**وزارة التعليم العالي**  
**معهد العبور العالي للإدارة والحاسبات ونظم المعلومات**

**منصة إلكترونية للسياحة في مصر**

**مشروع تخرج استيفاءً لمتطلبات الحصول على درجة البكالوريوس**

**إعداد:**  
[

**إشراف:**  
د/ [اسم المشرف]  
م/ [اسم المشرف المساعد]

**نظم المعلومات الإدارية**  
2023/2024

**المقدمة**

في عصر التحول الرقمي، أصبحت التقنيات الحديثة تلعب دورًا مهمًا في قطاع السياحة، حيث يحتاج السياح إلى معلومات دقيقة وشاملة عن الأماكن السياحية في مصر، وأفضل الأنشطة التي يمكن ممارستها، والمطاعم القريبة، وأسعار التذاكر. يهدف هذا المشروع إلى إنشاء منصة إلكترونية تساعد السياح في التخطيط لرحلاتهم بسهولة من خلال توفير بيانات دقيقة وحديثة عن المواقع السياحية في مصر. كما أن المنصة ستكون **متعددة اللغات** لدعم جميع الجنسيات، بما في ذلك المصريين والسياح الأجانب، لضمان تجربة مستخدم سلسة لجميع الزوار. إضافة إلى ذلك، ستشمل المنصة قسمًا خاصًا **للمرشدين السياحيين المعتمدين من وزارة الآثار**، حيث يمكن للسياح اختيار مرشد يتحدث لغتهم، مما يسهل تجربتهم السياحية.

**فكرة المشروع (المشكلة)**

تواجه السياحة في مصر العديد من التحديات فيما يخص المعلومات المتاحة للسياح، مثل:

* عدم توفر دليل سياحي رقمي شامل وسهل الاستخدام.
* صعوبة الوصول إلى معلومات محدثة حول الأماكن السياحية.
* غياب دليل متكامل عن الأنشطة والمطاعم القريبة من المواقع الأثرية والسياحية.
* عدم وجود تصنيفات موحدة لأسعار التذاكر والخدمات.
* الحاجة إلى دعم **تعدد اللغات** لتسهيل التعامل مع السياح من مختلف الجنسيات.
* نقص آلية تتيح للمستخدمين مشاركة **آرائهم ونصائحهم** حول الأماكن السياحية.
* غياب قسم **لربط السياح بالمرشدين السياحيين المعتمدين**، مما يحد من تجربة السياحة التفاعلية.
* عدم وجود فرصة لطلاب اللغات لممارسة مهاراتهم اللغوية مع السياح والمرشدين.

**أهداف المشروع**

* توفير منصة إلكترونية سهلة الاستخدام تساعد السياح في العثور على المعلومات بسهولة.
* تقديم تفاصيل دقيقة حول الأماكن السياحية، بما في ذلك الصور، الأوصاف، التقييمات.
* توفير دليل للمطاعم القريبة من المواقع السياحية مع تقييمات الزوار.
* عرض الأنشطة الترفيهية المتاحة في كل موقع وأسعارها.
* دعم **اللغات المختلفة** لجذب السياح من مختلف أنحاء العالم.
* إضافة **قسم خاص بآراء المستخدمين ونصائح الزوار السابقين** لتحسين تجربة المستخدمين الجدد.
* **ربط الفنادق والأماكن السياحية بالمنصة** مع توفير أرقام التواصل المباشرة.
* **إضافة قسم خاص للمرشدين السياحيين المعتمدين**، حيث يمكن للسياح التواصل مع مرشد يتحدث لغتهم بسهولة.
* **إنشاء فرصة لطلاب اللغات لممارسة لغاتهم مع المرشدين السياحيين والسياح لتعزيز مهاراتهم العملية.**

**تحليل النظام**

**طرق التحليل**

تم استخدام عدة طرق لتحليل النظام لضمان تحقيق الأهداف المرجوة، مثل:

1. **تحليل النظام الحالي:** دراسة المنصات الحالية ومعرفة نقاط الضعف فيها.
2. **تحليل تدفق البيانات (DFD):** لتوضيح كيفية انتقال البيانات بين مكونات النظام.
3. **مخطط علاقات الكيانات (ERD):** لتحديد الكيانات الرئيسية في النظام والعلاقات بينها.
4. **تحليل أمنية النظام:** لضمان حماية البيانات وخصوصية المستخدمين.

**تصميم المنصة**

**أدوات التصميم**

* **Figma** لتصميم واجهات المستخدم.
* **Adobe Photoshop** لتعديل الصور والعناصر البصرية.
* **EdrawMax** لرسم المخططات.

**الواجهات الرئيسية للمنصة**

1. **الصفحة الرئيسية:** تحتوي على أقسام رئيسية مثل الوجهات السياحية، أفضل الأماكن، أحدث العروض.
2. **صفحة البحث:** تتيح البحث عن الأماكن السياحية حسب الفئة أو الموقع الجغرافي.
3. **صفحة التفاصيل:** تعرض معلومات شاملة عن كل موقع سياحي، بما في ذلك الصور، التقييمات، الأنشطة، والمطاعم القريبة.
4. **صفحة الحجوزات:** تمكن المستخدم من حجز تذاكر الدخول أو الجولات المصحوبة بمرشدين.
5. **صفحة الآراء والتقييمات:** حيث يمكن للمستخدمين مشاركة آرائهم ونصائحهم حول الأماكن التي زاروها، مما يساعد السياح الآخرين في اتخاذ قرارات أفضل.
6. **قسم المرشدين السياحيين:** يتيح للسياح البحث عن مرشد سياحي بناءً على اللغة والموقع، مع إمكانية التواصل المباشر عبر التطبيق.
7. **قسم دعم الطلاب:** حيث يمكن لطلاب اللغات التسجيل لممارسة لغاتهم مع المرشدين والسياح، مما يخلق بيئة تعليمية تفاعلية مفيدة.

**التنفيذ والبرمجة**

**أدوات البرمجة المستخدمة**

* **HTML, CSS, JavaScript** لإنشاء الواجهة الأمامية.
* **PHP و Firebase** لإدارة قاعدة البيانات وتخزين المعلومات.
* **Google Maps API** لتحديد مواقع الأماكن السياحية وعرضها على الخريطة.
* **تقنيات تعدد اللغات** لتقديم واجهة متعددة اللغات تلائم جميع المستخدمين.

**مزايا المنصة**

* **سهولة الاستخدام:** واجهة مبسطة للمستخدمين.
* **بيانات محدثة:** يتم تحديث البيانات تلقائيًا.
* **متعددة اللغات:** دعم أكثر من لغة لجذب السياح الدوليين.
* **تجربة تفاعلية:** عرض تقييمات المستخدمين وتوصياتهم.
* **نظام مراجعات وتقييمات:** حيث يمكن للمستخدمين مشاركة تجاربهم ونصائحهم عن الأماكن السياحية.
* **دليل شامل للمرشدين السياحيين المعتمدين مع بيانات التواصل.**
* **فرصة لطلاب اللغات للتفاعل مع السياح وتطوير مهاراتهم اللغوية.**

**ادوات جمع البيانات**

* الكتب العلمية في مجال البحث.
* المقابلات الشخصية.
* الانترنت

**الحدود الزمانية والمكانية للبحث:**

بالنسبة للحدود **المكانية** للمشروع فان جميع الدول تستطيع الدخول علي الموقع والاستعلام والتفاعل معه والحدود الزمانيه هو مشروع تخرج لدفعه نظم ومعلومات لمعاهد العبور دفعه 2025

1. **هيكل البحث:**

* يحتوي البحث على ستة فصول:
* **الفصل الأول** : يحتوي على الإطار العام .
* **الفصل الثاني** : يحتوي على الإطار النظري .
* **الفصل الثالث** : التحليل ويتم تحليل كامل للنظام
* **الفصل الرابع** : التصميم ويتم فيه تصميم كامل للنظام .
* **الفصل الخامس** : البرمجة والتنفيذ وهنا يتم التنفيذ الفعلي للنظام .
* **الفصل السادس** : الخاتمة والتوصيات والنتائج .
* الملاحق والمراجع.

## ما هو الويب؟

الويب هو نظام يعمل عبر الإنترنت يتيح للمستخدمين تصفح المحتوى والوصول إلى المعلومات باستخدام متصفحات الويب مثل Google Chrome، Mozilla Firefox، Microsoft Edge وغيرها. يتم تصميم المواقع الإلكترونية باستخدام لغات مثل HTML، CSS، JavaScript، ويتم استضافتها على خوادم ليتمكن المستخدمون من الوصول إليها عبر الإنترنت من أي جهاز متصل بالشبكة.

## مميزات الويب

✅ متاح للجميع: يمكن لأي شخص يمتلك اتصالًا بالإنترنت الوصول إلى المواقع الإلكترونية دون الحاجة إلى تحميل أو تثبيت أي تطبيقات.

✅ يدعم جميع الأنظمة: يمكن فتح مواقع الويب على مختلف الأجهزة مثل الهواتف، الأجهزة اللوحية، أجهزة الكمبيوتر دون الحاجة إلى توافق خاص.

✅ سهولة التحديث: يمكن تحديث محتوى المواقع الإلكترونية بسهولة ودون الحاجة إلى إعادة تحميل التطبيق مثلما يحدث في تطبيقات الهواتف الذكية.

✅ متوافق مع محركات البحث: يمكن تحسين المواقع الإلكترونية ليتم العثور عليها بسهولة عبر Google و Bing من خلال تقنيات SEO.

✅ تكلفة أقل: تطوير موقع إلكتروني أقل تكلفة مقارنة بتطوير تطبيق أندرويد أو iOS.

## أسباب نجاح الويب

📌 التجاوب (Responsive Design): تصميم المواقع ليعمل على مختلف الأجهزة بأحجام شاشة مختلفة.

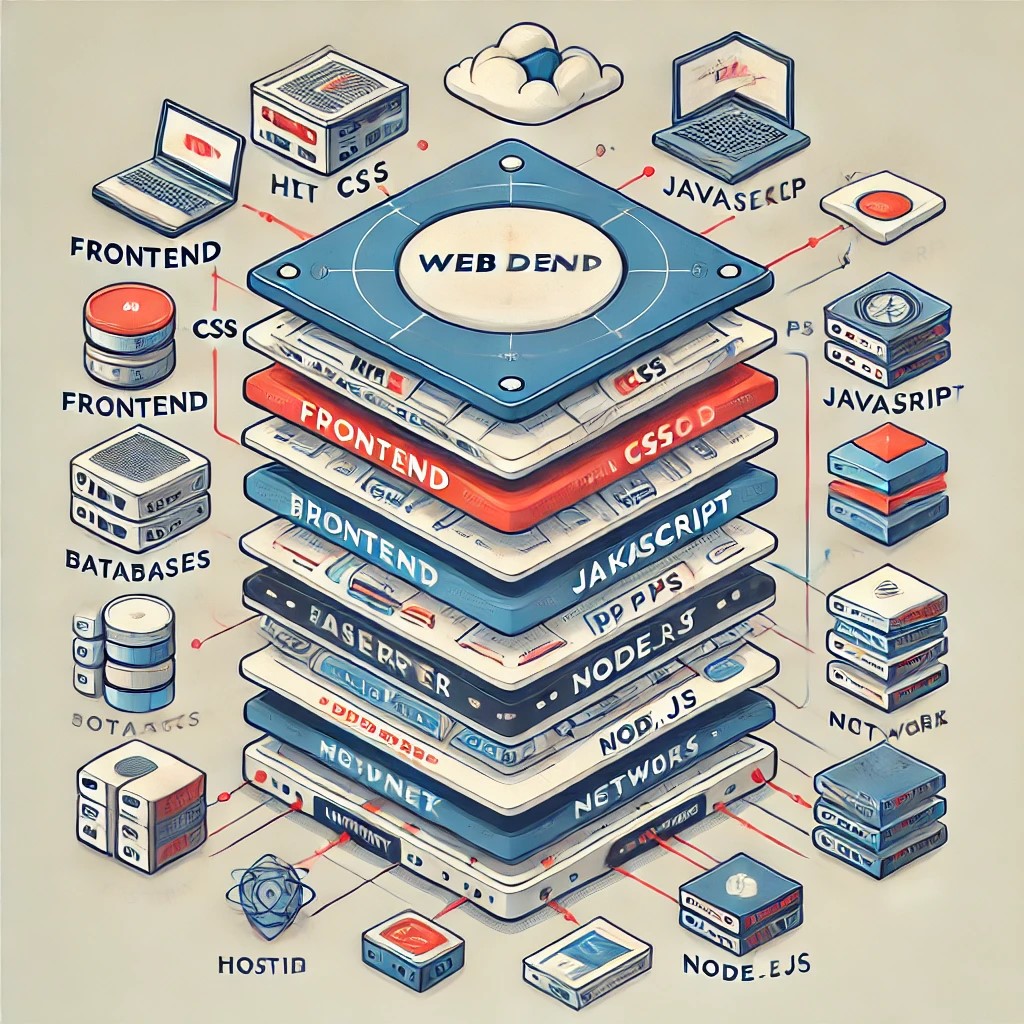
📌 سرعة التحميل: المواقع السريعة والجيدة الأداء تجذب الزوار وتزيد من معدلات التفاعل.

📌 أمان وحماية البيانات: المواقع التي توفر شهادات SSL وتشفير البيانات تعطي ثقة أكبر للمستخدمين.

📌 تحسين محركات البحث (SEO): استخدام تقنيات SEO يساعد في زيادة الزيارات العضوية من محركات البحث.

📌 التجربة السهلة (User Experience - UX): التصميم البسيط وسهولة التنقل في الموقع يعزز من رضا المستخدمين.

**هيكل بناء الويب**



* هذه صورة توضيحية لهيكل بناء الويب، تُظهر الطبقات المختلفة مثل **الواجهة الأمامية (Frontend)، والخلفية (Backend)، والشبكة (Network)** مع توضيح تدفق البيانات بينها.

### ****أهم مكتبات الويب****

* هذه المكتبات مصممة خصيصًا لتطوير تطبيقات الويب، وهي عبارة عن **APIs** يستخدمها مطورو الويب لاستغلال موارد المتصفح والخادم، وتشمل:
* **Session Management إدارة الجلسات** تُستخدم لإدارة جلسات المستخدم، مثل تسجيل الدخول والخروج وتخزين بيانات الجلسة.
* **Web Storage تخزين الويب** يشمل **LocalStorage وSessionStorage** لتخزين البيانات في المتصفح دون الحاجة إلى قاعدة بيانات خارجية.
* **Fetch API / Axios**: تُستخدم لإرسال واستقبال البيانات من السيرفر عبر **HTTP requests**.
* **DOM Manipulation التعامل مع الـ DOM)**: تسمح بتحديث وتحريك عناصر واجهة المستخدم ديناميكيًا باستخدام JavaScript.
* **WebSockets**: تُستخدم لإنشاء اتصالات لحظية بين المتصفح والخادم، مفيدة للدردشة وتحديثات البيانات في الوقت الفعلي.
* **Geolocation API** تُستخدم للحصول على موقع المستخدم الجغرافي من خلال المتصفح.
* **Push Notification API** تُستخدم لإرسال إشعارات للمستخدمين عبر المتصفح حتى في حالة عدم فتح الموقع.
* **Service Workers**: تُستخدم لتوفير تجربة **Offline Mod** وتحسين أداء المواقع عبر التخزين المؤقت للبيانات.
* **Canvas API** تُستخدم لإنشاء رسومات وجرافيكس داخل المتصفح، مثل الألعاب وتطبيقات التصميم.
* **WebRTC API** تُستخدم لإنشاء اتصالات فيديو وصوت مباشرة بين المستخدمين بدون الحاجة إلى خوادم وسيطة.

#### **الفرق الأساسي بين مكتبات الأندرويد والويب:**

📌 **مكتبات الأندرويد** تعمل على مستوى نظام التشغيل مباشرة وتتحكم في موارد الجهاز مثل المكالمات، الإشعارات، والبيانات المخزنة.  
📌 **مكتبات الويب** تعمل على مستوى المتصفح، وتتحكم في الموارد المتاحة داخل بيئة الويب مثل التخزين المؤقت، التفاعل مع المستخدم، والاتصال بالسيرفر.

### ****مكونات نظام الويب من حيث النشاط، دورة الحياة، وشكل الجداول والنماذج****

#### **1. مكونات نظام الويب من حيث النشاط**

يتكون نظام الويب من عدة مكونات أساسية تتفاعل مع بعضها البعض لتقديم تجربة مستخدم متكاملة، وتشمل:

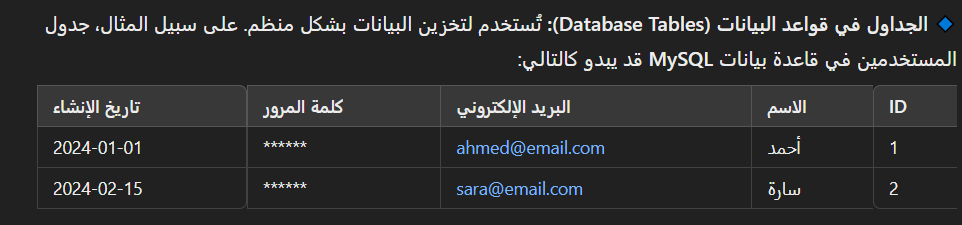
* **المتصفح (Browser):** هو الوسيط بين المستخدم والموقع الإلكتروني، حيث يعالج الأكواد مثل HTML, CSS, JavaScript ويعرضها للمستخدم.
* **الخادم (Server):** يستقبل الطلبات من المتصفح ويعيد إرسال الصفحات والبيانات المطلوبة، وغالبًا ما يكون Apache أو Nginx أو Node.js.
* **قاعدة البيانات (Database):** تخزن المعلومات مثل بيانات المستخدمين، المنتجات، والمحتوى، وتُدار باستخدام أنظمة مثل **MySQL, PostgreSQL, MongoDB**.
* **طبقة التطبيق (Application Layer):** تشمل الكود المسؤول عن معالجة البيانات والتفاعل بين الواجهة الأمامية والخلفية، مثل **PHP, Python, JavaScript (Node.js)**.
* **واجهة المستخدم (UI):** تتكون من الصفحات والتصاميم التي يراها المستخدم، وتُبنى باستخدام **HTML, CSS, JavaScript, React, Vue, Angular**.
* **واجهات برمجة التطبيقات (APIs):** تتيح التواصل بين مكونات النظام أو مع أنظمة خارجية مثل **RESTful APIs, GraphQL**.

#### **2. دورة حياة نظام الويب (Web Application Lifecycle)**

يمر نظام الويب بعدة مراحل لضمان تطويره، تشغيله، وصيانته، وهي:

1. **التخطيط (Planning):** تحديد الأهداف، الجمهور المستهدف، والمتطلبات الفنية.
2. **التصميم (Design):** إنشاء التصميم الأولي للواجهات وتجربة المستخدم (UX/UI).
3. **التطوير (Development):** بناء النظام باستخدام لغات البرمجة المناسبة وقواعد البيانات.
4. **الاختبار (Testing):** فحص الموقع بحثًا عن الأخطاء وتحسين الأداء والأمان.
5. **النشر (Deployment):** رفع الموقع على الخوادم وجعله متاحًا للمستخدمين.
6. **الصيانة والتحديث (Maintenance & Updates):** تحسين الأداء، إصلاح الأخطاء، وإضافة ميزات جديدة بناءً على ملاحظات المستخدمين.

* شكل الجداول والنماذج في نظام الويب

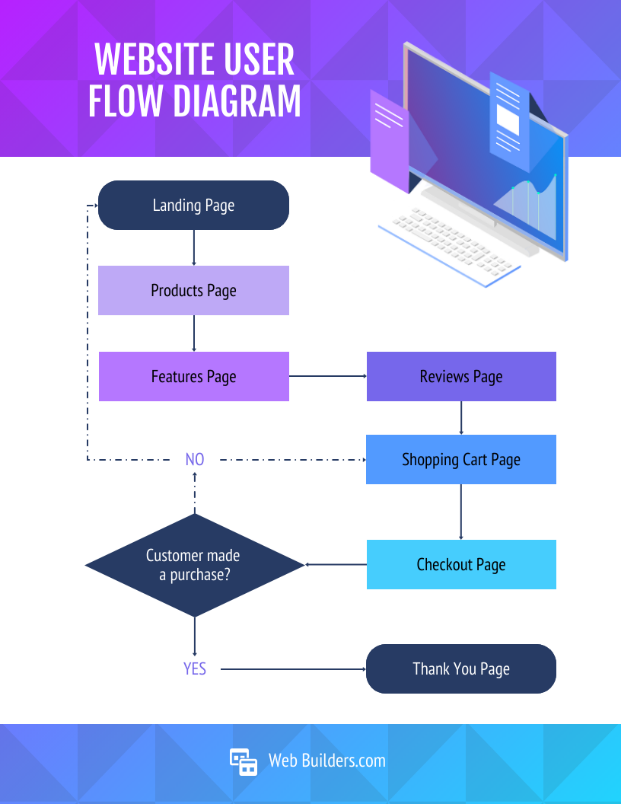


**النماذج في قواعد البيانات (Schema Design):** عند تصميم قاعدة بيانات لموقع إلكتروني، يمكن استخدام العلاقات بين الجداول:

* **جدول المستخدمين (Users Table)** يحتوي على بيانات المستخدمين.
* **جدول الطلبات (Orders Table)** يحتوي على عمليات الشراء ويرتبط بالمستخدمين عبر **User\_ID**.
* **جدول المنتجات (Products Table)** يحتوي على تفاصيل المنتجات المتاحة للشراء.

**الملخص**

* **نظام الويب يتكون من المتصفح، الخادم، قاعدة البيانات، وطبقة التطبيق.**
* **يمر بتخطيط، تصميم، تطوير، اختبار، نشر، وصيانة مستمرة.**
* **الجداول في قواعد البيانات تخزن البيانات بطريقة منظمة، بينما النماذج في الويب تُستخدم لجمع البيانات من المستخدمين.**
* **مخطط لعمل نظام الويب**



### ****أهم أدوات وتقنيات تطوير الويب****

#### **1. بيئة Visual Studio Code (VS Code)**

**Visual Studio Code** هو أحد أشهر بيئات التطوير المتكاملة (IDE) لتطوير الويب، وهو مفتوح المصدر ومقدم من **Microsoft**. يتميز بدعم **JavaScript, HTML, CSS, Node.js** والعديد من اللغات الأخرى، بالإضافة إلى مجموعة من الإضافات (Extensions) التي تسهل تطوير المشاريع.

#### **2. حزمة لغة JavaScript (JavaScript Standard Edition)**

**JavaScript** هي اللغة الأساسية لتطوير تطبيقات الويب الديناميكية، وهي جزء لا يتجزأ من **HTML و CSS**. يتم تطويرها بواسطة **ECMAScript Standard**، وتستخدم في المتصفحات لإضافة التفاعلية والتحكم في العناصر.

#### **3. طريقة تثبيت بيئة تطوير الويب (Setting up a Web Development Environment)**

لتثبيت بيئة تطوير متكاملة للويب، يمكن اتباع الخطوات التالية:

1. **تحميل وتثبيت Visual Studio Code** أو أي محرر كود مثل **Sublime Text أو Atom**.
2. **تثبيت Node.js** لإنشاء بيئة تشغيل للـ JavaScript من جانب الخادم.
3. **إعداد نظام التحكم في الإصدارات Git** لإدارة المشاريع ومزامنتها مع **GitHub أو GitLab**.
4. **تنصيب متصفح حديث** مثل **Google Chrome أو Firefox** لاختبار التطبيقات.
5. **إعداد قواعد البيانات** مثل **MySQL أو PostgreSQL أو MongoDB** إذا كان المشروع يحتاج إلى قاعدة بيانات.

#### **4. تقنية قاعدة البيانات Firebase (Firebase Realtime Database)**

**Firebase** هو نظام أساسي للخدمات السحابية مقدم من **Google**، يوفر قواعد بيانات **Realtime Database و Firestore** التي تعمل بدون الحاجة إلى خادم، مما يسمح بحفظ البيانات ومزامنتها بين المستخدمين في الوقت الحقيقي.

#### **5. ربط مشروع ويب بتقنية Firebase**

لربط موقع ويب بـ **Firebase**، يتم تنفيذ الخطوات التالية:

1. إنشاء حساب على **Firebase Console** وإضافة مشروع جديد.
2. اختيار **Web App** وإضافة **Firebase SDK** إلى المشروع.
3. استخدام **Firebase Authentication** لتسجيل الدخول باستخدام Google أو Facebook.
4. تخزين البيانات باستخدام **Firestore Database** أو **Realtime Database**.
5. نشر المشروع باستخدام **Firebase Hosting**.

**النتائج والتوصيات**

**النتائج**

* تم تطوير منصة إلكترونية متكاملة تتيح للسياح التخطيط لرحلاتهم بسهولة.
* إمكانية ربط المنصة بالمؤسسات السياحية والفنادق لتقديم خدمات إضافية.
* تحسين تجربة المستخدم من خلال توفير بيانات محدثة وموثوقة.
* إضافة قسم **الآراء والتقييمات** لدعم تفاعل المستخدمين ومشاركة تجاربهم.
* إنشاء قسم خاص **للمرشدين السياحيين المعتمدين** ليسهل على السياح العثور على مرشد مناسب.
* تقديم فرصة تعليمية لطلاب اللغات لممارسة لغاتهم مع السياح والمرشدين.

**التوصيات**

* **تطوير تطبيق جوال ليكون مكملًا للمنصة الإلكترونية.**
* **إضافة ميزة الذكاء الاصطناعي لاقتراح الوجهات السياحية بناءً على تفضيلات المستخدم.**
* **دمج وسائل الدفع الإلكتروني لحجز التذاكر والخدمات عبر الإنترنت.**
* **تعزيز ميزة تعدد اللغات لتشمل عددًا أكبر من اللغات العالمية.**
* **توسيع قاعدة بيانات المرشدين السياحيين لزيادة الخيارات المتاحة للسياح.**
* **التعاون مع الجامعات لتقديم فرص تدريبية لطلاب اللغات من خلال المنصة.**

**الخاتمة**

إنشاء منصة إلكترونية للسياحة في مصر يعد خطوة مهمة نحو تعزيز قطاع السياحة الرقمية، حيث يتيح هذا النظام للسياح الحصول على معلومات موثوقة ومحدثة بسهولة. بفضل هذه المنصة، يمكن تحسين تجربة السياح وزيادة عدد الزوار، مما يساهم في دعم الاقتصاد الوطني. كما أن دعم تعدد اللغات، إضافة قسم المرشدين السياحيين، وربط الطلاب بالسياح، يجعل المنصة أكثر شمولًا وتأثيرًا.

**المراجع**

* وزارة السياحة المصرية
* مصادر رقمية متخصصة في تطوير المنصات الإلكترونية
* تجارب سابقة في مجال السياحة الرقمية

**تمت بحمد الله**