

تطوير موقع لأرشفة مشاريع التخرج لقسم تقنية معلومات

■ متطلبات المستخدم User requirements

من متطلبات المستخدم مايلي:

■ تسجيل البيانات الأساسية للموقع :

1. تخزين كافة بيانات القسم مع إمكانية التعديل
2. تخزين كافة بيانات البرامج مع إمكانية حذفها وتعديلها أو الاستعلام عنها.
3. تخزين بيانات العام الدراسي مع إمكانية الحذف والتعديل والعرض.
4. تخزين كافة بيانات المشرفين مع إمكانية الحذف والتعديل والعرض.
5. تخزين كافة بيانات المشاريع ورفع ملفات المشاريع مع إمكانية الحذف والتعديل والعرض
6. تخزين بيانات أخبار القسم مع إمكانية التعديل والحذف والاستعلام عنها.
7. إعطاء الصلاحيات وكلمات المرور لمستخدمي لوحة تحكم الموقع

■ متطلبات الطالب:

1. الاطلاع على مشاريع القسم بمختلف التخصصات
2. إمكانية تحميل المشاريع

■ متطلبات الصفحات الرئيسية للموقع:

1. عرض كافة بيانات القسم
2. عرض أحدث أخبار القسم
3. عرض التخصصات التابعة للقسم
4. عرض جميع المشاريع الخاص بالقسم مع فلترتها حسب الأعوام الدراسية والبرامج وأيضا المشرفين

5. عرض معلومات عن أعضاء هيئة التدريس أي المشرفين على المشاريع في القسم
6. إمكانية التواصل مع القسم عبر الموقع

■ التقنيات واللغات المستخدمة في بناء الموقع:

1. برنامج EdrawMax

2. HTML

3. CSS

Laravel=>web project

.4 JAVASCRIPT

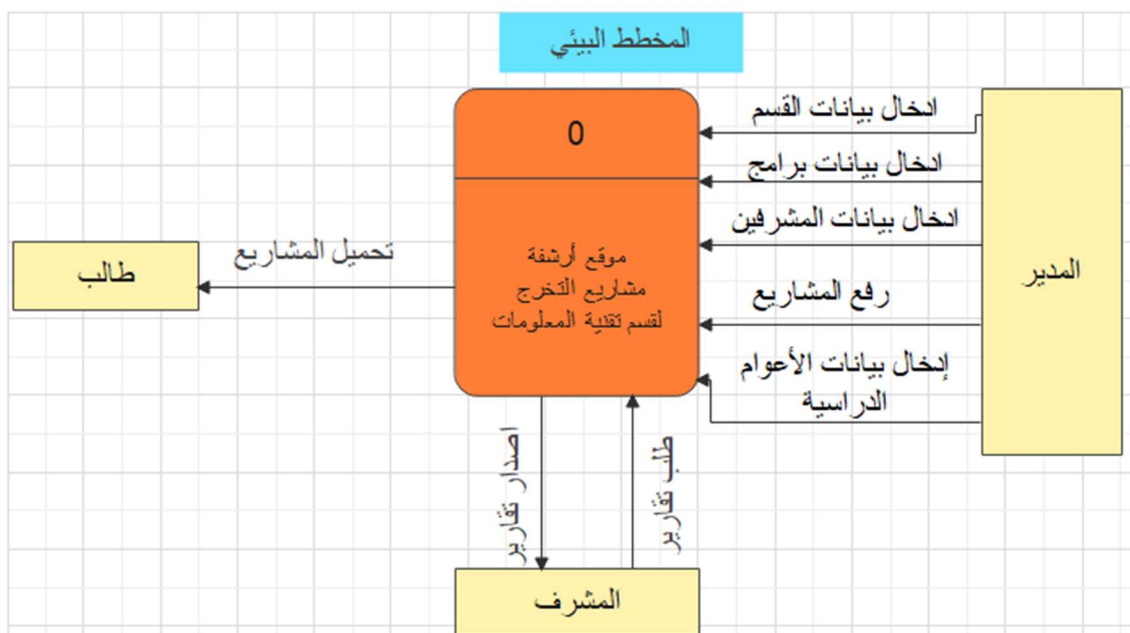
.5 PHP-LARAVEL Framework

.6 Bootstrap library

.7 JQuery library

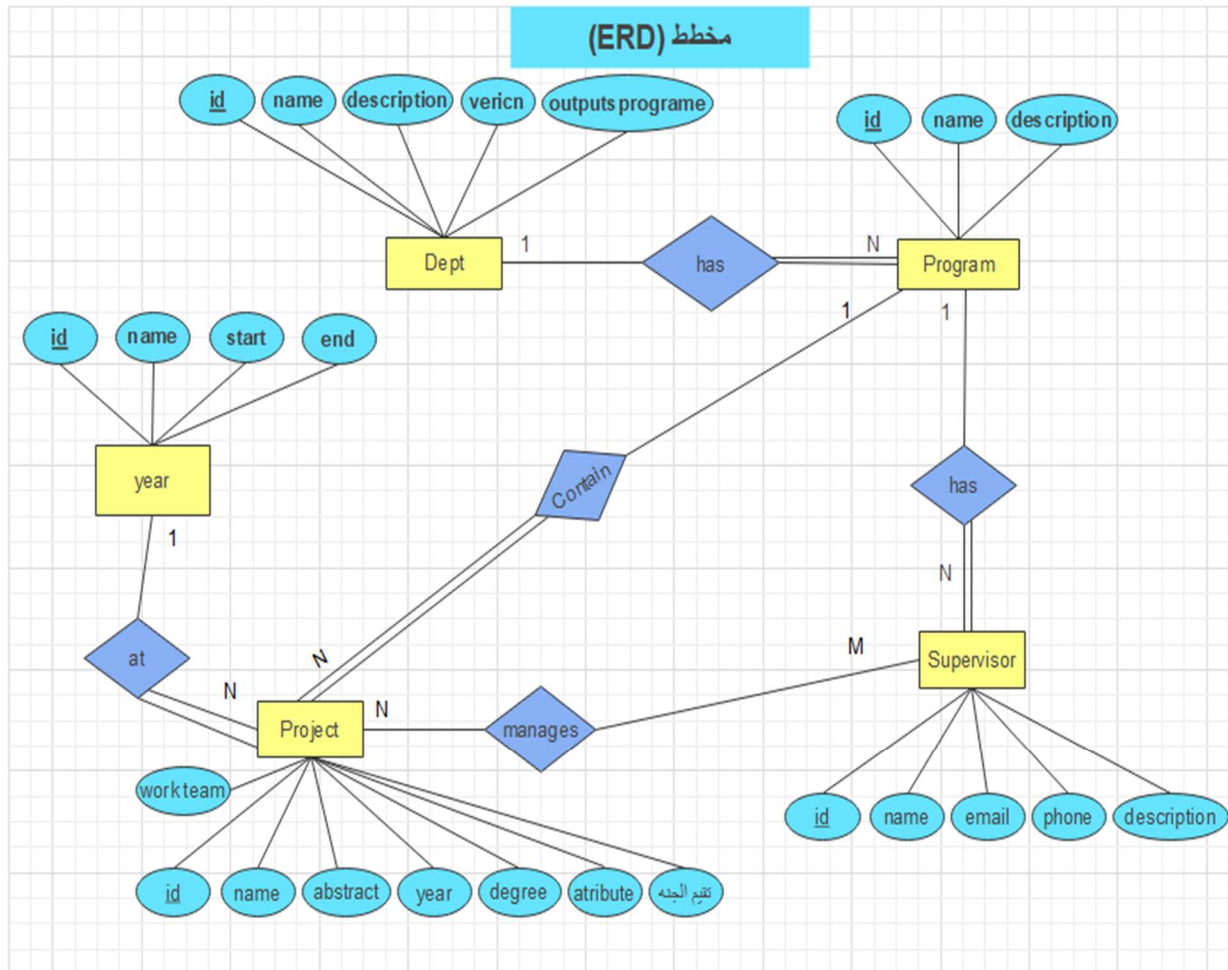
.8 ...

■ المخطط البيئي لتدفق البيانات : Context Data Flow Diagram

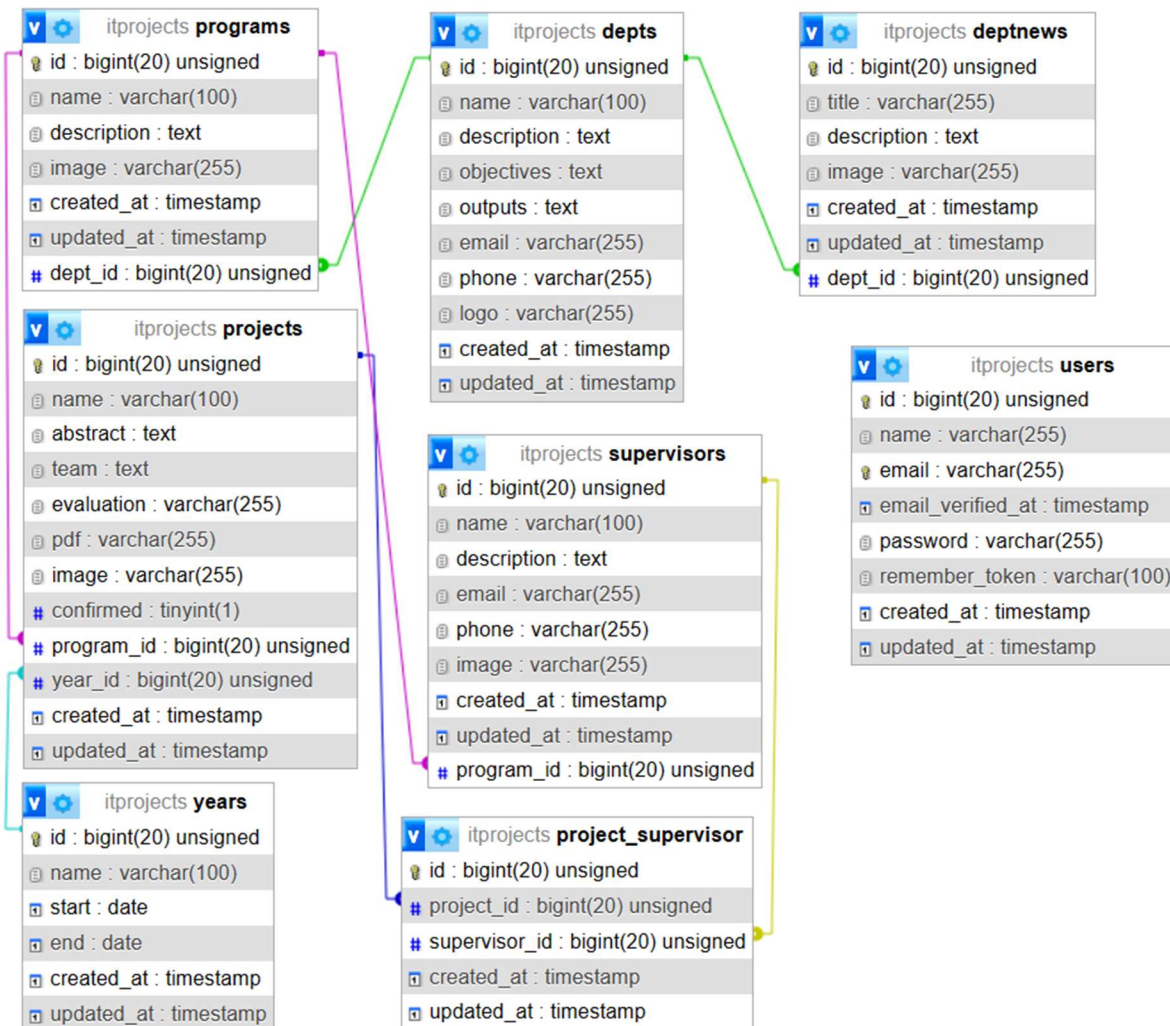


■ المخطط الصفري

■ مخطط العلاقة ERD : The ERD relationship sechema ERD



■ مخطط الاسكيميا : Schema of Iskima



خطوات التطبيق العملي :

1. انشاء مشروع لارفل باسم ITProjects
 2. انشاء قاعدة بيانات باسم ITprojects باستخدام MySQL
 3. تصميم قاعدة البيانات حسب مخطط ERD بعد تحويله الى مخطط Schema
- يمكننا انشاء الجداول من phpMyAdmin والربط بين الجداول او استخدام migration في مشروع لارافل
- نفتح مشروع لارافل ونبدأ بتعديل ملف .env. للاتصال بقاعدة البيانات

Laravel=>web project

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3307
DB_DATABASE=ITprojects
DB_USERNAME=root
DB_PASSWORD=
```

في terminal نكتب الامر التالي لإنشاء جدول الأقسام مع النموذج ووحدة التحكم
(table,model,controller)

```
php artisan make:model Dept -mcr
```

نلاحظ إنشاء جدول باسم depts ونموذج باسم Dept ووحدة تحكم باسم DeptController
إضافة حقول جدول الأقسام depts

```
Schema::create('depts', function (Blueprint $table) {
    $table->id();
    $table->string('name',100);
    $table->text('description')->nullable();
    $table->text('objectives')->nullable();
    $table->text('outputs')->nullable();
    $table->string('email');
    $table->string('phone');
    $table->string('logo');
    $table->timestamps();
});
```

نكرر الامر السابق لإنشاء بقية الجداول مع النماذج ووحدات التحكم ومن ثم نضيف حقول الجدول كما يلي:
إضافة حقول جدول البرامج

```
Schema::create('programs', function (Blueprint $table) {
    $table->id();
    $table->string('name',100);
    $table->text('description')->nullable();
    $table->string('image');
    $table->timestamps();
    $table->foreignId('dept_id')->
    >constrained('depts')->cascadeOnDelete();
});
```

إضافة حقول جدول الأعوام الدراسية

```
Schema::create('years', function (Blueprint $table) {
    $table->id();
    $table->string('name',100);
    $table->date('start');
    $table->date('end');
    $table->timestamps();
});
```

إضافة حقول جدول المشرفين

```
Schema::create('supervisors', function (Blueprint $table) {
    $table->id();
    $table->string('name',100);
    $table->text('description')->nullable();
    $table->string('email');
    $table->string('phone');
    $table->string('image');
    $table->timestamps();
    $table->foreignId('program_id')-
>constrained('programs')->cascadeOnDelete();
});
```

إضافة حقول جدول المشاريع

```
Schema::create('projects', function (Blueprint $table) {
    $table->id();
    $table->string('name',100);
    $table->text('abstract');
    $table->text('team')->nullable();
    $table->string('evaluation');//التقييم
    $table->string('pdf');
    $table->string('image');
    $table->boolean('confirmed')->default(false);
    $table->foreignId('program_id')-
>constrained('programs')->cascadeOnDelete();
    $table->foreignId('year_id')-
>constrained('years')->cascadeOnDelete();
    $table->timestamps();
});
```

إضافة حقول جدول العلاقة (المشرفين-المشروع)

```
Schema::create('project_supervisor', function (Blueprint
$table) {
    $table->id();
    $table->foreignId('project_id')-
>constrained('projects');
    $table->foreignId('supervisor_id')-
>constrained('supervisors');
    $table->timestamps();
});
```

إضافة حقول جدول أخبار القسم

```
Schema::create('deptnews', function (Blueprint $table) {
    $table->id();
    $table->string('title');
    $table->text('description')->nullable();
    $table->string('image');
    $table->timestamps();
    $table->foreignId('dept_id')-
>constrained('depts')->cascadeOnDelete();
});
```

نرحل الجداول الى قاعدة البيانات باستخدام الامر `php artisan migrate`

العلاقات بين الجداول one to many or many to many نستخدم دوال في Models

في نموذج Dept نضيف التالي

```
protected $fillable=['name','description',
'objectives','outputs','email','phone','logo'];
public function programs()
{
    return $this->hasMany(Program::class,'dept_id');
}
public function deptnews()
{
    return $this->hasMany(Deptnew::class,'dept_id');
}
```

في نموذج Program

```
protected $fillable=['name','description','image'];
public function dept()
{
    return $this->belongsTo(Dept::class);
}
public function supervisors()
{
    return $this->hasMany(Supervisor::class,'program_id');
}
public function projects()
{
    return $this->hasMany(Project::class,'program_id');
}
```

في نموذج Year

```
protected $fillable=['name','start','end'];
public function projects()
{
    return $this->hasMany(Project::class,'year_id');
}
```

في نموذج Supervisor

```
protected $fillable=['name','description',
'email','phone','image','program_id'];
public function program()
{
    return $this->belongsTo(Program::class);
}
public function projects()
{
    return $this->belongsToMany(Project::class);
}
```

في نموذج Project

```
protected $fillable=['name','abstract','team',
'evaluation','pdf','image','confirmed',
'program_id','year_id'];
public function program()
{
}
```


Laravel=>web project

```
        return $this->belongsTo(Program::class);
    }
    public function year()
    {
        return $this->belongsTo(Year::class);
    }
    public function supervisors()
    {
        return $this->belongsToMany(Supervisor::class);
    }
}
```

في نموذج Deptnew

```
protected
$fillable=['title','description','image','dept_id'];
public function dept()
{
    return $this->belongsTo(Dept::class);
}
```