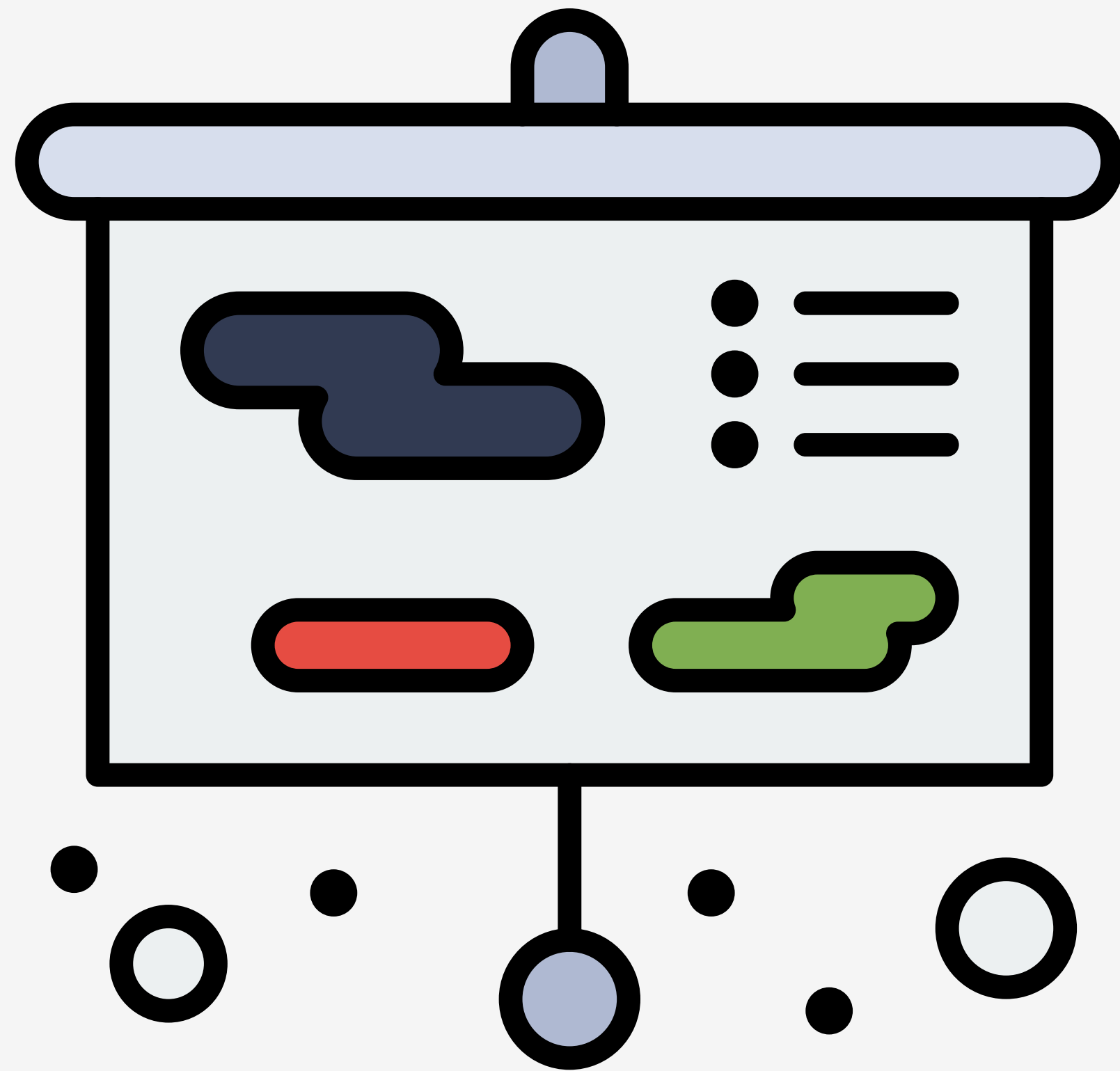

SERVER-SIDE WEB DEVELOPMENT

Lecture 10

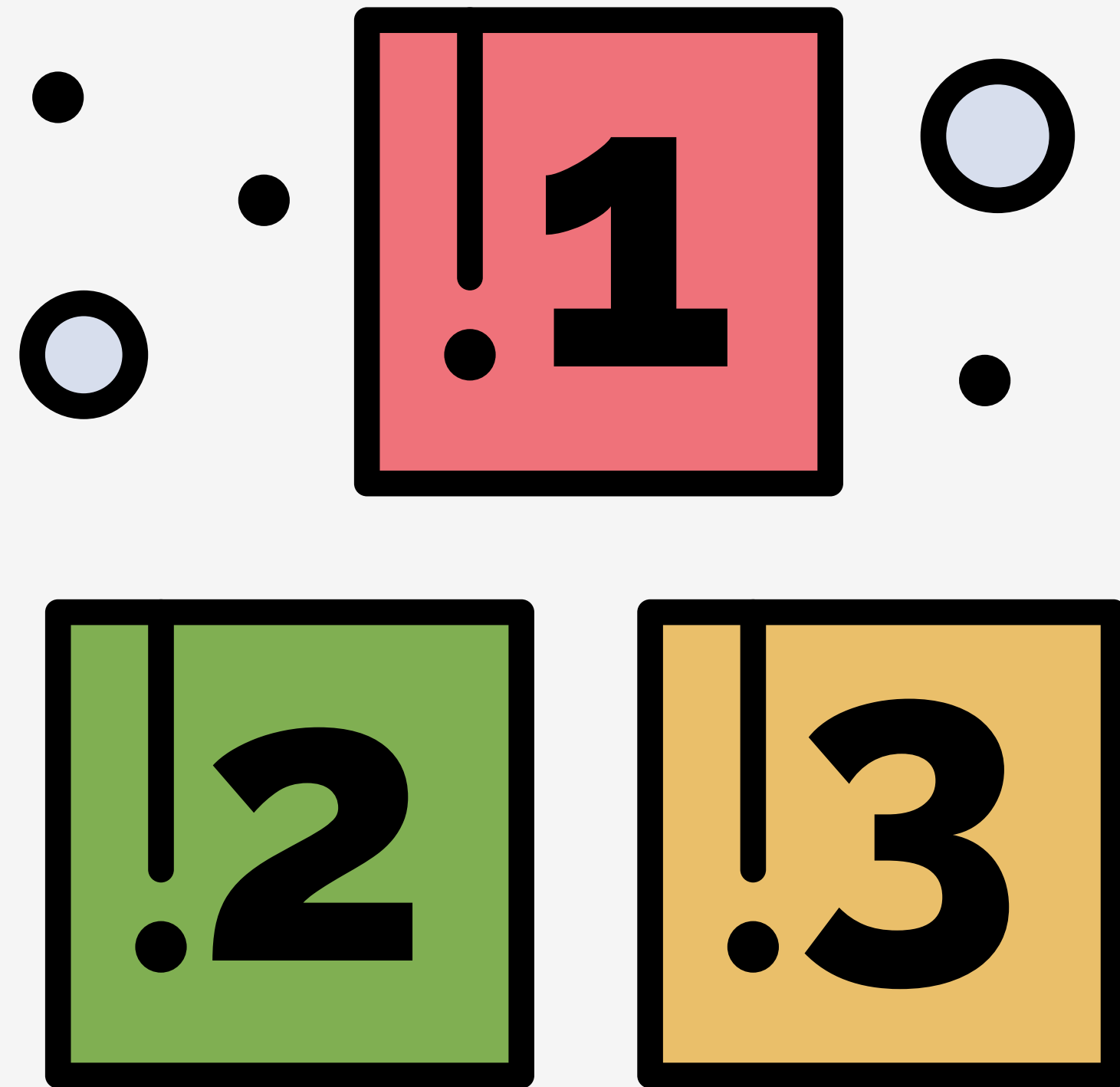
TODAY'S TOPICS

- Authentication



AUTHENTICATION

AUTHENTICATION



- Authentication is the process of verifying a user
- Laravel make it very easy to implement authentication using the **artisan** command
- Laravel authentication does require a **users** table
- Laravel UI provides a quick way creating an Authentication scaffold

TERMINAL

Install Laravel UI

```
composer require laravel/ui --dev
```

Install authentication

```
php artisan ui bootstrap --auth
```

Install dependencies

```
npm install
```

Create a users table

```
php artisan migrate
```



Documentation

Laravel has wonderful, thorough documentation covering every aspect of the framework. Whether you are new to the framework or have previous experience with Laravel, we recommend reading all of the documentation from beginning to end.



Laracasts

Laracasts offers thousands of video tutorials on Laravel, PHP, and JavaScript development. Check them out, see for yourself, and massively level up your development skills in the process.



Laravel News

Laravel News is a community driven portal and newsletter aggregating all of the latest and most important news in the Laravel ecosystem, including new package releases and tutorials.



Vibrant Ecosystem

Laravel's robust library of first-party tools and libraries, such as [Forge](#), [Vapor](#), [Nova](#), and [Envoyer](#) help you take your projects to the next level. Pair them with powerful open source libraries like [Cashier](#), [Dusk](#), [Echo](#), [Horizon](#), [Sanctum](#), [Telescope](#), and more.

Login

E-Mail Address

...

Password

...

☐ Remember Me

Login

[Forgot Your Password?](#)

Register

Name

E-Mail Address

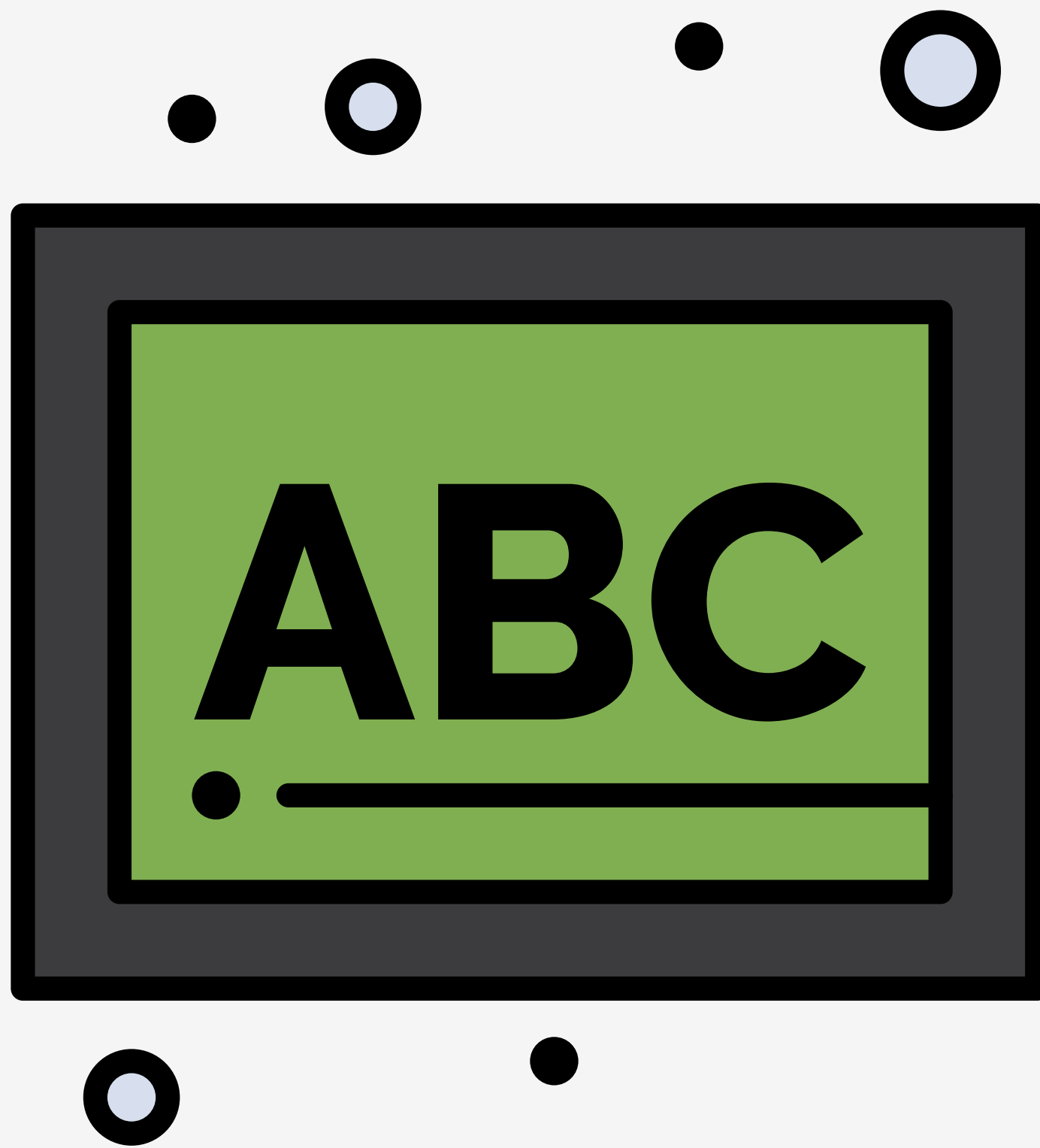
Password

Confirm Password

Register

WORKING WITH USERS

WORKING WITH USERS



- Laravel makes it easy to get information about the currently authenticated user
- The **Auth** class is used to retrieve data about a user
- It is also possible to restrict pages to authenticated users through the use of **middleware**

USERS

```
// BooksController.php
public function index () {
    $books = Book::where('user_id', Auth::id());
    return view(books, ['books', $books])
}
```

```
<!-- books.blade.php -->
<h1>{{ Auth::user()->name }}'s Books</h1>
<ul>
    @foreach ($books as $book)
        <li>{{ $book->title }}</li>
    @endforeach
</ul>
```

RESTRICT ACCESS

```
// web.php
```

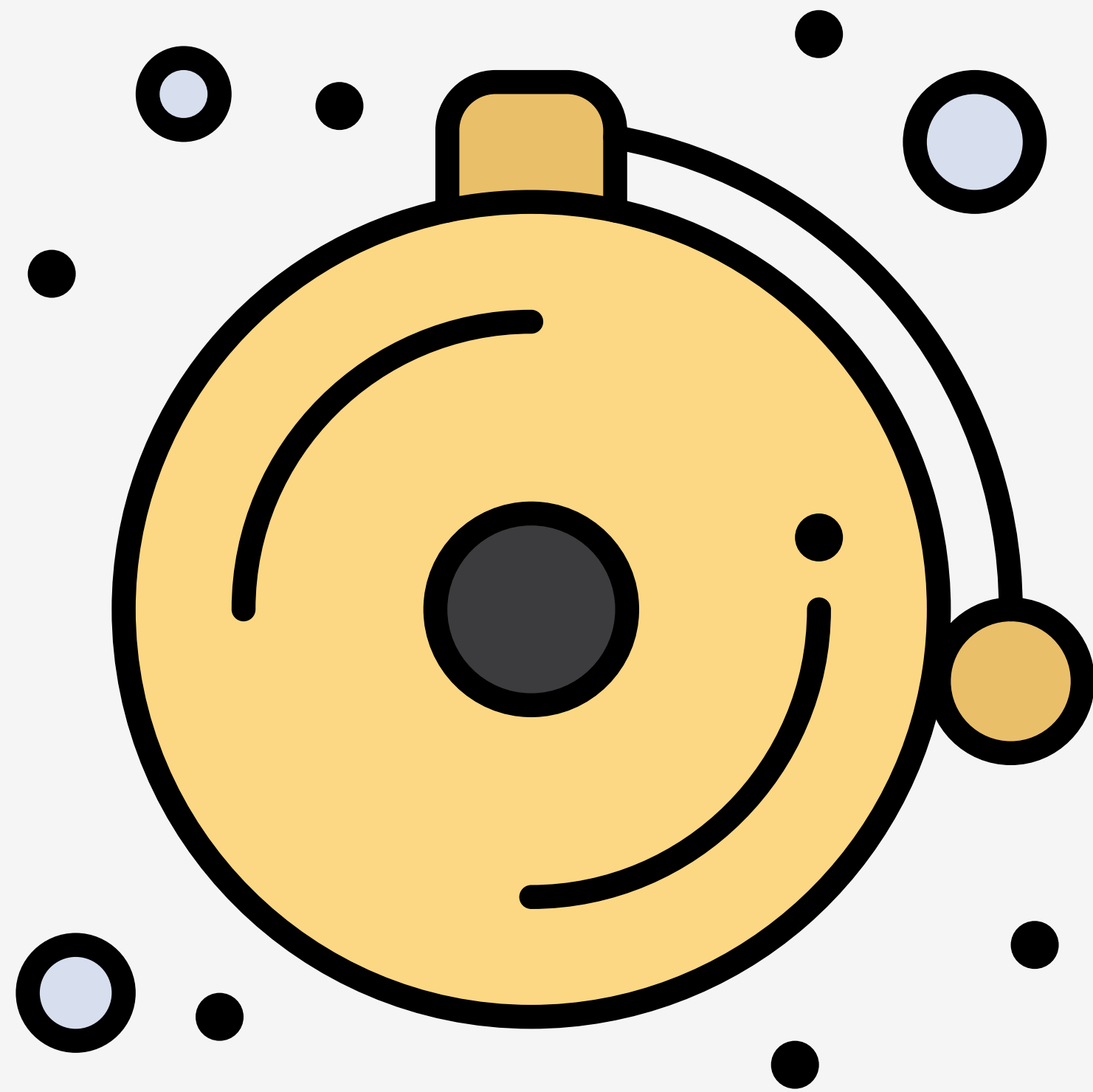
```
Route::get( '/books', 'BooksController@index' )  
    ->middleware( 'auth' )
```

```
// BooksController.php
```

```
public function __construct() {  
    $this->middleware( 'auth' );  
}
```

DEMO

NEXT TIME...



- Review / Work Period
- Due: Grocery List - Dec. 3
- Due: Hybrid #4 - Dec. 10
- Due: Movie Mayhem - Dec. 17
- Due: Task Manager - Dec. 17