



جامعة طرابلس - كلية تقنية المعلومات

قسم هندسة البرمجيات

إعادة استخدام البرمجيات

(ITIS422)

تقرير عن WAFs المبنية على MVC pattern

(ربيع 2024)



قام بهذا العمل الطالب : محمد علي السائح المسجل تحت رقم القيد : 2191801491

والطالب : احمد محمد النجار المسجل تحت رقم القيد : 2191804659

## مقدمة :

مشروع إدارة الطلاب هو تطبيق ويب يستخدم نمط التصميم MVC (Model-View- Controller) لتنظيم ومعالجة البيانات بشكل فعال. يتيح هذا النظام للمستخدمين إدارة معلومات الطلاب، الدورات الدراسية، وسجلات الأداء الأكاديمي بطريقة منظمة وسهلة الاستخدام.

## الأهداف :

- توفير واجهة مستخدم سهلة الاستخدام : لتسهيل عملية إدارة معلومات الطلاب والدورات الدراسية.
- تنظيم البيانات : باستخدام نمط MVC لتنظيم البيانات ومعالجتها بشكل فعال.
- التأكد من التكامل والمرونة : تقديم نظام يمكنه التكيف مع التغيرات المستقبلية بسهولة.
- تحسين الكفاءة : تبسيط عمليات الإدارة الأكاديمية لتوفير الوقت والجهد.

## التقنيات المستخدمة :

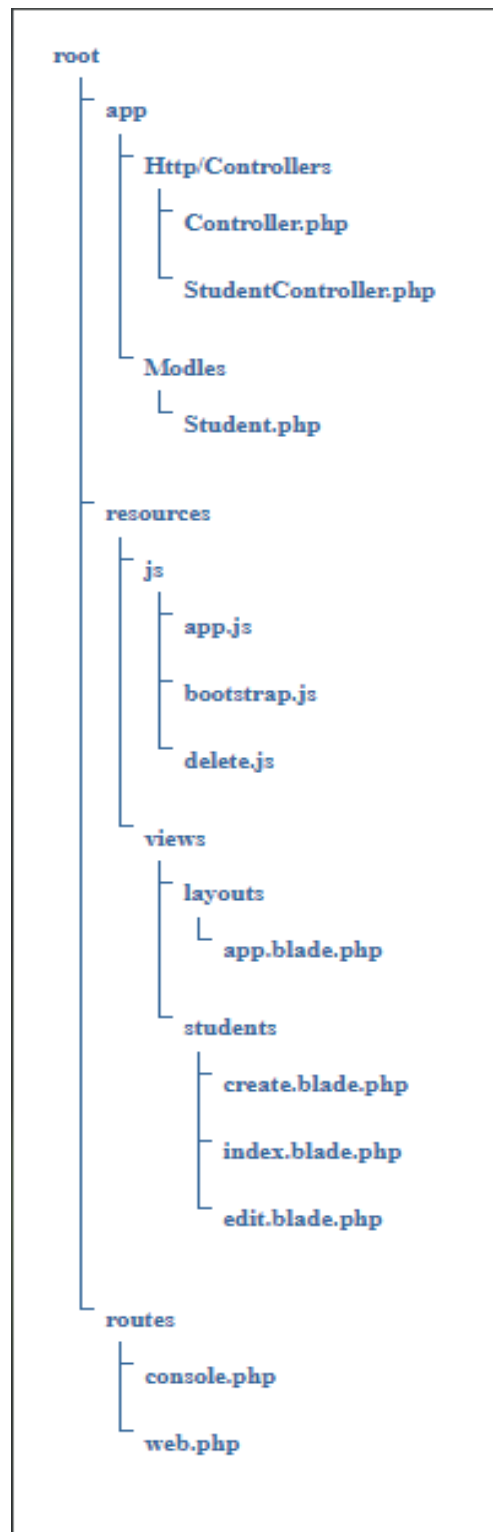
- لغة البرمجة : PHP
- إطار العمل : Laravel
- قاعدة البيانات : SQLite
- أدوات إدارة الحزم : Composer

## نمط MVC :

المشروع يتبع هيكل MVC الذي بدوره ينقسم إلى ثلاثة أجزاء رئيسية:

1. **Model** : يمثل البيانات وقواعد الأعمال. يحتوي على ملفات تعكس بنية قاعدة البيانات مثل ملفات تعريف الطلاب والدورات.
2. **View** : يعرض البيانات للمستخدمين. يحتوي على ملفات HTML التي تقدم الواجهات الرسومية.
3. **Controller** : يعالج المدخلات من المستخدمين ويتحكم في تدفق البيانات بين الـ Model والـ View.

## هيكل المشروع :



- **Model (النموذج)** : يمثل الطبقة المسؤولة عن البيانات والمنطق. في هذا المشروع، النموذج يتعامل مع بيانات الطلاب ويتفاعل مع قاعدة البيانات لإجراء عمليات CRUD (إنشاء، قراءة، تحديث، حذف).

```
app > Models > StudentModel.php
1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7
8  class Student extends Model
9  {
10     use HasFactory;
11     protected $fillable = ['name', 'address', 'age',];
12 }
13
```

- **View (العرض)** : هي الطبقة التي تعرض البيانات للمستخدمين. تتضمن الصفحات التي يراها المستخدم مثل صفحات تسجيل الطلاب وعرض الدورات.

#### • ملف create.blade.php المسؤول عن اضافة طالب جديد

```
resources > views > students > create.blade.php
1  @extends('layouts.app')
2
3  @section('content')
4  <div class="container" style="background: linear-gradient(45deg, #006400, #00BFFF, #FFA500, #87CEEB); padding: 20px; border-radius: 8px;">
5      <h2 class="my-4 text-end">إضافة طالب جديد</h2>
6      <form method="post" action="{{ url('/api/add-student') }}">
7          @csrf
8          <div class="mb-3 text-end">
9              <label for="name" class="form-label">الاسم</label>
10             <input type="text" class="form-control" id="name" name="name" placeholder="ادخل اسم الطالب" required dir="rtl">
11         </div>
12         <div class="mb-3 text-end">
13             <label for="address" class="form-label">العنوان</label>
14             <input type="text" class="form-control" id="address" name="address" placeholder="ادخل عنوان الطالب" required dir="rtl">
15         </div>
16         <div class="mb-3 text-end">
17             <label for="age" class="form-label">العمر</label>
18             <input type="number" class="form-control" id="age" name="age" placeholder="ادخل عمر الطالب" required dir="rtl">
19         </div>
20         <div class="text-center">
21             <button type="submit" class="btn btn-primary btn-lg">إرسال</button>
22         </div>
23     </form>
24 </div>
25 @endsection
26
```

## ○ ملف index.blade.php الذي يحتوى على صفحة العرض

```
resources > views > students > index.blade.php
1  @extends('layouts.app')
2
3  @section('content')
4
5      <div class="container" style="background: linear-gradient(45deg, #006400, #008FFF, #FFA500, #87CEEB); padding: 20px; border-radius: 8px;">
6          <div class="mt-3 d-flex justify-content-center mb-3">
7              <button class="btn btn-danger me-2" onclick="deleteAllStudents()">حذف كل الطلاب</button>
8              <a class="btn btn-primary ms-2" href="/add-student">إضافة طالب جديد</a>
9          </div>
10         @if(!is_null($students) && count($students) > 0)
11             <div class="table-responsive">
12                 <table class="table table-bordered text-center table-success">
13                     <thead>
14                         <tr>
15                             <th>المعر</th>
16                             <th>العنوان</th>
17                             <th>الاسم</th>
18                             <th>Num</th>
19                         </tr>
20                     </thead>
21                     <tbody>
22                         @foreach($students as $index => $student)
23                             <tr>
24                                 <td>{{ $student->age }}</td>
25                                 <td>{{ $student->address }}</td>
26                                 <td>{{ $student->name }}</td>
27                                 <td>{{ $index + 1 }}</td>
28                             </tr>
29                         @endforeach
30                     </tbody>
31                 </table>
32             </div>
33         @else
34             <div class="text-center">
35                 <p>No students found.</p>
36             </div>
37         @endif
38         <div class="mt-3 d-flex justify-content-center mb-3">
39             <a href="/edit-student/{{ $student->id }}" class="btn btn-warning me-2">تعديل بيانات الطالب</a>
40         </div>
41     </div>
42     <script src="https://cdn.jsdelivr.net/npm/axios/dist/axios.min.js"></script>
43
44     <script>
45     function deleteStudent(id) {
46         axios
47             .post('/api/deleteStudent/{{ $id }}')
48             .then((response) => {
49                 if (response.data.success) {
50                     alert(response.data.message);
51                     location.reload();
52                 } else {
53                     alert("Failed to delete student: " + response.data.message);
54                 }
55             })
56             .catch((error) => {
57                 console.error("Error:", error);
58                 alert("An error occurred while deleting the student.");
59             });
60     }
61
62     function deleteAllStudents() {
63         axios
64             .post("/api/deleteAll")
65             .then((response) => {
66                 if (response.data.success) {
67                     alert(response.data.message);
68                     location.reload();
69                 } else {
70                     alert("Failed to delete students: " + response.data.message);
71                 }
72             })
73             .catch((error) => {
74                 console.error("Error:", error);
75                 alert("An error occurred while deleting students.");
76             });
77     }
78     </script>
79 @endsection
80
```

## • ملف edit.blade.php المسؤول عن تعديل بيانات الطلاب الموجودين

```
resources > views > students > edit.blade.php
1  @extends('layouts.app')
2
3  @section('content')
4  <div class="container" style="background: linear-gradient(45deg, #006400, #00BFFF, #FFA500, #87CEEB); padding: 20px; border-radius: 8px;">
5      <h2 class="my-4 text-end">تعديل بيانات الطالب</h2>
6      <form method="post" action="{{ url('/api/update-student') }}">
7          @csrf
8          <div class="mb-3 text-end">
9              <label for="student_id" class="form-label">رقم الطالب</label>
10             <input type="text" class="form-control" id="student_id" name="student_id" placeholder="ادخل رقم الطالب" required dir="rtl">
11         </div>
12         <div class="mb-3 text-end">
13             <label for="name" class="form-label">الاسم الجديد</label>
14             <input type="text" class="form-control" id="name" name="name" placeholder="ادخل الاسم الجديد للطالب" required dir="rtl">
15         </div>
16         <div class="mb-3 text-end">
17             <label for="address" class="form-label">العنوان الجديد</label>
18             <input type="text" class="form-control" id="address" name="address" placeholder="ادخل العنوان الجديد للطالب" required dir="rtl">
19         </div>
20         <div class="mb-3 text-end">
21             <label for="age" class="form-label">العمر الجديد</label>
22             <input type="number" class="form-control" id="age" name="age" placeholder="ادخل العمر الجديد للطالب" required dir="rtl">
23         </div>
24         <div class="text-center">
25             <button type="submit" class="btn btn-primary btn-lg">حفظ التعديلات</button>
26         </div>
27     </form>
28 </div>
29 @endsection
30
```

- **Controller (المتحكم)** : يعمل كوسيط بين النموذج والعرض. يعالج طلبات المستخدمين، يستدعي النموذج المناسب، ويختار العرض المناسب لعرض البيانات.

## ملف StudentController.php

```
app > Http > Controllers > StudentController.php
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6  use App\Models\Student;
7
8  class StudentController extends Controller
9  {
10     // عرض قائمة الطلاب
11     public function index()
12     {
13         $students = Student::all();
14         return view('students.index', compact('students'));
15     }
16
17     // عرض صفحة إضافة طالب جديد
18     public function create()
19     {
20         return view('students.create');
21     }
22
23     // تخزين بيانات الطالب الجديد
24     public function store(Request $request)
25     {
26         // التحقق من صحة البيانات المدخلة
27         $request->validate([
28             'name' => 'required',
29             'address' => 'required',
30             'age' => 'required|integer',
31         ]);
32
33         // إنشاء الطالب في قاعدة البيانات
34         Student::create($request->only(['name', 'address', 'age']));
35
36         // إعادة التوجيه مع رسالة نجاح
37         return redirect('/')->with('success', 'Student added successfully.');
```

## ○ باقي ملف StudentController.php

```
app > Http > Controllers > StudentController.php
9  {
56
57  // عرض نموذج تعديل بيانات الطالب
58  public function edit($id)
59  {
60      $student = Student::find($id);
61
62      if (!$student) {
63          return redirect('/')->with('error', 'Student not found.');
```

```
64      }
65
66      return view('students.edit', compact('student'));
67  }
68
69  // تحديث بيانات الطالب
70  public function update(Request $request)
71  {
72      $request->validate([
73          'student_id' => 'required|exists:students,id',
74          'name' => 'required',
75          'address' => 'required',
76          'age' => 'required|integer',
77      ]);
78
79      $student = Student::find($request->student_id);
80
81      if (!$student) {
82          return redirect('/')->with('error', 'Student not found.');
```

```
83      }
84
85      $student->name = $request->name;
86      $student->address = $request->address;
87      $student->age = $request->age;
88      $student->save();
89
90      return redirect('/')->with('success', 'Student updated successfully.');
```

```
91  }
92
93  // حذف جميع الطلاب
94  public function deleteAll()
95  {
96      try {
97          Student::truncate();
98          return response()->json(['success' => true, 'message' => 'Database has been cleaned.']);
99      } catch (\Exception $e) {
100          return response()->json(['success' => false, 'message' => 'Failed to delete students.'], 500);
101      }
102  }
103  }
104
```

- **Routes (المسارات) :** هي التي تحدد كيف يتم توجيه الطلبات من المتصفح أو التطبيق إلى الوظائف المناسبة في الـ Controllers. ملف web.php الذي يحتوي على المسارات

```
routes > web.php
1  <?php
2
3  use App\Http\Controllers\StudentController;
4  use Illuminate\Support\Facades\Route;
5
6  Route::get('/', [StudentController::class, 'index']);
7  Route::get('/add-student', [StudentController::class, 'create']);
8  Route::post('/api/add-student', [StudentController::class, 'store']);
9  Route::post('/api/deleteStudent{id}', [StudentController::class, 'delete']);
10 Route::post('/api/deleteAll', [StudentController::class, 'deleteAll']);
11 Route::get('/edit-student/{id}', [StudentController::class, 'edit']);
12 Route::post('/api/update-student', [StudentController::class, 'update']);
13
```

## النتائج :

MVC Pattern

Home

إضافة طالب جديد

حذف كل الطلاب

Num	الاسم	العنوان	العمر
1	محمد علي السائح	عين زارة	22
2	احمد محمد النجار	سوق الجمعة	23
3	عبد الحميد الواعر	جامعة طرابلس	50

تعديل بيانات الطلاب

MVC Pattern

Home

إضافة طالب جديد

الاسم :

ادخل اسم الطالب

العنوان :

ادخل عنوان الطالب

العمر :

ادخل عمر الطالب

إرسال

MVC Pattern

Home

تعديل بيانات الطالب

رقم الطالب :

ادخل رقم الطالب

الاسم الجديد :

ادخل الاسم الجديد للطالب

العنوان الجديد :

ادخل العنوان الجديد للطالب

العمر الجديد :

ادخل العمر الجديد للطالب

حفظ التعديلات

## الخاتمة :

نمط تصميم (MVC (Model-View-Controller يعتبر من أكثر الأنماط فعالية لتنظيم وتطوير تطبيقات الويب. هذا مشروع يوضح كيفية تطبيق هذا النمط بشكل متكامل لتحقيق تنظيم عالٍ وسهولة في الصيانة والتطوير المستقبلي. من خلال فصل البيانات، الواجهة، والمنطق التجاري، يوفر المشروع بنية مرنة قابلة لإعادة الاستخدام حيث يمكن تعديلها وتطويرها بمرور الوقت لتلبية الاحتياجات المتغيرة.