




# ΔΙΑΔΙΚΤΥΑΚΗ ΕΦΑΡΜΟΓΗ ΔΙΑΧΕΙΡΙΣΗΣ ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΑΘΗΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ  
ΤΟΥ  
ΤΣΙΑΤΜΑ ΙΩΑΝΝΗ



Επιβλέπων καθηγητής  
Χατζηγεωργίου Αλέξανδρος

# Λειτουργίες εφαρμογής


- Υποστηρίζει τρεις ρόλους χρηστών: Admin, Teacher και Student.
- εγγραφή / σύνδεση χρήστη
- δημιουργία και διαχείριση προφίλ χρήστη
- δημιουργία ηλεκτρονικών μαθημάτων
- οργάνωση του ψηφιακού εκπαιδευτικού υλικού σε ενότητες
- ανάρτηση και πρόσβαση στο ψηφιακό εκπαιδευτικό υλικό
- επικοινωνία καθηγητών και μαθητών μέσω μηνυμάτων και αρχείων
- εγγραφή / απεγγραφή μαθητή σε / από μάθημα
- παροχή στατιστικών, εγχειρίδια χρήσης

# Ηλεκτρονική Μάθηση – Θεωρητικό υπόβαθρο

- Τι είναι η Ηλεκτρονική Μάθηση
- Κατηγορίες Ηλεκτρονικής Μάθησης
  - Σύγχρονη
  - Ασύγχρονη
- Πλεονεκτήματα – Μειονεκτήματα Ηλεκτρονικής Μάθησης
- Συστήματα Διαχείρισης Εκπαίδευσης (LMS)
- Πλατφόρμες LMS - Παραδείγματα
  - Moodle
  - Η-τάξη (E-Class)

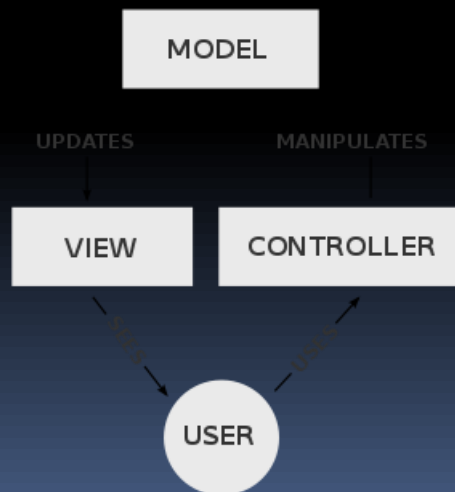


# Γιατί να χρησιμοποιήσουμε ένα PHP Framework

- Εξοικονομεί χρόνο ανάπτυξης
  - Παρέχει καλές πρακτικές κωδικοποίησης
  - Παρέχει μεγαλύτερη ασφάλεια
- 

# Το μοντέλο MVC

Το MVC (Model-Views-Controller) είναι ένα μοτίβο σχεδίασης στην ανάπτυξη λογισμικού που χωρίζει τη λογική του προγράμματος σε 3 διασυνδεδεμένα στοιχεία. τα Models, τα Views και τους Controllers.




# Περιγραφή και βασικά χαρακτηριστικά του Laravel


Δωρεάν, ανοικτού κώδικα PHP πλαίσιο (framework) που δημιουργήθηκε με κύριο στόχο την ανάπτυξη διαδικτυακών εφαρμογών που χρησιμοποιούν το μοντέλο MVC.

## Τεχνικά Χαρακτηριστικά:

Έτος έναρξης	2011
Τρέχουσα έκδοση	8
Είναι γραμμένο σε	PHP
Δημιουργός	Taylor Otwell
Άδεια	MIT License
Απαιτήσεις διακομιστή	PHP >= 7.2.5 (ή χρήση του Laravel Homestead)
Βάσεις Δεδομένων που υποστηρίζει	MySQL, PostgreSQL, SQLite, SQL Server



# Περιγραφή και βασικά χαρακτηριστικά του Laravel

- Blade
  - Παρέχει μια σειρά από χαρακτηριστικά και μεθόδους ασφαλείας
  - Migration
  - Eloquent ORM
  - Artisan CLI
  - Μεγάλη κοινότητα
- 

# Laravel vs Symfony

	LARAVEL	SYMFONY
<b>MODULARITY AND SCALING</b>	Χρησιμοποιεί εφαρμογές που βασίζονται στο πρότυπο MVC, με έναν αριθμό προκατασκευασμένων εξαρτήσεων. Αυτό το καθιστά ελαφρώς λιγότερο ευέλικτο, αλλά πιο φιλικό προς το χρήστη.	Υποστηρίζει το πρότυπο MVC. Χρησιμοποιεί ένα σύνολο επαναχρησιμοποιήσιμων PHP στοιχείων/βιβλιοθηκών. Δεν περιορίζει τους προγραμματιστές στο περιβάλλον του πλαισίου, αλλά τους επιτρέπει να επιλέξουν και να χρησιμοποιήσουν στην εφαρμογή τους μόνο τα στοιχεία που χρειάζονται.
<b>TEMPLATING ENGINE</b>	Η προεπιλεγμένη μηχανή προτύπων είναι το Blade.	Χρησιμοποιεί το Twig ως την προεπιλεγμένη μηχανή προτύπων.
<b>DATABASE SUPPORT</b>	Χρησιμοποιεί αντικειμενοστραφής χαρτογράφηση (ORM) για πρόσβαση σε δεδομένα μέσω του Eloquent. Υποστηρίζει τις ακόλουθες βάσεις δεδομένων: MySQL, PostgreSQL, SQLite και SQL Server.	Χρησιμοποιεί αντικειμενοστραφής χαρτογράφηση (ORM) για πρόσβαση σε δεδομένα μέσω του Doctrine. Υποστηρίζει τις ακόλουθες βάσεις δεδομένων: Drizzle, MySQL, Oracle, PostgreSQL, SAP Sybase SQL Anywhere, SQLite, SQL Server.
<b>DATA MODELING</b>	Απαιτεί ουσιαστική γνώση SQL.	Δεν απαιτεί σημαντική γνώση SQL.

# Laravel vs CodeIgniter

	Laravel	CodeIgniter
<b>Built-in Module</b>	Διαθέτει διάφορες ενσωματωμένες μονάδες (modules), που βοηθούν στη διαδικασία ανάπτυξης	Δεν διαθέτει ενσωματωμένες μονάδες (modules).
<b>Structure</b>	Ακολουθεί το πρότυπο MVC.	Ενθαρρύνει την ανάπτυξη εφαρμογών που βασίζονται στο πρότυπο MVC, αλλά μπορεί επίσης να τροποποιηθεί για να χρησιμοποιηθεί το πρότυπο HMVC.
<b>Database Support</b>	Βάσεις Δεδομένων που υποστηρίζει: MySQL, PostgreSQL, SQLite, SQL Server.	Βάσεις Δεδομένων που υποστηρίζει: MySQL, PostgreSQL, SQLite3
<b>Templates</b>	Χρησιμοποιεί έναν ισχυρό μηχανισμό προτύπων που ονομάζεται Blade.	Δεν συνοδεύεται από ενσωματωμένη μηχανή προτύπων. Οι προγραμματιστές πρέπει να χρησιμοποιήσουν εξωτερικά εργαλεία προτύπων.
<b>Beginners</b>	Δεν είναι εύκολο για τους αρχάριους προγραμματιστές να μάθουν και να χρησιμοποιήσουν το πλαίσιο, λόγω των πολλών δυνατοτήτων που προσφέρει.	Είναι πραγματικά εύκολο για τους αρχάριους προγραμματιστές να μάθουν και να χρησιμοποιήσουν το CodeIgniter, γεγονός που το καθιστά το κορυφαίο PHP πλαίσιο για αρχάριους.

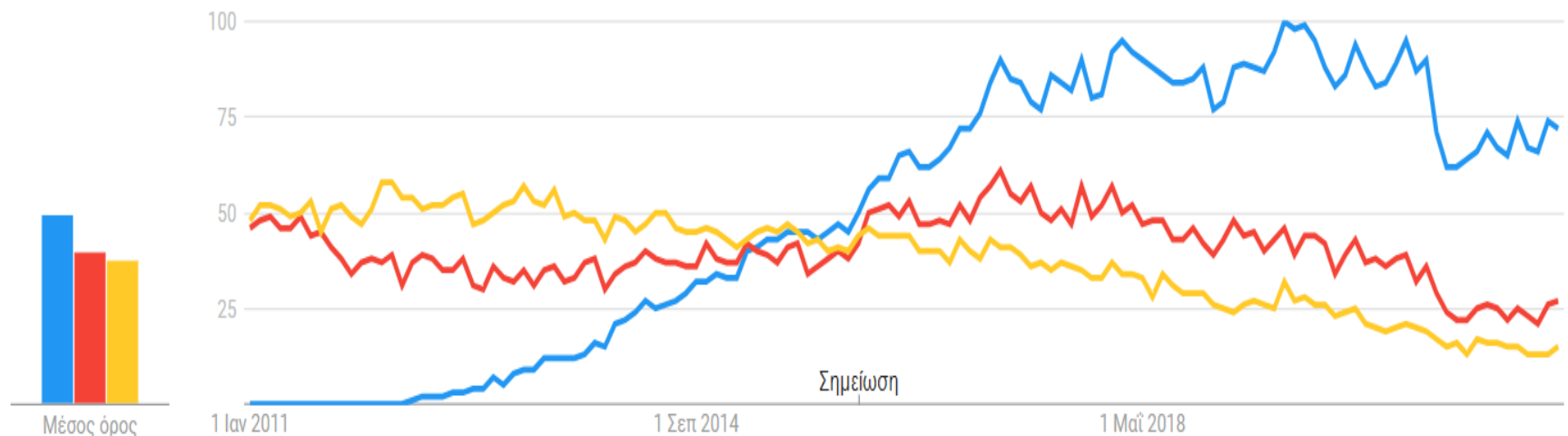
# Συμπεράσματα Σύγκρισης

- Όταν θέλουμε, να αναπτύξουμε εφαρμογές ιστού με εξουσιοδότηση χρήστη, το Laravel μπορεί να μας βοηθήσει πολύ. Χρησιμοποιώντας εργαλεία όπως: Eloquent ORM, Migration και Artisan CLI, μπορούμε να το πετύχουμε εύκολα και γρήγορα, αποφεύγοντας τον πολύ περίπλοκο και πολύ μεγάλο κώδικα.
- Το Symfony απευθύνεται σε μεγάλης κλίμακας και πιο πολύπλοκες εφαρμογές ιστού. Επίσης, στοχεύει σε προχωρημένους προγραμματιστές γιατί είναι λίγο πιο δύσκολο στην εκμάθηση και στην χρήση του, ειδικά στην αρχή.
- το CodeIgniter είναι εύκολο τόσο στην κατανόηση, όσο και στην χρήση. Αν θέλουμε να δημιουργήσουμε μια ελαφριά web εφαρμογή, η οποία να διαθέτει εντυπωσιακά χαρακτηριστικά με κομψή διεπαφή χρήστη, τότε θα πρέπει να επιλέξουμε το CodeIgniter.

# Επιλογή PHP Framework σύμφωνα με τη δημοτικότητα και τις τάσεις της αγοράς

● Laravel ● Symfony ● CodeIgniter

Παγκοσμίως, 1/1/11 - 22/10/21, Υπολογιστές και Ηλεκτρονικές συσκευές



Πηγή: Google Trends (trends.google.com – Σύγκριση 2021)

# Τεχνολογίες που χρησιμοποιήθηκαν

## ➤ Front-end

- HTML
- Bootstrap

## ➤ Back-end

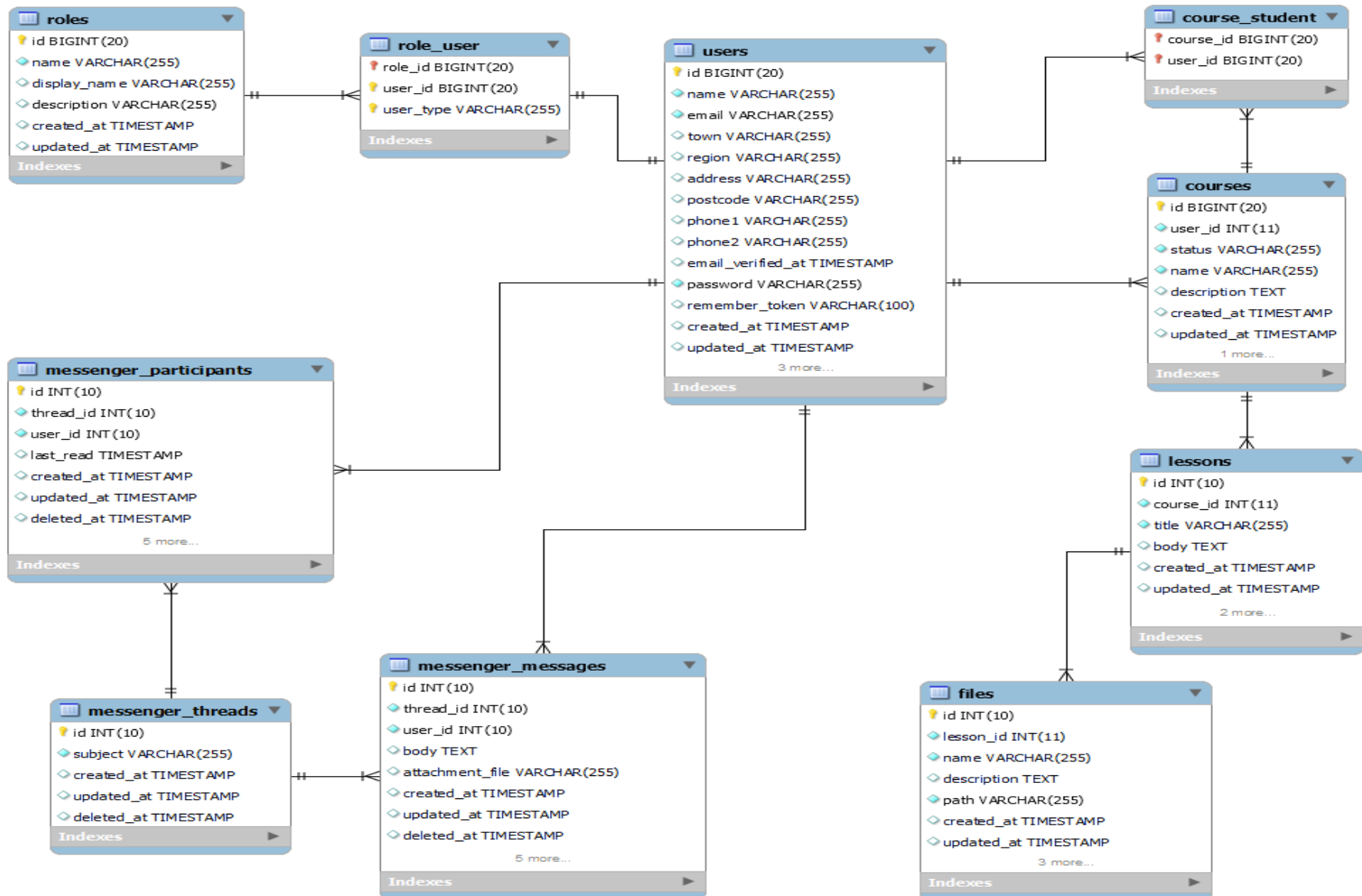
- Laravel
- MySQL

# Βασικά Δομικά Στοιχεία της Εφαρμογής

- Επιπλέον πακέτα (packages) που χρησιμοποιήθηκαν στο Laravel
  - Laravel Breeze
  - Laravel-Breadcrumbs
  - Laravel Messenger
  - Pusher

# Βασικά Δομικά Στοιχεία της Εφαρμογής

## ➤ Βάση Δεδομένων - Διάγραμμα Οντοτήτων-Συσχετίσεων



# Βασικά Δομικά Στοιχεία της Εφαρμογής

## ➤ Διαδρομές (routes)

```
// routes for students
Route::group(['middleware' => ['auth', 'role:student']], function() {
    // enroll
    Route::get('/dashboard/students/enrollment', 'App\Http\Controllers\EnrollmentController@index')
        ->name('enrollment.index');
    Route::put('/dashboard/students/enrollment/{courseId}', [App\Http\Controllers\EnrollmentController::class, 'store'])
        ->name('enrollment.store');
    // courses
    Route::get('/dashboard/students/courses', [StudentsCoursesController::class, 'index'])
        ->name('studentscourses.index');
    Route::get('/dashboard/students/courses/{courses}', [StudentsCoursesController::class, 'show'])
        ->name('studentscourses.show');
    // lessons
    Route::get('/dashboard/students/courses/lessons/{lessonId}', [StudentsLessonsController::class, 'show'])
        ->name('studentslessons.show');
    // files
    Route::get('/dashboard/students/courses/lessons/{lessonId}/files', [StudentsFilesController::class, 'index'])
        ->name('studentsfiles.index');
});
```

# Βασικά Δομικά Στοιχεία της Εφαρμογής

## ➤ Ελεγκτές (controllers)

```
class EnrollmentController extends Controller
{
    public function index()
    {
        $user_id = Auth::user()->id;

        $registeredCourses = DB::table('course_student')
            ->where('course_student.user_id', $user_id)
            ->distinct()
            ->get();

        $availableCourses = DB::table('courses')
            ->join('users', 'users.id', '=', 'courses.user_id')
            ->select('users.name AS teacher_name', 'courses.id',
                'courses.status', 'courses.name AS course_name')
            ->where('courses.status', '=', '1')
            ->paginate(5);

        return view('enrollment.index',
            compact('availableCourses', 'registeredCourses'))
            ->with('i', (request()->input('page', 1) - 1) * 5);
    }

    public function store(Request $request, $courseId)
    {
        $user_id = $request->user()->id;

        DB::table('course_student')->insert([
            'course_id' => $courseId,
            'user_id' => $user_id
        ]);

        return redirect()->route('enrollment.index')
            ->with('message', 'Η εγγραφή στο μάθημα έγινε επιτυχώς');
    }
}
```

# Βασικά Δομικά Στοιχεία της Εφαρμογής

## ➤ Προβολές (views)

```
<body class="hold-transition sidebar-mini">
<!-- Site wrapper -->
<div class="wrapper">
  <!-- Navbar -->
  @include('layouts.partials.header')
  <!-- /.navbar -->

  <!-- Main Sidebar Container -->
  @include('layouts.partials.sidebar')
  <!-- Content Wrapper. Contains page content -->
  <div class="content-wrapper">
    <!-- Content Header (Page header) -->

    <!-- Main content -->
    <section class="content">
      <!-- Default box -->
      @yield('content')
    </section>
    <!-- /.content -->
  </div>
  @include('layouts.partials.footer')
  <!-- /.content-wrapper -->
  <!-- up button -->
  <button onclick="topFunction()" id="myBtn" title="Κορυφή">
    <i class="fa fa-arrow-up" aria-hidden="true"></i>
  </button>
</div>
<!-- ./wrapper -->
```



# Παρουσίαση της Εφαρμογής