



المواصفات الفنية □

## نظام التسجيل والاعتماد

Technical Specification Document  
Registration & Accreditation System

2025

## Table of Contents

|   |    |
|---|----|
| التصور الإداري لنظام التسجيل والاعتماد .....                            | 4  |
| الهدف من النظام .....   | 4  |
| الفئات المستفيدة .....  | 4  |
| Workflow(الإداري) دورة العمل .....                                      | 4  |
| السياسات والإجراءات .....   | 4  |
| التحديات المحتملة .....   | 5  |
| الحلول المقترحة .....   | 5  |
| المخرجات الرئيسية للنظام .....  | 5  |
| المواصفات الفنية لنظام التسجيل والاعتماد .....                          | 6  |
| مقدمة .....   | 6  |
| (Functional Requirements) المتطلبات الوظيفية .....                      | 6  |
| (System Architecture) البنية التقنية .....                              | 7  |
| (Database Schema) قاعدة البيانات .....                                  | 7  |
| (Workflows) تدفقات العمل .....  | 8  |
| (UI/UX Requirements) الواجهات .....                                     | 9  |
| (Non-Functional Requirements) المتطلبات غير الوظيفية .....              | 9  |
| (Development Plan) خطة التنفيذ .....                                    | 9  |
| تصور مقترح للشاشات - نظام تسجيل اللاعبين في دوري الجامعات المصرية ..... | 10 |
| UX تجربة المستخدم : .....   | 20 |
| الهدف .....   | 20 |
| المكونات الأساسية: .....  | 20 |
| (Form Modularization) تقسيم النماذج .....                               | 20 |
| (High-level Workflow) تدفق العمل .....                                  | 21 |
| (SQL) نموذج قاعدة البيانات .....  | 21 |
| مقترحة — شاشات رئيسية ووظائفها (UI) واجهات .....                        | 24 |
| (API) واجهة برمجة التطبيقات — .....                                     | 25 |
| (Validation Rules) قواعد التحقق .....                                   | 25 |
| إدارة المستندات والتحقق .....   | 26 |
| (RBAC) صلاحيات المستخدم .....   | 26 |
| الأمن وحماية البيانات .....   | 26 |
| الأداء والقابلية للتوسع .....   | 27 |
| وقبول النظام (QA) اختبارات .....  | 27 |

|   |    |
|---|----|
| تقارير ومخرجات نهائية .....   | 27 |
| مميزات أساسية خطة تنفيذ مقترحة .....  | 27 |
| الواجهات الرئيسية المقترحة .....  | 28 |
| (HTML + React form) نسخة مُبسّطة للنظام .....                                 | 32 |
| إعادة طرح المقترح ليتناسب مع دورة الألعاب الجامعية الإفريقية – مصر 2026 ..... | 47 |
| الهدف من النظام .....   | 47 |
| الفئات المستفيدة .....  | 47 |
| دورة العمل (Workflow) .....   | 47 |
| التكيف الإفريقي (مقارنة بالنسخة المصرية) .....                                | 48 |
| المخرجات الرئيسية .....   | 48 |
| البنية التقنية .....  | 49 |
| خطة التنفيذ (مقترحة لمصر 2026) .....  | 49 |

## التصور الإداري لنظام التسجيل والاعتماد

### الهدف من النظام

- توحيد عملية تسجيل الرياضيين والإداريين في البطولات الجامعية.
- ضمان الشفافية والمصداقية من خلال توثيق المستندات واعتمادها إلكترونياً.
- تسهيل إدارة البيانات (فرق – طلاب – حكام – مدربين).
- تسريع إجراءات الاعتماد وتجنب الازدواجية أو الأخطاء اليدوية.

### الفئات المستفيدة

1. الطلاب الرياضيون: تقديم بياناتهم والمستندات الأكاديمية/الطبية.
2. المدربون والإداريون: تسجيل الفرق وتأكيد مشاركات الطلاب.
3. الجامعات: مراجعة واعتماد تسجيلات طلابها.
4. الاتحاد/اللجنة المنظمة: الإشراف على العملية وضمان صحة البيانات.
5. الحكام والمتطوعون: إدخال بياناتهم واعتمادهم للمشاركة.

### دورة العمل الإداري (Workflow)

1. فتح باب التسجيل: إعلان رسمي يوضح المواعيد والمتطلبات.
2. تسجيل الطلاب: رفع البيانات + الوثائق المطلوبة (بطاقة جامعية – شهادة قيد – كشف طبي).
3. مراجعة الجامعات: تعتمد أو ترفض طلبات طلابها.
4. اعتماد اللجنة المنظمة: التحقق النهائي وإصدار بطاقات الاعتماد (Accreditation).
5. إدارة التعديلات: فتح نافذة محددة لتعديلات أو استبدال اللاعبين.
6. إصدار القوائم النهائية: قائمة الفرق/الأفراد المعتمدة للمشاركة.

### السياسات والإجراءات

- مستندات إلزامية: بطاقة الرقم القومي + بطاقة جامعية + شهادة قيد + كشف طبي.
- مواعيد صارمة: أي طلب خارج المواعيد الرسمية لا يقبل إلا بقرار خاص.
- اعتماد متعدد المستويات: الطالب → الجامعة → اللجنة المنظمة.
- صلاحيات واضحة: كل جهة ترى وتعديل ما يخصها فقط.

## نظام التسجيل والاعتماد

### التحديات المحتملة

- إدخال بيانات غير دقيقة أو ناقصة.
- تأخر الجامعات في اعتماد طلبات طلابها.
- ازدواجية التسجيل أو محاولات الغش.
- الضغط على النظام في الأيام الأخيرة.

### الحلول المقترحة

- رسائل تنبيه/إشعارات تلقائية بالمواعيد النهائية.
- واجهة بسيطة للطلاب + تدريب إداري للجامعات.
- آليات تحقق تلقائي (مقارنة الرقم القومي – قاعدة بيانات جامعية).
- خطة دعم فني (Help Desk) خلال فترة التسجيل.

### المخرجات الرئيسية للنظام

- قاعدة بيانات شاملة للمشاركين.
- بطاقات اعتماد إلكترونية/مطبوعة.
- تقارير فورية (أعداد – جنسيات – أعمار – مشاركات).
- سجلات اعتماد رسمية يمكن الرجوع إليها بعد البطولة.

## المواصفات الفنية لنظام التسجيل والاعتماد

### مقدمة

- اسم النظام: نظام التسجيل والاعتماد للبطولات الجامعية.
- الهدف: أتمتة عملية تسجيل واعتماد المشاركين (طلاب – مدربين – إداريون – حكام – متطوعون) لضمان المصداقية، السرعة، وسهولة إدارة البيانات.
- المستخدمون المستهدفون: الطلاب، مسؤولو الجامعات، اللجنة المنظمة، الحكام، المتطوعون.

## المتطلبات الوظيفية (Functional Requirements)

### أ) وحدة تسجيل المستخدمين (الجامعات)

- تسجيل حساب جديد (بريد إلكتروني / كلمة مرور).
- رفع المستندات المطلوبة (بطاقة جامعية، شهادة قيد، صورة شخصية، كشف طبي).
- تحديث البيانات الشخصية والأكاديمية.

### ب) وحدة المراجعة الفنية (الاتحاد)

- استعراض طلبات التسجيل الخاصة بالطلاب.
- اعتماد / رفض الطلبات.

### ج) وحدة الاعتماد (الاتحاد)

- مراجعة واعتماد نهائي لكل الطلبات.
- إدارة قوائم الفرق النهائية.
- إصدار بطاقات الاعتماد.

### د) وحدة المتطوعين

- تقديم الطلبات (بيانات + مؤهلات).
- اعتمادهم من اللجنة المنظمة.

### هـ) تقارير وإحصائيات

- إحصائيات المشاركين حسب الجامعة، الدولة، الجنس، العمر.
- قوائم الفرق المعتمدة.
- تتبع حالة كل طلب (معلق – مقبول – مرفوض).

## البنية التقنية (System Architecture)

• نوع النظام: Web-based + Mobile responsive.

• هيكل:

- **Front-end:** ReactJS / Angular (واجهة المستخدم).
- **Back-end:** Node.js أو Django (API + أعمال).
- **قاعدة بيانات:** PostgreSQL أو MySQL.
- **استضافة:** Cloud (AWS / Azure) أو سيرفر محلي.

• **التكامل:** إمكانية الربط مع أنظمة أخرى عبر API.

## قاعدة البيانات (Database Schema)

**جدول Users:**

- UserID (PK)
- FullName
- Email
- PasswordHash
- Role (Student, Coach, Admin, Referee, Volunteer, UniversityRep)

**جدول Students:**

- StudentID (FK → Users)
- UniversityID (FK → Universities)
- SportID (FK → Sports)
- Documents (ID, EnrollmentProof, MedicalReport, Photo)
- Status (Pending, Approved, Rejected)

## جدول Universities :

- UniversityID (PK)
- Name
- Country
- RepresentativeID (FK → Users)

## جدول Sports :

- SportID (PK)
- Name
- Category (Individual/Team)

## جدول Accreditation :

- AccID (PK)
- UserID (FK → Users)
- AccreditationType (Athlete, Coach, Referee, Volunteer, Official)
- IssueDate
- QRCode / Barcode

## تدفقات العمل (Workflows)

### تدفق تسجيل الطالب

1. الجامعة → إنشاء حساب → إدخال بيانات + رفع مستندات.
2. القطاع الفني → مراجعة → اعتماد / رفض.
3. الاعتماد → مراجعة نهائية → اعتماد وإصدار بطاقة.

### تدفق اعتماد الجامعات

- لكل جامعة لوحة تحكم خاصة بها لمراجعة طلبات طلابها فقط.

### تدفق إصدار الاعتماد

- بعد الاعتماد النهائي → النظام يولد بطاقة (PDF + QR Code)

## نظام التسجيل والاعتماد

### الواجهات (UI/UX Requirements)

- واجهة الجامعة: تسجيل بيانات + متابعة حالة الطلب.
- واجهة القطاع الفني: مراجعة جماعية + اعتماد/رفض الطلبات.
- واجهة الاعتماد: إشراف كامل + تقارير + إصدار الاعتمادات.
- واجهة المتطوعين: تسجيل بيانات + متابعة الحالة.
- لوحة تقارير: **Dashboard** إحصائيات ورسوم بيانية لحظية.

### المتطلبات غير الوظيفية (Non-Functional Requirements)

- الأمان: تشفير SSL ، تشفير كلمات المرور. (BCrypt/SHA256)
- الأداء: دعم 10,000+ مستخدم متزامن.
- التوافق: دعم جميع المتصفحات الحديثة + الهواتف الذكية.
- التوافر Uptime: لا يقل عن 99.5%.
- النسخ الاحتياطي: يومي + استعادة فورية عند الطوارئ.

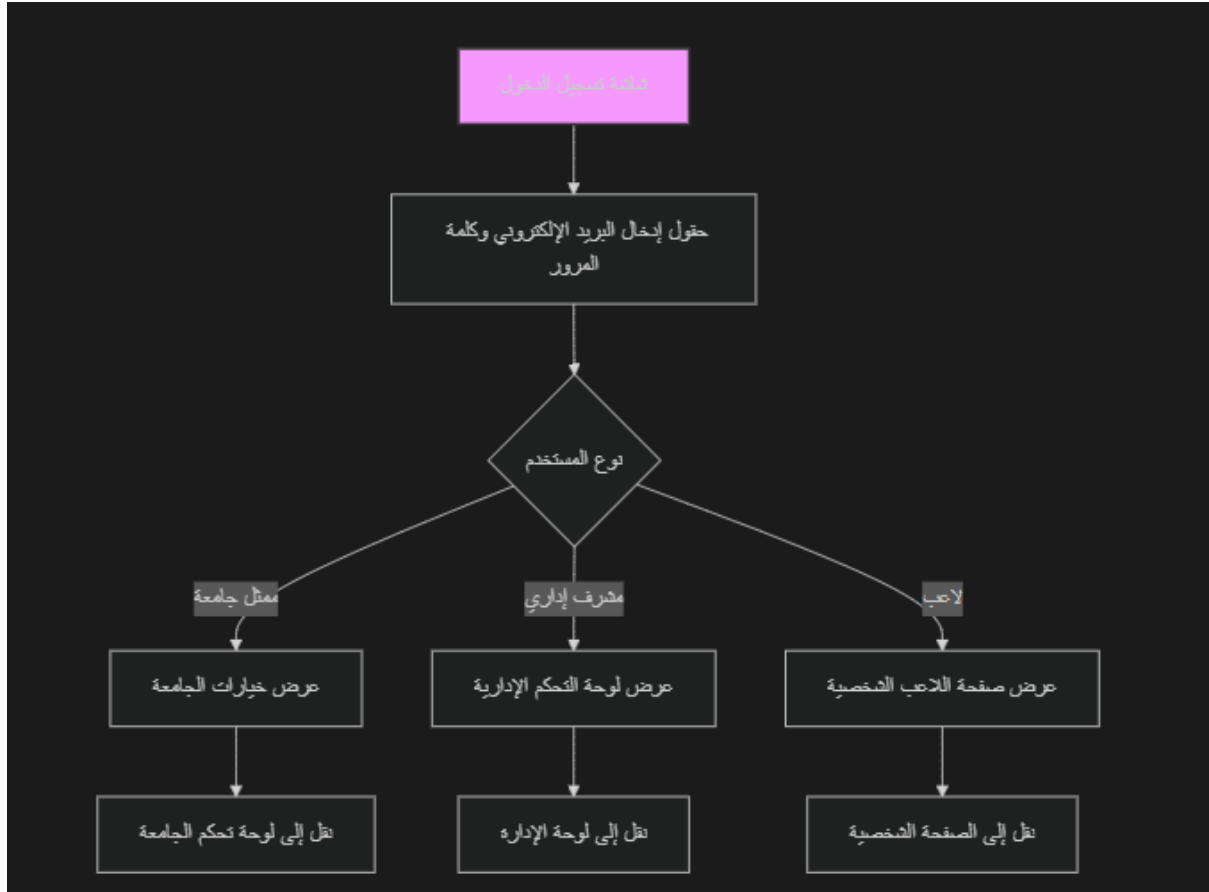
### خطة التنفيذ (Development Plan)

- المرحلة 1: تحليل المتطلبات + تصميم قاعدة البيانات.
- المرحلة 2: تطوير الوحدات الأساسية (تسجيل – اعتماد الجامعات).
- المرحلة 3: تطوير لوحة الإدارة والتقارير.
- المرحلة 4: اختبار أمني + اختبار ضغط.
- المرحلة 5: إطلاق تجريبي (Pilot) مع جامعة واحدة.
- المرحلة 6: الإطلاق الرسمي + تدريب المستخدمين.

## تصور مقترح للشاشات - نظام تسجيل اللاعبين في دوري الجامعات المصرية

1. شاشات المصادقة وإدارة المستخدمين

### 1.1 شاشة تسجيل الدخول (Login Screen)



### 1.2 شاشة استعادة كلمة المرور (Password Recovery)

- حقول إدخال البريد الإلكتروني
- زر "إرسال رابط الاستعادة"
- رسالة تأكيد بإرسال الرابط
- رابط العودة لتسجيل الدخول

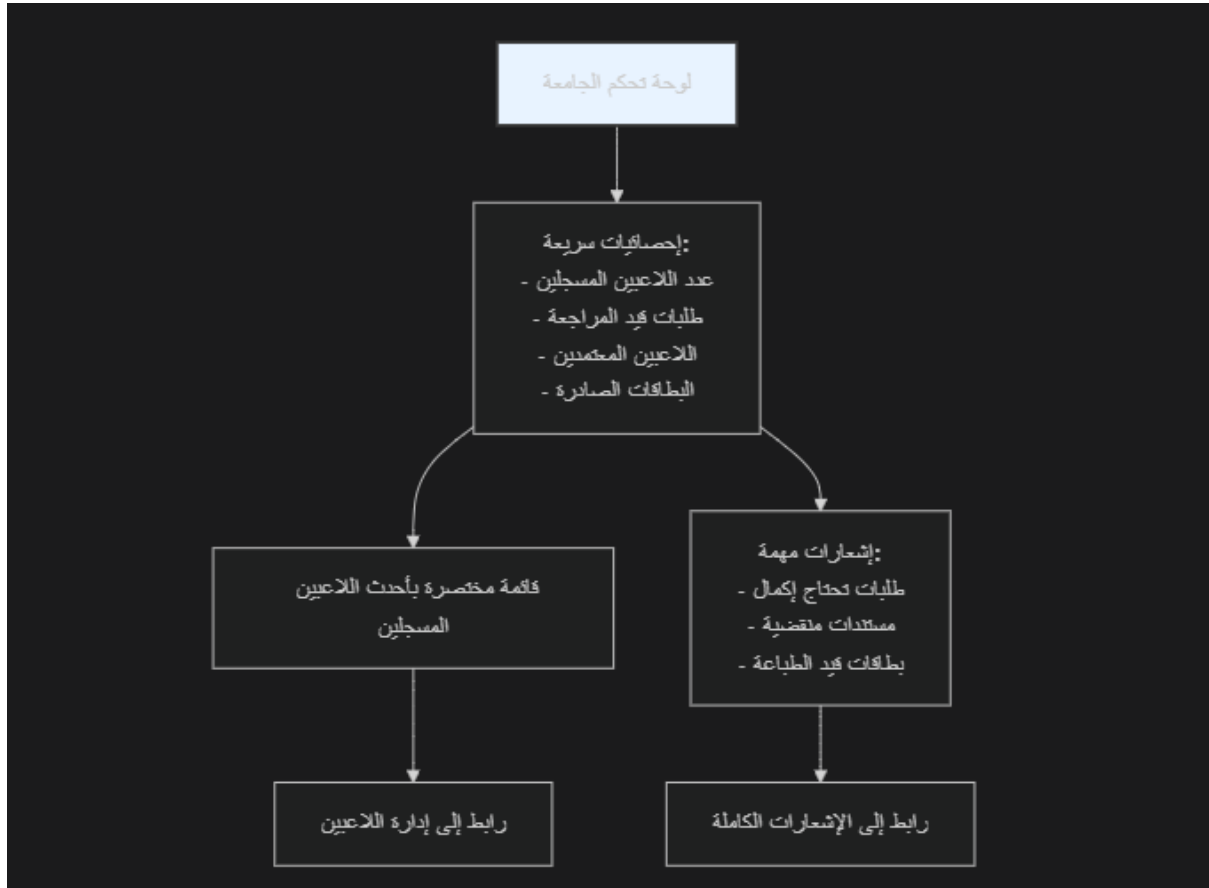
### 1.3 شاشة إعادة تعيين كلمة المرور (Reset Password)

- حقول إدخال كلمة المرور الجديدة
- حقول تأكيد كلمة المرور الجديدة
- زر "تعيين كلمة المرور"

• إشعار بنج العملية

2. شاشات إدارة اللاعبين (لوحة تحكم الجامعة)

2.1 لوحة التحكم الرئيسية (Dashboard)



2.2 شاشة تسجيل لاعب جديد (New Player Registration)

2.3 شاشة إدارة اللاعبين (Players Management)

- شريط بحث (بالاسم، الرقم القومي، الرقم الجامعي)
- عوامل تصفية (حالة اللاعب، الكلية، الرياضة)
- جدول عرض اللاعبين مع أعمدة (الاسم، الكلية، الحالة، الإجراءات)
- أزرار إجراءات (عرض، تعديل، حذف)
- زر "إضافة لاعب جديد"
- pagination للتنقل بين الصفحات.

## 2.4 شاشة تفاصيل اللاعب (Player Details)

- قسم المعلومات الشخصية
- قسم المعلومات الأكاديمية
- قسم المستندات المرفوعة (مع معاينة)
- قسم حالة الطلب وتاريخ المراجعات
- قسم البطاقة (إذا كانت صادرة)
- أزرار الإجراءات (تعديل، إعادة تقديم، إلغاء)

## 3. شاشات المراجعة والاعتماد (لوحة المشرفين)

### 3.1 لوحة مراجعة الطلبات (Review Dashboard)



### 3.2 شاشة مراجعة مفصلة (Detailed Review)

- قسم كبير لمعاينة المستندات (بجودة عالية)
- أدوات تكبير وتصغير للمستندات
- قائمة بالبيانات المدخلة مع إمكانية التعديل

- منطقة لكتابة ملاحظات المراجعة
- أزرار (موافقة، رفض، طلب مستندات إضافية)
- سجل المراجعات السابقة (إذا existed)

### 3.3 شاشة إدارة البطاقات (Cards Management)

- بحث برقم البطاقة أو اسم اللاعب
- جدول البطاقات الصادرة مع (رقم البطاقة، الاسم، تاريخ الإصدار، تاريخ الانتهاء، الحالة)
- إمكانية تصدير البيانات (Excel, PDF)
- إحصاءات عن البطاقات (منتهية الصلاحية، قيد الطباعة، ملغاة)
- زر إصدار بطاقة جديدة
- زر إعادة إصدار بطاقة
- زر إلغاء بطاقة

### 4. شاشات اللاعبين

#### 4.1 الصفحة الشخصية للاعب (Player Profile)



## 4.2 شاشة متابعة الطلب (Application Status)

- شريط تقدم يوضح مراحل الطلب
- تفاصيل كل مرحلة (تاريخ البدء، التاريخ المتوقع، الحالة)
- قائمة بالمستندات المطلوبة والمقدمة
- إشعارات بأي إجراء مطلوب من اللاعب
- وسائل التواصل مع الدعم الفني

## 5. شاشات التقارير والإحصائيات

### 5.1 لوحة التقارير (Reports Dashboard)

- مخططات بيانية للإحصائيات (لاعبين حسب الجامعة، حسب الرياضة، حسب الحالة)
- تقارير قابلة للتخصيص (الفترة الزمنية، الجامعة، الرياضة)
- إمكانية تصدير التقارير (PDF, Excel, CSV)
- تقارير جاهزة (تقرير اللاعبين المسجلين، تقرير البطاقات الصادرة، تقرير الفرق)

### 5.2 شاشة إنشاء تقرير مخصص (Custom Report)

- خيارات التخصيص (نوع التقرير، الفترة الزمنية، المعايير)
- معاينة قبل الطباعة/التصدير
- خيارات التصدير (صيغة الملف، الحقول المطلوبة)
- حفظ قوالب التقارير للاستخدام المستقبلي

## 6. عناصر واجهة المستخدم المشتركة

### 6.1 شريط التنقل الرئيسي (Navigation Bar)

- شعار النظام واسم الجامعة
- قائمة التنقل الرئيسية
- إشعارات المستخدم
- صورة الملف الشخصي وقائمة المستخدم

### 6.2 عناصر التفاعل

- أزرار بألوان موحدة (أخضر للإجراءات الإيجابية، أحمر للإجراءات السلبية، أزرق للمحايدة)
- رسائل تنبيه وتأكيد قبل الإجراءات المهمة

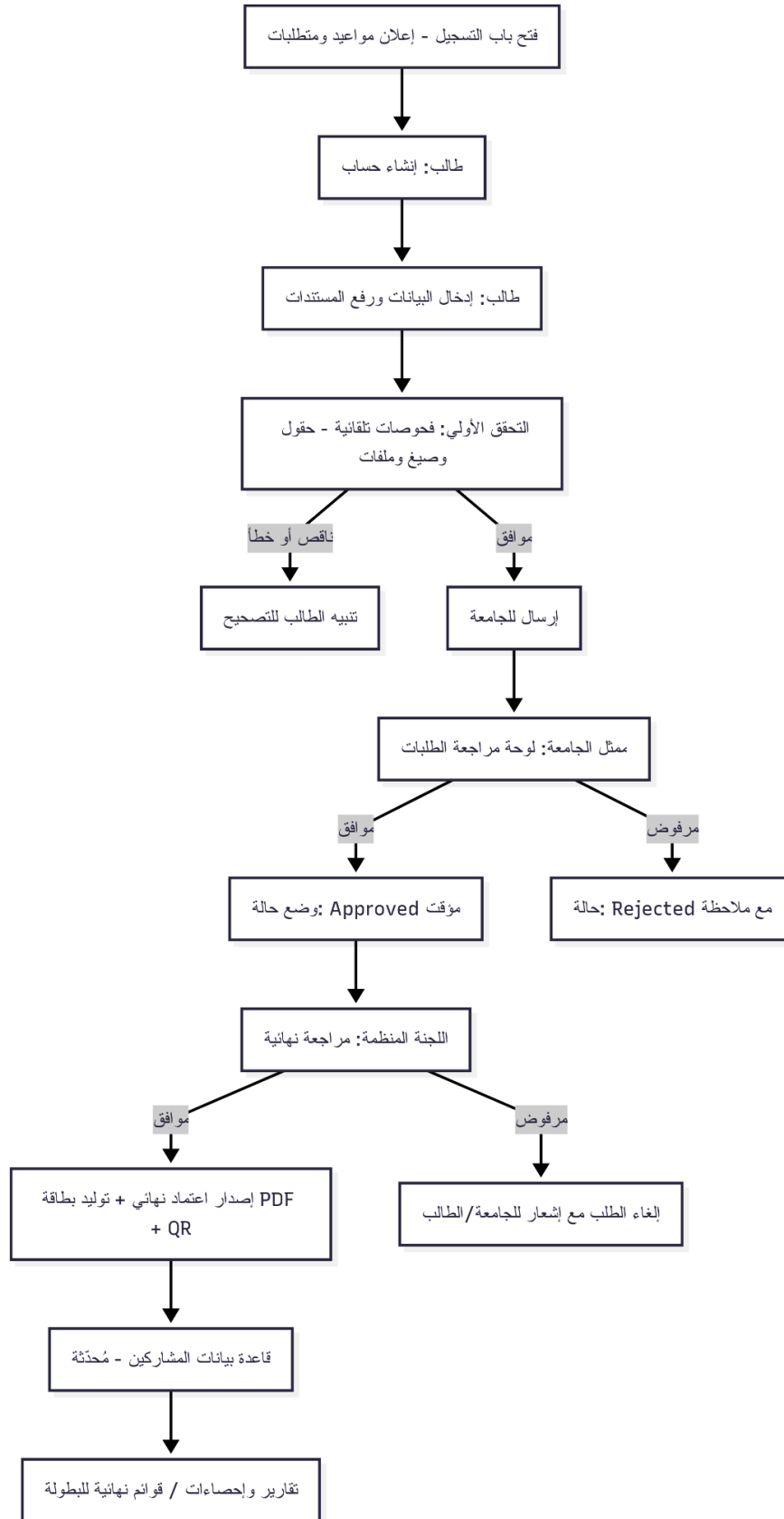
- تلميحات عند تمرير المؤشر (tooltips)
- مؤشرات تقدم للعمليات الطويلة

## 6.3 التصميم العام

- ألوان رئيسية: (#1a4f7e) وأبيض
- ألوان ثانوية: درجات من الأزرق الفاتح والرمادي
- خط عربي "Cairo": أو "Tajawal"
- خط إنجليزي "Roboto": أو "Open Sans"
- تصميم متجاوب يعمل على جميع أحجام الشاشات

# نظام التسجيل والاعتماد

## 1- تدفق تسجيل الطالب واعتماده Flowchart

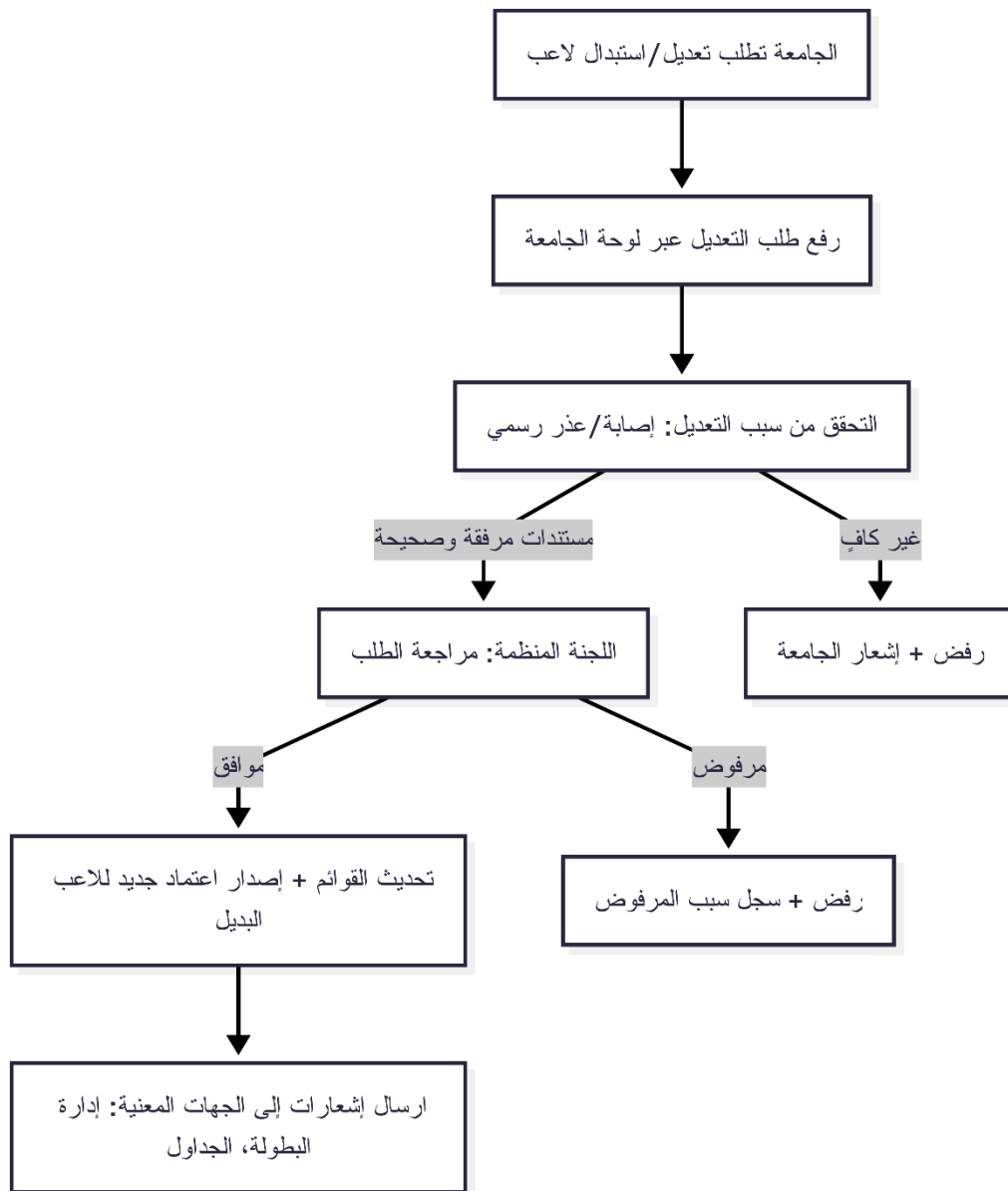


## ملاحظات سريعة على الـ Flowchart:

- خطوة "التحقق الأولي" تشمل فحوصات تلقائية مثل: تطابق رقم القيد/الهوية، صيغ ملفات PDF/صور، وجود جميع المستندات.
- حالات الرفض يجب أن تحتوي دائمًا على سبب مدون وميزة إعادة التقديم خلال نافذة زمنية محددة.
- إصدار الاعتماد يولد بطاقة تحتوي على QR/Barcode لسهولة الفحص عند البوابات المادية والسرية أو من خلال الهاتف المحمول.

## 2- تدفق استبدال/تعديل لاعب داخل الفترة المسموح بها Flowchart

## نظام التسجيل والاعتماد



**ملاحظة:** يجب تقييد عدد ونوع الاستبدالات حسب لوائح البطولة (يوضع في السياسات).

ER Diagram -3



## تجربة المستخدم UX :

**الهدف:** بناء نظام تسجيل واعتماد للبطولة يوفر تجربة سلسلة للطلاب والجامعة واللجنة المنظمة، ويطبق منطقاً شرطياً (conditional forms) لعرض الأسئلة الخاصة بكل رياضة فقط، مع تحقق ملفات ومراحل اعتماد متعددة ومخرجات تقارير وبطاقات اعتماد QR.

### المكونات الأساسية:

- واجهة مستخدم. (Web responsive + Mobile friendly)
- واجهة جامعات خاصة. (University Portal)
- لوحة إدارة للجنة المنظمة. (Admin Portal)
- API لخدمات التكامل و. Mobile
- قاعدة بيانات مركزية مع إدارة مستندات. (file storage)
- خدمة إصدار بطاقات PDF مع QR.

### تقسيم النماذج (Form Modularization)

نقترح تقسيم الاستمارة إلى 3 نماذج/مراحل رئيسية:

#### 1. General Form — بيانات أساسية (Mandatory)

- الاسم الرباعي، الرقم القومي، تاريخ الميلاد، النوع، الجامعة (قائمة)، الكلية، الفرقة، الهاتف، البريد الإلكتروني، حالة اللاعب (مشارك سابق/جديد)، موافقات (مشاركة معلومات، إقرار طبي).

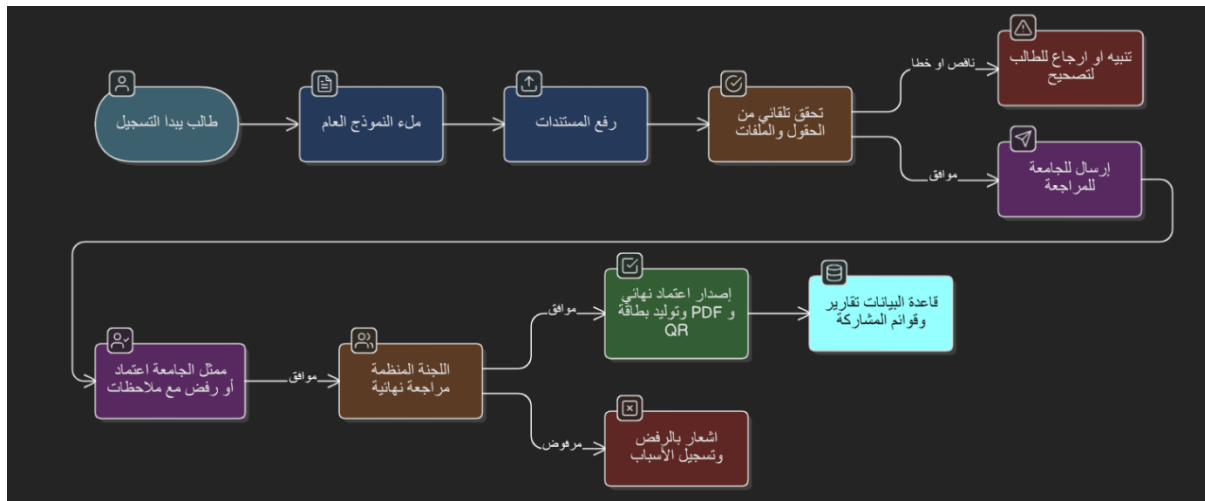
#### 2. Documents Upload — رفع المستندات (Mandatory)

- صورة شخصية. (jpg/png ≤ 2MB)
- شهادة قيد/بطاقة جامعية. (pdf/jpg ≤ 2MB)
- كشف طبي / كارنيه اتحاد / إقرار. (pdf ≤ 3MB)
- الحفظ مع حالة تحقق + (Pending / Verified / Rejected) سبب الرفض.

#### 3. Sport-Specific Form — حقول رياضية شرطية (Dynamic)

- تعرض بناءً على اختيار اللعبة: لكل لعبة نموذج فرعي يحتوي الحقول المطلوبة فقط (الوزن، النوع: فردي/زوجي/فرق، رقم الظهر، نوع المسابقة، المسافات/السباقات، فئة الإعاقة إن وجدت).
- يقيد النظام عدد المسابقات المسموح لكل لاعب (مثلاً: سباقان + تتابع واحد).

## تدفق العمل (High-level Workflow)



## نموذج قاعدة البيانات (SQL)

يمكن تعديل أنواع الحقول بحسب قواعد البيانات المختارة

```
CREATE TABLE users (
    id SERIAL PRIMARY KEY,
    full_name VARCHAR(255) NOT NULL,
    email VARCHAR(255) UNIQUE,
    password_hash VARCHAR(255),
    role VARCHAR(50) NOT NULL, /* student, university_rep, admin, referee */
    created_at TIMESTAMP DEFAULT now()
);
```

```
CREATE TABLE universities (
    id SERIAL PRIMARY KEY,
    name VARCHAR(255) NOT NULL,
    country VARCHAR(100),
    rep_user_id INT REFERENCES users(id)
);
```

```
CREATE TABLE students (  
  id SERIAL PRIMARY KEY,  
  user_id INT REFERENCES users(id),  
  university_id INT REFERENCES universities(id),  
  national_id VARCHAR(20) UNIQUE NOT NULL,  
  birth_date DATE,  
  gender VARCHAR(10),  
  faculty VARCHAR(255),  
  year_level VARCHAR(50),  
  phone VARCHAR(30),  
  status VARCHAR(50), /* pending, uni_approved, org_approved, rejected */  
  created_at TIMESTAMP DEFAULT now()  
);
```

```
CREATE TABLE sports (  
  id SERIAL PRIMARY KEY,  
  code VARCHAR(50),  
  name VARCHAR(255),  
  category VARCHAR(50) /* individual, team, para */  
);
```

```
CREATE TABLE participations (  
  id SERIAL PRIMARY KEY,  
  student_id INT REFERENCES students(id),  
  sport_id INT REFERENCES sports(id),
```

```
event_type VARCHAR(255), /* e.g. "100m", "Kata", "Taekwondo - Poomsae"
*/
weight_class VARCHAR(50),
is_para BOOLEAN DEFAULT false,
disability_type VARCHAR(255),
back_number VARCHAR(20),
qualifying_number VARCHAR(50),
status VARCHAR(50), /* pending, uni_approved, org_approved, rejected */
created_at TIMESTAMP DEFAULT now()
);
```

```
CREATE TABLE documents (
id SERIAL PRIMARY KEY,
student_id INT REFERENCES students(id),
doc_type VARCHAR(100),
file_path VARCHAR(500),
file_size INT,
mime_type VARCHAR(50),
verification_status VARCHAR(20) DEFAULT 'pending', /* pending, verified,
rejected */
note TEXT,
uploaded_at TIMESTAMP DEFAULT now()
);
```

```
CREATE TABLE accreditations (
id SERIAL PRIMARY KEY,
```

## نظام التسجيل والاعتماد

```
user_id INT REFERENCES users(id),
acc_type VARCHAR(50),
issue_date DATE,
expiry_date DATE,
qr_data TEXT
);
```

واجهات (UI) مقترحة — شاشات رئيسية ووظائفها

### 1. صفحة التسجيل (Wizard):

- Step 1: بيانات شخصية (الأهم؛ تحقق فوري من الرقم القومي والسن).
- Step 2: الجامعة والكلية autofill إذا كان البريد الجامعي معرّفًا.
- Step 3: اختيار الرياضة (searchable dropdown). عند اختيار الرياضة يظهر القسم الخاص بها ديناميكياً.
- Step 4: رفع المستندات (التحقق من الحجم/النوع قبل الرفع).
- Step 5: مراجعة + إرسال → عرض حالة التقديم مع رقم متابعة.

### 2. لوحة جامعة (University Dashboard)

- قائمة بالطلاب المرتبطة بالجامعة مع فلاتر (Pending / Awaiting Org / Approved / Rejected).
- شاشة تفصيلية لكل تسجيل مع إمكان تعديل الحقول الفنية (رقم الظهر، الرقم التأهيلي) واعتماد/رفض مع سبب.

### 3. لوحة اللجنة المنظمة (Admin Dashboard)

- نظرة عامة (counts, statuses).
- مراجعة نهائية وإصدار الاعتمادات وطباعتها.
- أدوات استيراد/تصدير / CSV طباعة قوائم مخصصة.

### 4. شاشة إصدار بطاقة الاعتماد

- توليد PDF مع: اسم، جامعة، الصورة، الرياضة/البطاقة، QR code مشفر يحتوي (acc\_id + user\_id + checksum).

## واجهة برمجة التطبيقات — (API)

### (RESTful JSON)

- **POST /api/auth/register** — تسجيل مستخدم جديد (student).
- **POST /api/auth/login** — تسجيل دخول (JWT).
- **GET /api/students/{id}** — جلب بيانات الطالب (roles-based).
- **POST /api/students/{id}/documents** — رفع مستند.
- **POST /api/students/{id}/participations** — إضافة مشاركة رياضية (يختلف حسب الرياضة payload).
- **GET /api/university/{id}/requests** — جلب طلبات الجامعة للمراجعة.
- **POST /api/university/{id}/requests/{req\_id}/approve** — اعتماد/رفض — من الجامعة مع ملاحظات.
- **POST /api/admin/accreditations/{student\_id}/generate** — توليد + اعتماد نهائي QR.
- **GET /api/reports/participants?sport=...&status=...** — تقارير قابلة للتصدير.

مثال JSON لإنشاء مشاركة (participation)

```
{
  "sport_id": 38,
  "event_type": "100m",
  "weight_class": null,
  "is_para": false,
  "back_number": "23",
  "qualifying_number": "Q-1245"
}
```

### قواعد التحقق (Validation Rules)

- الرقم القومي: تحقق طول وصيغة، استنتاج تاريخ الميلاد + التحقق من العمر ( $\geq 25$  حسب قواعد الاستثمار).

## نظام التسجيل والاعتماد

- توحيد أنواع الملفات المسموح بها. pdf, jpg, jpeg, png: حدود الحجم: MB 2 لكل ملف صور، MB 3 للـ pdf.
- لكل رياضة: تحقق عدد المسابقات المسموح والقيود (مثلاً: السباحة — كل لاعب  $\geq 2$  مسابقات + تتابع واحد).
- الحقول الإلزامية لا تُقبل فارغة، national\_id, full\_name, university\_id, sport\_id (إن شارك)، image.

### إدارة المستندات والتحقق

- عند الرفع: ملفات تُخزن مؤقتاً بعد فحص MIME type، ثم تُنقل إلى تخزين دائم (S3 أو storage server).
- كل مستند يحصل على حالة pending، ويعرض في لوحة الجامعة/اللجنة مع زر verify أو reject مع حقل سبب.
- الاحتفاظ بنسخ أصلية لمدة (سنة على الأقل) بغرض التدقيق.

### صلاحيات المستخدم (RBAC)

- Roles: student, university\_rep, org\_reviewer, admin, referee, volunteer\_coordinator.
- Matrix أمثلة:
- Student: إنشاء/تعديل بياناته حتى مرحلة uni\_approved. رفع مستندات. متابعة الحالة.
- University\_rep: مشاهدة/تعديل سجلات طلاب الجامعة، اعتماد/رفض.
- Org\_reviewer: مراجعة نهائية، إصدار اعتمادات، تعديل قوائم نهائية.
- Admin: إعداد النظام، إدارة الجامعات، تقارير شاملة.

### الأمن وحماية البيانات

- المصادقة JWT: مع فترة صلاحية قصيرة. refresh tokens +
- كلمات المرور: تخزين مشفر بـ bcrypt.
- الاتصال HTTPS: فقط. (TLS)
- حماية الملفات signed URLs: للوصول المؤقت إلى الملفات.
- سجلات (Audit logs): كل عملية اعتماد/رفض/تعديل تُسجل (user, timestamp, action, ip).

## نظام التسجيل والاعتماد

- سياسة احتفاظ بالبيانات متوافقة مع قواعد الخصوصية المحلية.

### الأداء والقدرة للتوسع

- استخدم CDN لملفات الصور و PDF.
- قاعدة بيانات قابلة للتوسيع (RDS / Managed DB).
- فصل services: Storage, API, Auth, Worker لإنشاء PDF و QR.
- دعم تحميل دفعات (bulk upload) من الجامعات، مع job queue لمعالجة الملفات.

### اختبارات (QA) وقبول النظام

- Unit tests لكل endpoint.
- Integration tests للرحلة التسجيل الكاملة.
- Load test: محاكاة ذروة تسجيل (مثلاً 5,000 تسجيل في يوم واحد).
- Acceptance criteria: قدرة تسجيل دون فقدان بيانات، توليد بطاقة QR صحيحة، عمولة مراجعة واعتماد تعمل من خلال واجهات الجامعة.

### تقارير ومخرجات نهائية

- تقارير per-university counts, per-sport counts, pending documents, rejected reasons.
- تصدير CSV/XLS لكل القوائم النهائية (للطباعات والجداول).
- واجهة طباعة بطاقات اعتماد (PDF batch).

### خطة تنفيذ مقترحة لميزات أساسية

1. Week 0: Requirement freeze + ERD + UI wireframes.
2. Week 1–3: Authentication + User management + Students basic form + Documents upload.
3. Week 4–6: University portal (review/approve) + sport-specific forms (top 10 sports).
4. Week 7–8: Admin portal + accreditation generation (PDF + QR).
5. Week 9: QA + Load testing + Pilot with 1–2 universities.
6. إطلاق واسع بعد التعديلات.

ملاحظة: يمكنك تعديل الجدول الزمني حسب فريق التطوير والموارد المتاحة.

## الواجهات الرئيسية المقترحة

### 1. واجهة الطالب – (Student Portal) نموذج متعدد الخطوات (Wizard)

- Step 1: البيانات الشخصية.
- Step 2: الجامعة + الكلية + الفرقة.
- Step 3: رفع المستندات.
- Step 4: اختيار الرياضة (حقول شرطية).
- Step 5: مراجعة + تأكيد.

## Student Registration Form (Wireframe)

### Step 1: Personal Info

- Name
- National ID
- Birth Date
- Gender

### Step 2: University Info

- University Dropdown
- Faculty
- Year

### Step 3: Documents Upload

[Photo] [ID Card] [Medical Cert]

### Step 4: Sport Selection

- Sport Dropdown
- Conditional Fields (Weight, Event)

### Step 5: Review & Submit

[Submit Button]

## 2. واجهة الجامعة: (University Dashboard)

- قائمة الطلاب مع فلتر حسب الحالة. (Pending / Approved / Rejected)
- شاشة مراجعة طالب → بيانات كاملة + مستندات مرفوعة + زر اعتماد/رفض + تعليق.

### University Dashboard (Wireframe)

#### Header: University Dashboard

[Filter: Pending | Approved | Rejected]

#### Table: Students List

Columns: Name | Sport | Status | Actions

[Approve] [Reject] [View Details]

## 3. واجهة اللجنة المنظمة: (Admin Dashboard)

- لوحة إحصاءات. (Dashboard)
- قائمة الطلاب حسب الجامعة/اللجنة/الحالة.
- شاشة إصدار الاعتماد. (PDF + QR)

### Admin Dashboard (Wireframe)

#### Header: Admin Dashboard

Stats Widgets: Total Students | Approved | Pending

Filters: University, Sport, Status

Student List with bulk approve/reject

Button: Generate Accreditation (PDF + QR)

## 4. بطاقة الاعتماد: (Accreditation Card)

○ تحتوي: صورة شخصية، اسم، جامعة، رياضة، QR.

## Accreditation Card (Wireframe)

|       |                               |
|-------|-------------------------------|
| Photo | Name: [Student Name]          |
|       | University: [University Name] |
|       | Sport: [Sport/Event]          |
|       | Accreditation: Athlete        |
| QR    |                               |

## نسخة مُبسّطة للنظام (HTML + React form)

```
import React, { useState } from "react";

// StudentRegistrationForm.jsx
// Single-file React component (tailwind classes used).
// - Multi-step (wizard) form
// - Basic client-side validation
// - Conditional sport-specific fields
// - Mock submit handlers (replace with real API calls)

export default function StudentRegistrationForm() {
  const [step, setStep] = useState(1);
  const [loading, setLoading] = useState(false);
  const [errors, setErrors] = useState({});

  const initial = {
    fullName: "",
    nationalId: "",
    birthDate: "",
    gender: "",
    university: "",
    faculty: "",
    yearLevel: "",
    phone: "",
    email: "",
    playerStatus: "new",
  }
```

```

acceptDataSharing: false,
medicalDeclaration: false,
photo: null,
idCard: null,
medicalDoc: null,
sportId: "",
sportEvent: "",
weightClass: "",
backNumber: "",
isPara: false,
disabilityType: "",
};

```

```

const [form, setForm] = useState(initial);

```

```

const sports = [
  { id: "", name: "-- اختر الرياضة --" },
  { id: "taekwondo", name: "Taekwondo" },
  { id: "swimming", name: "Swimming" },
  { id: "athletics", name: "Athletics" },
  { id: "weightlifting", name: "Weightlifting" },
  { id: "tabletennis", name: "Table Tennis" },
];

```

```

function handleChange(e) {
  const { name, value, type, checked, files } = e.target;

```

```

if (type === "checkbox") return setForm({ ...form, [name]: checked });
if (type === "file") return setForm({ ...form, [name]: files[0] });
setForm({ ...form, [name]: value });
}

function validateStep(currentStep = step) {
  const err = {};
  if (currentStep === 1) {
    if (!form.fullName) err.fullName = "الاسم مطلوب";
    if (!form.nationalId) err.nationalId = "الرقم القومي مطلوب";
    if (!form.birthDate) err.birthDate = "تاريخ الميلاد مطلوب";
    if (!form.university) err.university = "اختر الجامعة";
  }
  if (currentStep === 2) {
    if (!form.photo) err.photo = "أضف صورة شخصية";
    if (!form.idCard) err.idCard = "أضف صورة البطاقة/شهادة القيد";
  }
  if (currentStep === 3) {
    if (!form.sportId) err.sportId = "اختر الرياضة";
    // Example rule: swimming can have event selection
    if (form.sportId === "swimming" && !form.sportEvent) err.sportEvent = "اختر السباق";
  }
  setErrors(err);
  return Object.keys(err).length === 0;
}

```

```
async function handleNext() {  
  if (!validateStep()) return;  
  setStep((s) => Math.min(5, s + 1));  
}
```

```
function handlePrev() {  
  setStep((s) => Math.max(1, s - 1));  
}
```

```
async function handleSubmit(e) {  
  e.preventDefault();  
  if (!validateStep(4)) return;  
  setLoading(true);
```

```
// Build FormData for files  
const fd = new FormData();  
Object.keys(form).forEach((k) => {  
  const val = form[k];  
  if (val === null || val === "") return;  
  fd.append(k, val);  
});
```

```
// Mock submit: replace with real POST to API e.g. /api/auth/register  
await new Promise((r) => setTimeout(r, 900));
```

```

setLoading(false);

alert("تم إرسال الاستمارة بنجاح (تجريبي). سيتم تحويلك إلى صفحة المتابعة");

setForm(initial);

setStep(1);
}

// Small helper components

const Field = ({ label, children, error }) => (
  <div className="mb-3">
    <label className="block text-sm font-medium text-gray-700">{label}</label>
    {children}
    {error && <p className="text-red-600 text-xs mt-1">{error}</p>}
  </div>
);

return (
  <div className="max-w-3xl mx-auto p-6">
    <div className="bg-white shadow rounded-lg p-6">
      <h2 className="text-lg font-bold mb-4">نموذج تسجيل الطالب - الدورة الرياضية</h2>

      <div className="mb-4 flex gap-2 text-xs">
        {[1, 2, 3, 4, 5].map((i) => (
          <div
            key={i}

```

```
className={`flex-1 text-center py-2 rounded ${step === i ? 'bg-blue-600 text-white' : 'bg-gray-100'}`}
```

```
>
```

```
{i} خطوة
```

```
</div>
```

```
)))
```

```
</div>
```

```
<form onSubmit={handleSubmit}>
```

```
{/* Step 1: Personal */}
```

```
{step === 1 && (
```

```
<div>
```

```
<Field label="الاسم رباعي" error={errors.fullName}>
```

```
<input name="fullName" value={form.fullName}  
onChange={handleChange}
```

```
className="mt-1 block w-full border rounded p-2" />
```

```
</Field>
```

```
<div className="grid grid-cols-2 gap-4">
```

```
<Field label="الرقم القومي" error={errors.nationalId}>
```

```
<input name="nationalId" value={form.nationalId}  
onChange={handleChange}
```

```
className="mt-1 block w-full border rounded p-2" />
```

```
</Field>
```

```
<Field label="تاريخ الميلاد" error={errors.birthDate}>
```

```
<input type="date" name="birthDate" value={form.birthDate}
onChange={handleChange}
```

```
className="mt-1 block w-full border rounded p-2" />
```

```
</Field>
```

```
</div>
```

```
<div className="grid grid-cols-2 gap-4">
```

```
<Field label="النوع">
```

```
<select name="gender" value={form.gender}
onChange={handleChange}
```

```
className="mt-1 block w-full border rounded p-2">
```

```
<option value="">-- اختر --</option>
```

```
<option value="male">طالب</option>
```

```
<option value="female">طالبة</option>
```

```
</select>
```

```
</Field>
```

```
<Field label="حالة اللاعب">
```

```
<select name="playerStatus" value={form.playerStatus}
onChange={handleChange}
```

```
className="mt-1 block w-full border rounded p-2">
```

```
<option value="new">لاعب جديد</option>
```

```
<option value="returning">مقيد في الدورة السابقة</option>
```

```
</select>
```

```
</Field>
```

```
</div>
```

```

<Field label="البريد الإلكتروني">

  <input name="email" value={form.email}
onChange={handleChange}

  className="mt-1 block w-full border rounded p-2" />

</Field>

</div>

})

{/* Step 2: University */}
{step === 2 && (
  <div>

    <Field label="الجامعة" error={errors.university}>

      <select name="university" value={form.university}
onChange={handleChange}

        className="mt-1 block w-full border rounded p-2">

          <option value="">-- اختر الجامعة --</option>

          <option value="cairo">جامعة القاهرة</option>

          <option value="alex">جامعة الاسكندرية</option>

          <option value="alz">جامعة الزقازيق</option>

        </select>

      </Field>

      <div className="grid grid-cols-2 gap-4">

        <Field label="الكلية">

          <input name="faculty" value={form.faculty}
onChange={handleChange}

```

```

{ /* Step 3: Documents */ }

{ step === 3 && (
  <div>
    <Field label="الصورة الشخصية" error={errors.photo}>
      <input type="file" name="photo" accept="image/*"
        onChange={handleChange}
        className="mt-1 block w-full" />
    </Field>
  </div>
) }

```

```

<Field label="بطاقة/شهادة قيد" error={errors.idCard}>
  <input type="file" name="idCard" accept="image/*,.pdf"
onChange={handleChange}
  className="mt-1 block w-full" />
</Field>

<Field label="المستند الطبي (إن وجد)">
  <input type="file" name="medicalDoc" accept="image/*,.pdf"
onChange={handleChange}
  className="mt-1 block w-full" />
</Field>

<div className="flex gap-3 items-center mt-4">
  <input type="checkbox" name="acceptDataSharing"
checked={form.acceptDataSharing} onChange={handleChange} />
  <label className="text-sm">أوافق على مشاركة المعلومات</label>
</div>

<div className="flex gap-3 items-center mt-2">
  <input type="checkbox" name="medicalDeclaration"
checked={form.medicalDeclaration} onChange={handleChange} />
  <label className="text-sm">أقر بأنني لائق طبياً</label>
</div>
</div>
  )}

```

```

{ /* Step 4: Sport selection (dynamic fields) */}

{step === 4 && (

  <div>

    <Field label="اختر الرياضة" error={errors.sportId}>

      <select name="sportId" value={form.sportId}
onChange={handleChange}

        className="mt-1 block w-full border rounded p-2">

          {sports.map((s) => (

            <option key={s.id} value={s.id}>{s.name}</option>

          ))}

        </select>

      </Field>

    { /* Conditional: swimming */}

    {form.sportId === "swimming" && (

      <div>

        <Field label="اختيار السباق" error={errors.sportEvent}>

          <select name="sportEvent" value={form.sportEvent}
onChange={handleChange}

            className="mt-1 block w-full border rounded p-2">

              <option value="">-- اختر سباق --</option>

              <option value="50free">50 م حرة</option>

              <option value="100free">100 م حرة</option>

            </select>

          </Field>

        
```

```

<Field label="رقم الظهر">

    <input name="backNumber" value={form.backNumber}
onChange={handleChange}

        className="mt-1 block w-full border rounded p-2" />

</Field>

</div>

)}

{/* Conditional: taekwondo */}
{form.sportId === "taekwondo" && (
    <div>

        <Field label="المسابقة (كروجي/بومزا)">

            <select name="sportEvent" value={form.sportEvent}
onChange={handleChange}

                className="mt-1 block w-full border rounded p-2">

                    <option value="">-- اختر --</option>

                    <option value="poomsae">Poomsae</option>

                    <option value="kyorugi">Kyorugi</option>

                </select>

            </Field>

            <Field label="فئة الوزن">

                <input name="weightClass" value={form.weightClass}
onChange={handleChange}

                    className="mt-1 block w-full border rounded p-2" />

            </Field>

        </div>
    )}

```

```
}}
```

```
{/* Common flag for para-sports */}
```

```
<div className="flex gap-3 items-center mt-3">
```

```
  <input type="checkbox" name="isPara" checked={form.isPara}  
  onChange={handleChange} />
```

```
  <label className="text-sm">ذو احتياجات خاصة</label>
```

```
</div>
```

```
{form.isPara && (
```

```
  <Field label="نوع الإعاقة">
```

```
    <input name="disabilityType" value={form.disabilityType}  
    onChange={handleChange}
```

```
    className="mt-1 block w-full border rounded p-2" />
```

```
  </Field>
```

```
})
```

```
</div>
```

```
})
```

```
{/* Step 5: Review */}
```

```
{step === 5 && (
```

```
  <div>
```

```
    <h3 className="font-semibold mb-2">مراجعة البيانات</h3>
```

```
    <div className="bg-gray-50 p-4 rounded">
```

```
      <p><strong>الاسم:</strong> {form.fullName}</p>
```

```
      <p><strong>الرقم القومي:</strong> {form.nationalId}</p>
```

```

<p><strong>الجامعة:</strong> {form.university}</p>

<p><strong>الرياضة:</strong> {form.sportId || '-'}
{form.sportEvent ? ` (${form.sportEvent}) ` : ''}</p>

<p><strong>مستندات مرفوعة</strong> {form.photo ? 'صورة، ': ''}
"{{form.idCard ? 'بطاقة، ': ''}}{form.medicalDoc ? 'طبي': ''}"</p>

</div>

</div>

}}

<div className="mt-6 flex justify-between">

<div>

{step > 1 && (

  <button type="button" onClick={handlePrev} className="px-4
py-2 bg-gray-200 rounded">السابق</button>

)}

</div>

<div className="flex gap-3">

{step < 5 && (

  <button type="button" onClick={handleNext} className="px-4
py-2 bg-blue-600 text-white rounded">التالي</button>

)}

{step === 5 && (

  <button type="submit" disabled={loading} className="px-4 py-
2 bg-green-600 text-white rounded">

    {loading ? 'إرسال الاستمارة' : 'إرسال...'}

  </button>

)}


```

```
</button>

})

</div>

</div>

</form>

</div>

</div>

);
```

{

## إعادة طرح المقترح ليتناسب مع دورة الألعاب الجامعية الإفريقية – مصر 2026

### ◆ النسخة المعدلة للمقترح

#### الهدف من النظام

- أتمتة تسجيل واعتماد جميع المشاركين في دورة الألعاب الجامعية الإفريقية 2026 (طلاب – مدربين – إداريون – حكام – متطوعون – وفود الدول).
- توحيد الإجراءات على مستوى إفريقي مع احترام الفوارق المحلية لكل جامعة ودولة.
- توفير نظام مركزي يدعم تعدد اللغات (عربية/إنجليزية/فرنسية) ليتناسب مع الدول الإفريقية.
- تسهيل إصدار بطاقات الاعتماد الدولية (Accreditation Cards) مرتبطة ب QR/Barcode للفحص السريع.

#### الفئات المستفيدة

- الطلاب الرياضيون: تسجيل بيانات + مستندات (أكاديمية/طبية).
- المدربون والإداريون: اعتماد الفرق وتمثيل الجامعات/الدول.
- الجامعات الوطنية: مراجعة الطلبات الخاصة بطلابها واعتمادها أولاً.
- الاتحادات/اللجنة المنظمة: اعتماد نهائي + إصدار البطاقات.
- الحكام والمتطوعون: تقديم طلبات واعتمادهم من اللجنة.
- الوفود الإفريقية: الحصول على لوحات متابعة مركزية خاصة بدولتهم.

#### دورة العمل (Workflow)

1. فتح باب التسجيل: إعلان رسمي متعدد اللغات يوضح المواعيد والمتطلبات.
2. تسجيل الطلاب: إدخال البيانات + رفع الوثائق.
3. مراجعة الجامعات/الاتحاد المحلي: (NUSF) اعتماد أو رفض.
4. مراجعة اللجنة المنظمة – LOC مصر 2026: (التحقق النهائي).
5. إصدار بطاقات الاعتماد. (PDF + QR)
6. إدارة الاستبدالات: خلال فترة زمنية محددة حسب لوائح البطولة.
7. إصدار القوائم النهائية حسب الرياضة والدولة.

## نظام التسجيل والاعتماد

### التكليف الإفريقي (مقارنة بالنسخة المصرية)

- إضافة وحدة إدارة الوفود الدولية:
  - كل دولة لها مسؤول رسمي. (Head of Delegation)
  - صلاحيات خاصة بمراجعة واعتماد بيانات وفد دولته.
- دعم متطلبات السفر والتأشيرات:
  - رفع جواز السفر + صورة شخصية دولية.
  - تقارير لتسهيل استخراج التأشيرات الجماعية.
  - إضافة بيانات الوصول والمغادرة بالتواريخ والوقت.
- وحدة التكامل مع النقل والإقامة:
  - ربط المشاركين ببيانات الفندق/النقل.
  - إصدار بطاقة الاعتماد تستخدم كـ “هوية دخول + بطاقة إقامة”.
- إضافة قواعد تحقق أعمار/أهلية:
  - التحقق من الأعمار ( $\geq 25$  سنة).
  - السماح بالاستثناءات حسب لوائح FASU/FISU.

### المخرجات الرئيسية

- قاعدة بيانات مركزية لكل الدول المشاركة.
- استخدام اختصارات الدول طبقا للجنة الأولمبية الدولية.
- بطاقات اعتماد دولية. (Athlete, Coach, Referee, Volunteer, Official)
- تقارير باللغات الثلاثة عن:
  - المشاركين حسب الدولة/الجامعة/اللعبة.
  - توزيع الأعمار/الجنسيات/الجنس.
  - حالات الطلبات. (Pending, Approved, Rejected)
- تصدير التقارير بصيغ. (PDF, Excel, CSV)

## البنية التقنية

- Web + Mobile responsive.
- لغات متعددة. (i18n)
- تكامل API مع أنظمة الجامعات والاتحادات الوطنية.
- Cloud Hosting مع دعم 10,000+ مستخدم متزامن.

## خطة التنفيذ (مقترحة لمصر 2026)

- الربع الأخير 2025: التحليل + تصميم قاعدة البيانات + النماذج + بناء النظام.
- يناير – فبراير 2026: اختبار تجريبي في المحاكاة.
- مارس – مايو 2026: تعميم التسجيل على جميع الدول + تدريب مسؤولي الجامعات.
- يونيو 2026: اعتماد نهائي + إصدار البطاقات.
- يوليو 2026 (أثناء البطولة): دعم فني مباشر + مركز متابعة لحظي.