



جامعة تشرين

كلية الهندسة المعلوماتية

قسم البرمجيات ونظم المعلومات

السنة الخامسة

موقع ويب لتوفير فرص عمل في مختلف المجالات

A website that provides job opportunities in various fields

مشروع تخرج

إعداد:

- سليمان اسامه ميا

- مجد حسين حجي محمود

إشراف:

الدكتور رائد جابري

2023 -2024

المقدمة:

يعتبر العمل أحد أهم العوامل التي تؤثر على حياة الشباب، حيث يمثل لهم وسيلة لتحقيق الاستقلال المالي وتطوير الذات. في عالم اليوم، يتزايد عدد الشباب الذين يدخلون سوق العمل بشكل مستمر، مما يبرز أهمية توفير فرص عمل مناسبة تلبي طموحاتهم وتطلعاتهم.

تتعدد الأسباب التي تدفع الشباب للبحث عن العمل، منها الرغبة في تحسين مستوى المعيشة، واكتساب الخبرات العملية، وبناء شبكة من العلاقات الاجتماعية والمهنية. ومع تقدم التكنولوجيا وتغيرات السوق، تظهر العديد من الفرص الجديدة في مجالات متعددة مثل التكنولوجيا، التسويق الرقمي، والابتكار، على الرغم من وجود العديد من الفرص المتاحة، يواجه الشباب تحديات عدة في العثور على وظائف تتناسب مع مهاراتهم وطموحاتهم. يتطلب الأمر منهم تطوير مهارات جديدة والتكيف مع متطلبات السوق المتغيرة باستمرار. لذا، فإن تعزيز برامج التدريب والتوجيه المهني يعد أمرًا ضروريًا لمساعدتهم على الاندماج بنجاح في سوق العمل.

الفهرس

| | |
|----|--------------------------------------|
| 2 | المقدمة |
| 5 | قائمة الاشكال |
| 7 | تمهيد |
| 8 | ١ - توصيف النظام وتحليله |
| 9 | ١-١ المتطلبات الوظيفية |
| 9 | ١-٢ المتطلبات غير الوظيفية |
| 10 | ٣-١ مخططات UML |
| 10 | Class Diagram ○ |
| 11 | Sequence Diagram ○ |
| 12 | Activity Diagram ○ |
| 13 | BPMN ○ |
| 14 | UseCase Diagram ○ |
| 15 | ١-٢ أهمية مواقع الويب |
| 16 | ٢-٢ عناصر مواقع الويب |
| 17 | ٣-٢ أنواع مستعرضات مواقع الويب |
| 18 | ٤-٢ لغات مواقع الويب |
| 20 | ٥-٢ اختبارات مواقع الويب |

| | |
|----|------------------------------|
| 21 | التقنيات المستخدمة ٣-٣ |
| 21 |Laravel version 10 ٣-١ |
| 24 | Mvc معمارية ٣-٢ |
| 35 | React JS ٣-٣ |
| 48 | الخاتمة ٤-٤ |
| 49 | المراجع ٥-٥ |

قائمة الأشكال :

- 10 Class Diagram : ١ الشكل -
- 11 Sequeunce Diagram: ٢ الشكل -
- 12Activity Diagram ; ٣ الشكل -
- 13BPMN : ٤ الشكل -
- 14usecase Diagram : ٥ الشكل -
- 25user Model : ٦ الشكل -
- 25 JOB Model: ٧ الشكل -
- 26Category Model : ٨ الشكل -
- 27 Logout and Login Controller : ٩ الشكل -
- 28 Login : ١٠ خطوات عملية الشكل -
- 28 Logout : ١١ خطوات عملية الشكل -
- 28 Category : ١٢ خطوات إنشاء الشكل -
- 29 Category : ١٣ خطوات تعديل الشكل -
- 29Category : ١٤ خطوات حذف الشكل -
- 29 : ١٥ خطوات إنشاء فرصة عمل جديدة الشكل -
- 30 : ١٦ خطوات تعديل فرصة عمل الشكل -
- 30 : ١٧ خطوات حذف فرصة عمل الشكل -

| | |
|----|--|
| 31 | Routes Api : ١٨ الشكل - |
| 32 | Routes : ١٩ الشكل - |
| 33 |Routes : ٢٠ الشكل - |
| 34 |Admin information : ٢١ الشكل - |
| 34 | withToken() && unverified() : ٢٢ الشكل - |
| 35 | Users Database : ٢٣ الشكل - |
| 35 | Categories Database : ٢٤ الشكل - |
| 40 |Dashboard : ٢٥ الشكل - |
| 41 |Users Page : ٢٦ الشكل - |
| 41 |Update User : ٢٧ الشكل - |
| 42 | : ٢٨ الشكل : الصلاحيات المتوفرة |
| 42 |Add User : ٢٩ الشكل - |
| 42 |Categories : ٣٠ الشكل - |
| 43 |Update Category : ٣١ الشكل - |
| 43 |Jobs Page : ٣٢ الشكل - |
| 43 | Update Job Details : ٣٣ الشكل - |
| 44 |search : ٣٤ الشكل - |
| 44 |home page : ٣٥ الشكل - |
| 45 |categories Page : ٣٦ الشكل - |
| 45 |jobs Page : ٣٧ الشكل - |

- الشكل ٣٨ : Login Page 46

- الشكل ٣٩ : Register page 67

- الشكل ٤٠ : 403 Page 47

- الشكل ٤١ : 404 Page 47

- تمهيد :

تعتبر مواقع الويب من أبرز الابتكارات التكنولوجية التي غيرت طريقة تفاعل الأفراد والمجتمعات مع المعلومات والخدمات. منذ ظهور الإنترنت، أصبحت المواقع الإلكترونية جزءاً لا يتجزأ من حياتنا اليومية، حيث توفر لنا وسيلة للوصول إلى مجموعة واسعة من المحتوى، بدءاً من الأخبار والمعلومات التعليمية وصولاً إلى الترفيه والتجارة.

1 - توظيف النظام وتحليله :

يستطيع الأدمن انشاء مستخدم جديد، ويستطيع تعديل بيانات المستخدم (الاسم، والايمل) ، كما يستطيع تعديل صلاحية المستخدم من مستخدم عادي إلى مستخدم admin او مستخدم ناشر لفرص العمل (writer) وبالعكس، يستطيع الادمن انشاء category جديد كما يمكنه حذف وتعديل بيانات category موجود سابقا ، يستطيع الادمن إضافة فرص عمل جديدة كما يمكنه تعديل وحذف فرص عمل موجودة سابقاً ، كما يمكنه البحث عن طريق اسم المستخدم او عن طريق تاريخ انشاء المستخدم ، وايضا يمكنه البحث في قسم ال category او قسم jobs عن طريق الاسم والتاريخ ، يستطيع ال writer رؤية قسم ال category وال job يمكنه تعديل وإضافة وحذف اي category او job ، كما يمكنه البحث عن طريق الاسم والتاريخ، لا يستطيع writer رؤية قسم users في ال dashboard

يستطيع المستخدم العادي رؤية فرص العمل المتوفرة في الموقع مع إمكانية رؤية تفاصيل العمل و ايميل الشركة الناشرة لهذه الفرصة ، كما يمكنه البحث عن وظيفة معينة عن طريق اسمها ، ويمكنه رؤية ال categories المتوفرة التي يتضمن كل category فرص العمل المتعلقة بهذا ال category ، كما يمكنه البحث ضمن ال category عن طريق الاسم.

يمكن للمستخدم العادي رؤية نصائح عامة عن كيفية عمل cv احترافي و نصائح عن مقابلات الأعمال عبارة عن playlists من الفيديوهات القصيرة لكل قسم من النصائح كما يمكنه المستخدم إمكانية التواصل مع إدارة الموقع إذا كان هناك اي تساؤل او اذا كان يرغب بإخباره عن الجديد من فرص العمل المتوفرة في الموقع يمكن للمستخدم انشاء حساب علي الموقع عن طريق إدخال اسمه و ايميله وكلمة السر ، كما يمكن المستخدم تسجيل الدخول اذا كان يملك حسابا بشكل مسبق عن طريق إدخال ايميله وكلمة السر بعد مطابقة اذا كان يملك حسابا في قواعد البيانات ويمكنه تسجيل الدخول عن طريق حساب google، ويمكنه تسجيل الخروج

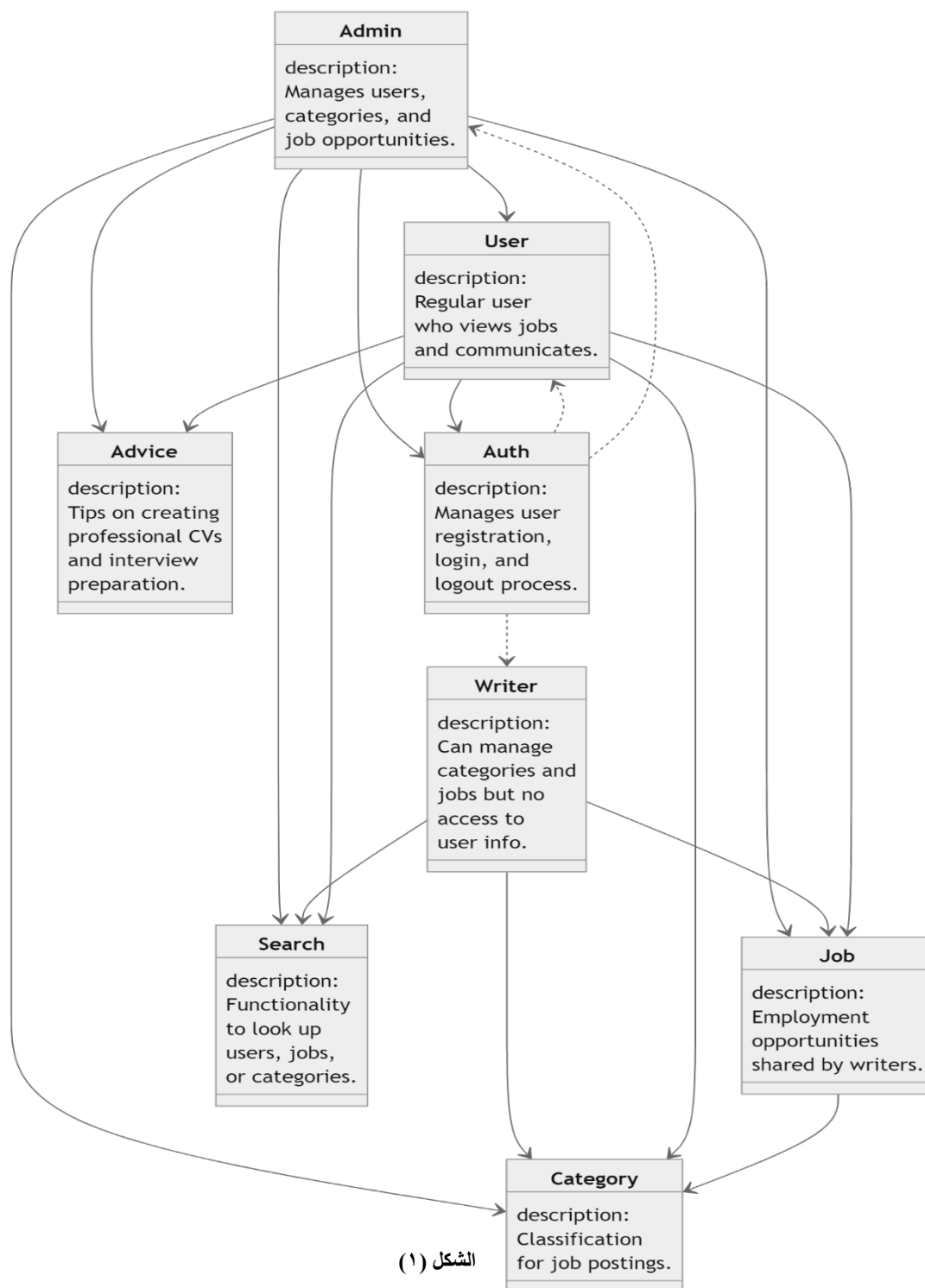
1.1 - المتطلبات الوظيفية:

- ١ - إمكانية استعراض جميع فرص العمل مع تفاصيل كل عمل.
- ٢ - إمكانية عرض جميع أصناف العمل .
- ٣ - إمكانية رؤية نصائح وخطوات عن كيفية كتابة cv احترافي.
- ٤ - إمكانية رؤية نصائح عن مقابلات العمل.
- ٥ - إمكانية التواصل مع إدارة الموقع.
- ٦ - إمكانية البحث عن وظائف العمل عن طريق الاسم.
- ٧ - يستطيع المدير من إضافة مستخدمين مع تعديلهم وحذفهم وتغيير الصلاحيات.
- ٨ - يستطيع المدير من إضافة فرص عمل و تعديلهم وحذفهم .
- ٩ - يستطيع المدير من إضافة category جديد و تعديله وحذفه.
- ١٠ - يستطيع المدير البحث عن فرص العمل وال category عن طريق الاسم او عن طريق تاريخ انشائهم .

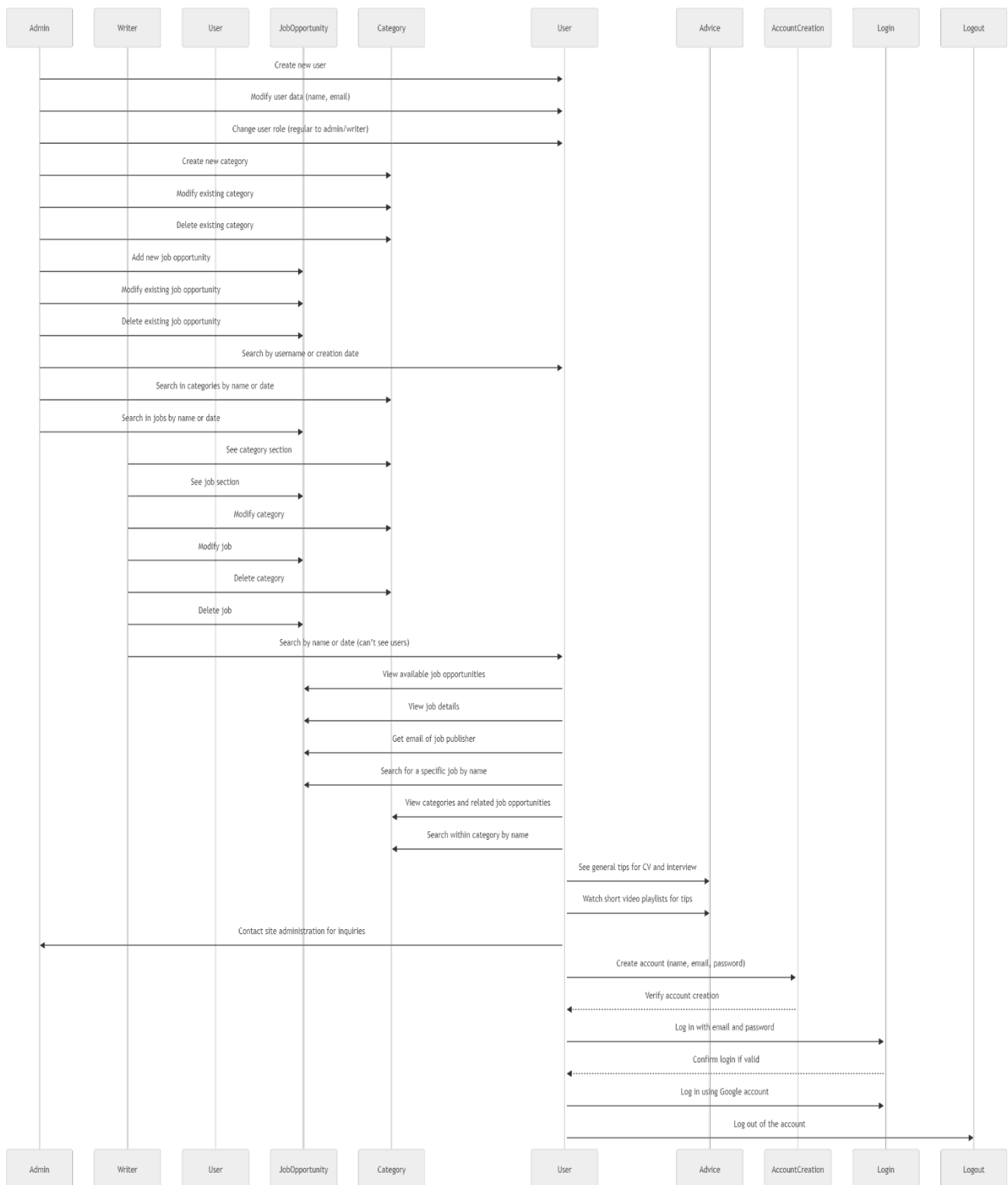
2.1 - المتطلبات غير الوظيفية :

- ١ - الموقع مناسب مع جميع الشاشات باستخدام bootstrap-React
- ٢ - البيانات قابلة للوصول فقط عن طريق المستخدمين المصرح لهم بالوصول حيث إن سماحيات الوصول تختلف باختلاف مستخدم النظام
- ٣ - النظام مقسم إلى ملفات منفصلة بحيث يتم التعديل بسلاسة وسهولة أكبر
- ٤ - النظام قابل التوسع في المستقبل وإضافة ميزات جديدة مثل انشاء cv احترافي والتواصل المباشر مع إدارة الشركة الناشرة لفرص العمل
- ٥ - حماية حساب المستخدم من عملية الإختراق إذا تم تعديل او استخدام ال token الخاص بالمستخدم المولد عن عملية تسجيل الدخول للموقع او عملية انشاء حساب على الموقع أو تم حذفه

○ Class Diagram

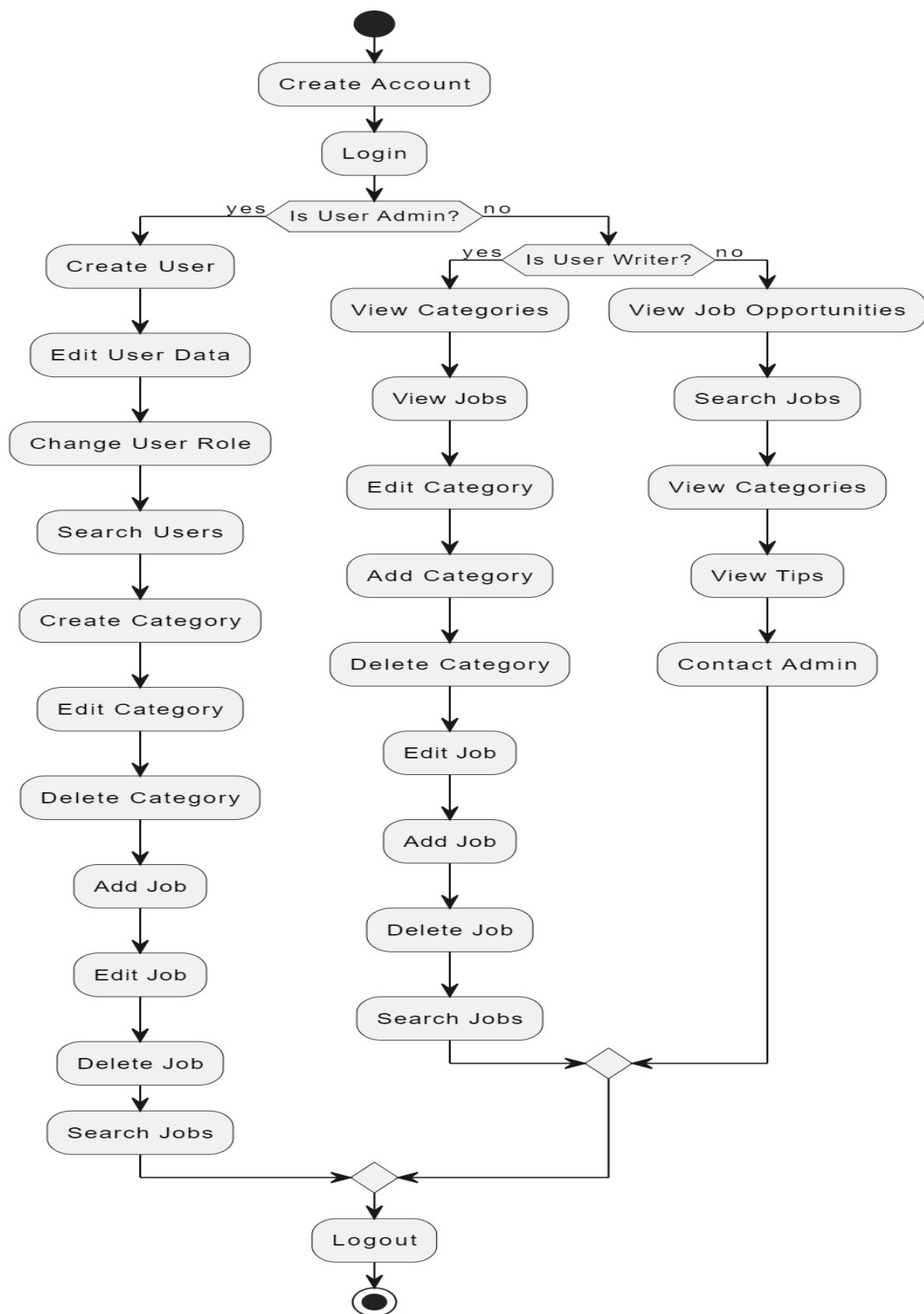


○ Sequence Diagram



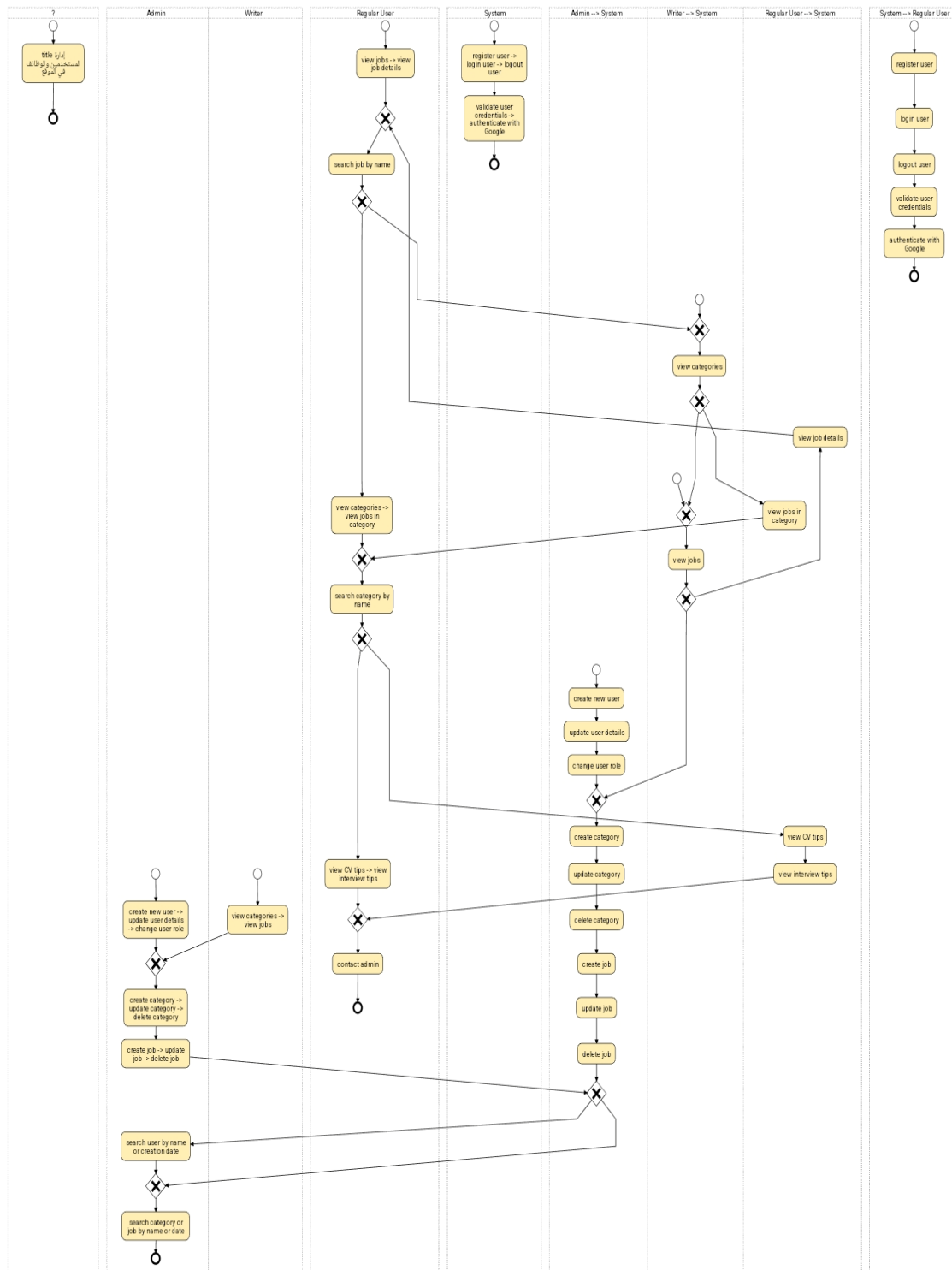
الشكل (٢)

○ Activity Diagram



الشكل (٣)

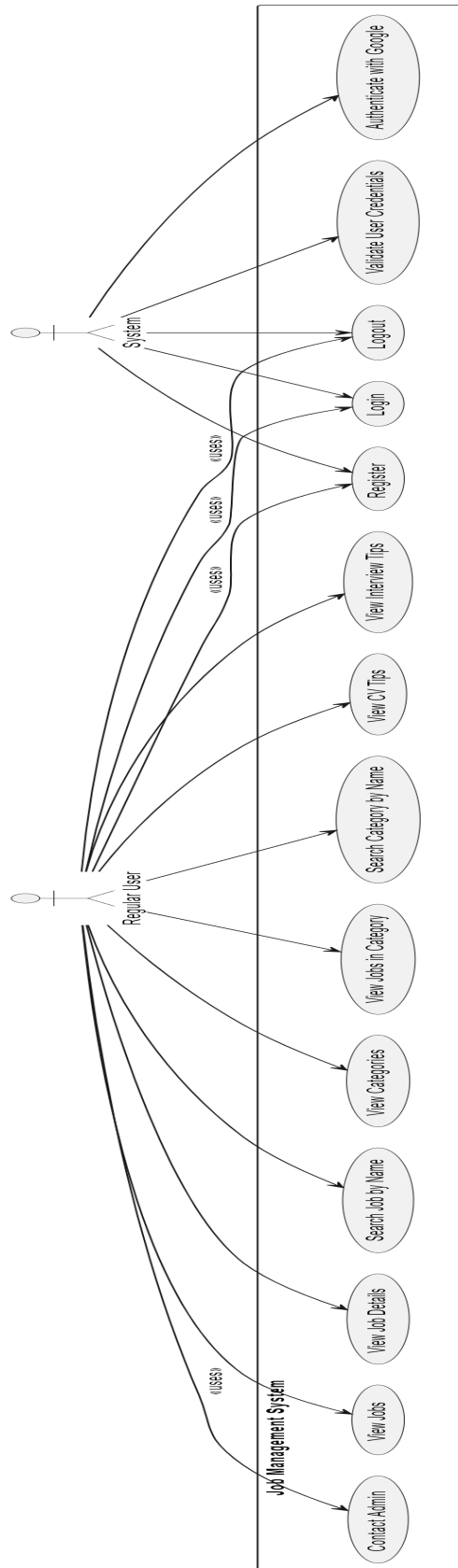
○ BPMN



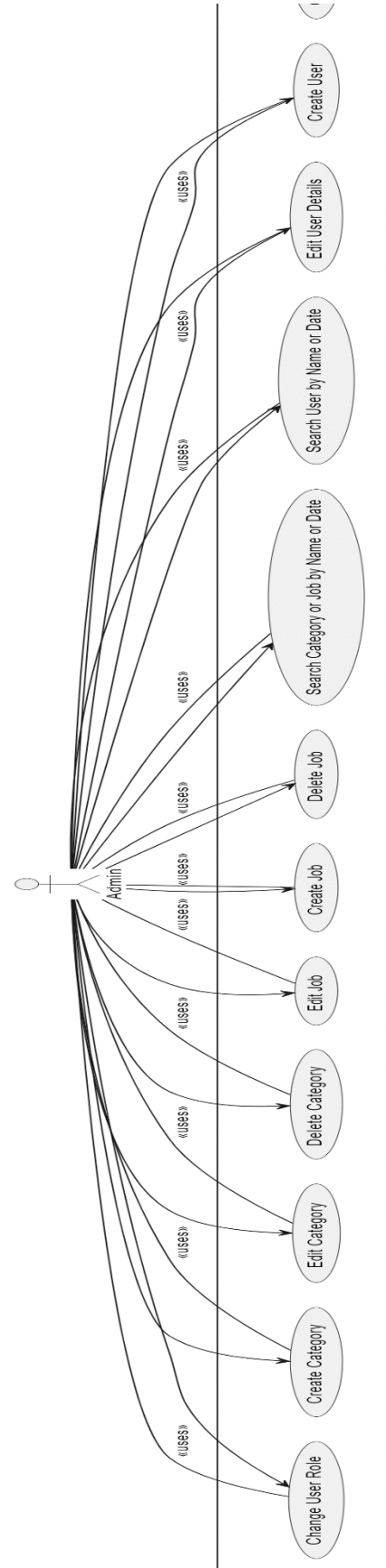
الشكل (٤)

○ usecase Diagram

Use Case Diagram for Job Management System



الشكل (٥)



١,٢ أهمية مواقع الويب :

تعتبر المواقع الإلكترونية أداة قوية للتواصل والتفاعل في عالمنا الرقمي. إليك بعض النقاط المهمة:

١. التواجد على مدار الساعة: تمكن المواقع الإلكترونية الشركات والأفراد من التواصل مع العملاء والجمهور على مدار الساعة، دون قيود زمنية.
٢. بناء المصداقية والثقة: تصميم موقع احترافي يعكس مصداقية العلامة التجارية ويبني الثقة لدى العملاء.
٣. زيادة المبيعات والأرباح: يمكن للمواقع الإلكترونية أن تكون متجرًا عبر الإنترنت لبيع المنتجات والخدمات، مما يزيد من فرص البيع والأرباح².
٤. تقليل التكاليف: توفر المواقع الإلكترونية تكاليف أقل مقارنةً بالمتاجر الفعلية، مثل تكاليف الإيجار والموظفين.
٥. زيادة الرؤية والتواجد العالمي: يمكن للمواقع الإلكترونية أن تصل إلى جمهور عالمي، مما يزيد من الوعي بالعلامة التجارية والمنتجات.

٢,٢ عناصر مواقع الويب:

عناصر مواقع الويب تتنوع بناءً على نوع الموقع والغرض منه، ولكن هناك بعض العناصر الأساسية التي غالبًا ما توجد في معظم المواقع. إليك قائمة بأهم عناصر مواقع الويب:

١. الرأس (Header):

- يتضمن عادةً الشعار، عنوان الموقع، وقائمة التنقل الرئيسية. يُعتبر نقطة البداية للمستخدمين للتوجه إلى صفحات مختلفة.

٢. قائمة التنقل (Navigation Menu):

- تسهل التنقل بين الصفحات المختلفة للموقع. يمكن أن تكون قائمة أفقية أو عمودية وتحتوي على روابط لأقسام الموقع المختلفة.

٣. المحتوى الرئيسي (Main Content):

- يتضمن النصوص، الصور، الفيديوها، والمعلومات الأخرى التي تشكل جوهر الموقع وتقدم القيمة للمستخدمين.

٤. الشريط الجانبي (Sidebar):

- يمكن أن يحتوي على روابط إضافية، مقالات ذات صلة، أو إعلانات. يُستخدم لتعزيز التجربة العامة للمستخدم.

٥. الذيل (Footer):

- يحتوي على معلومات إضافية مثل حقوق النشر، روابط السياسة الخصوصية، شروط الاستخدام، وروابط التواصل الاجتماعي.

٦. نموذج الاتصال (Contact Form):

- يتيح للمستخدمين التواصل مع مالكي الموقع أو الدعم الفني. يتضمن عادةً حقولاً للاسم، البريد الإلكتروني، والرسالة.

٧. أزرار الدعوة إلى العمل (Call to Action Buttons):

- تُستخدم لتوجيه المستخدمين نحو إجراء معين مثل التسجيل، الشراء، أو تحميل المحتوى.

٨. صورة الغلاف (Hero Image/Banner):

- صورة كبيرة تظهر في الجزء العلوي من الصفحة وتكون جذابة بصرياً لجذب انتباه الزوار.

٩. محتوى الوسائط الاجتماعية (Social Media Links):

- روابط تؤدي إلى حسابات وسائل التواصل الاجتماعي الخاصة بالموقع أو الشركة، مما يعزز التفاعل الاجتماعي.

١٠. محرك البحث الداخلي (Search Bar):

- يسمح للمستخدمين بالبحث عن محتوى محدد داخل الموقع بسهولة.

١١. التعليقات والمراجعات (Comments and Reviews):

- توفر مساحة للمستخدمين للتفاعل مع المحتوى من خلال إضافة تعليقات أو مراجعات، مما يعزز التفاعل.

١٢. الشهادات والتوصيات (Testimonials):

- عرض آراء العملاء السابقين حول المنتجات أو الخدمات، مما يعزز الثقة والمصداقية.

١٣. الخرائط (Maps):

- تُستخدم لتحديد موقع الشركة أو تقديم معلومات حول الأماكن الجغرافية.

١٤. الرسوم المتحركة والتفاعلات (Animations and Interactions):

- تُستخدم لتحسين تجربة المستخدم من خلال إضافة عناصر تفاعلية تجعل الموقع أكثر جذبًا.

٣,٢ أنواع مستعرضات مواقع الويب:

مستعرضات مواقع الويب، أو المتصفحات، تأتي بأنواع متعددة، وكل نوع له ميزاته وخصائصه. إليك أبرز أنواع مستعرضات مواقع الويب:

١. متصفحات سطح المكتب :

- جوجل كروم (Google Chrome): متصفح سريع وشائع، يدعم مجموعة واسعة من الإضافات ويتميز بواجهة مستخدم بسيطة.

- موزيلا فايرفوكس (Mozilla Firefox): يركز على الخصوصية والأمان، ويقدم ميزات تخصيص واسعة.

- مايكروسوفت إيدج (Microsoft Edge): متصفح جديد نسبيًا يعتمد على محرك Chromium، مع ميزات مثل القراءة المريحة وتكامل مع خدمات مايكروسوفت.

- سفاري (Safari): متصفح أبل، معروف بأدائه العالي وفعاليته في استهلاك الطاقة على أجهزة ماك وآيفون.

- أوبرا (Opera): يقدم ميزات فريدة مثل VPN مدمج ووضع توفير البيانات

٢. متصفحات الهواتف المحمولة

- جوجل كروم للهواتف المحمولة: يقدم تجربة مشابهة لنسخة سطح المكتب مع ميزات ملائمة للهواتف.

- سفاري للهواتف المحمولة: النسخة المحمولة من متصفح سفاري، مصممة بشكل خاص لأجهزة آيفون وآيباد.

- فايرفوكس للآندرويد: يوفر ميزات الخصوصية والأمان على الهواتف الذكية.

- أوبرا ميني (Opera Mini): مصمم لتقليل استهلاك البيانات وتحسين سرعة التصفح على الشبكات البطيئة.

٣. متصفحات خاصة بالخصوصية :

- تور (Tor Browser): مصمم لحماية الخصوصية، يستخدم شبكة تور لإخفاء هوية المستخدم.

- (Brave Browser): يركز على حماية الخصوصية ويقوم بحظر الإعلانات وتتبع البيانات بشكل افتراضي.

٤. متصفحات للأغراض الخاصة :

- متصفحات الألعاب: مثل "Opera GX" المصمم خصيصًا لتلبية احتياجات اللاعبين.

- متصفحات المؤسسات: مثل "Microsoft Edge" في نسخته المؤسسية، التي تأتي مع ميزات أمان إضافية وخصائص إدارة.

٥. متصفحات قديمة أو بديلة :

- إنترنت إكسبلورر (Internet Explorer): كان شائعًا في السابق ولكنه أصبح أقل استخدامًا مع ظهور متصفحات جديدة.

- كونكروور (Konqueror) وكاميليو (Camino): متصفحات أقل شهرة تستخدم في بعض أنظمة التشغيل أو البيئات الخاصة.

٢, ٤ لغات مواقع الويب:

لغات تطوير مواقع الويب تتنوع وتختلف بناءً على الغرض منها، سواء كان لتصميم الصفحات، أو البرمجة، أو إدارة البيانات. إليك أبرز اللغات المستخدمة في تطوير مواقع الويب

١. لغات تصميم الويب

- HTML (HyperText Markup Language):

لغة الترميز الأساسية لإنشاء هيكل صفحات الويب. تستخدم لتحديد العناصر مثل النصوص، الصور، الروابط، والجداول.

- CSS (Cascading Style Sheets):

تُستخدم لتنسيق مظهر صفحات الويب. تتيح للمطورين التحكم في الألوان، الخطوط، التخطيطات، والتفاعلات.

٢. لغات البرمجة من جهة العميل (Client-side)

- JavaScript: لغة برمجة ديناميكية تُستخدم لإضافة تفاعلية إلى صفحات الويب. يمكن استخدامها لإنشاء تأثيرات متحركة، التحقق من البيانات، والتفاعل مع المستخدم.

٣. لغات البرمجة من جهة الخادم (Server-side)

- PHP (Hypertext Preprocessor): لغة برمجة شائعة لتطوير التطبيقات الديناميكية على الويب. تُستخدم بشكل واسع في أنظمة إدارة المحتوى مثل ووردبريس.

- Python: تُستخدم في تطوير الويب عبر أطر عمل مثل Django وFlask. تتميز بالبساطة وسهولة القراءة.

- Ruby: تُستخدم بشكل رئيسي مع إطار عمل Ruby on Rails، وتساعد في بناء تطبيقات ويب قوية وسريعة.

- Java: تُستخدم في تطوير تطبيقات الويب الكبيرة والمعقدة، خاصة في المؤسسات.

- Node.js: بيئة تشغيل JavaScript على الخادم، تتيح للمطورين استخدام JavaScript لتطوير التطبيقات من جهة الخادم.

٤. قواعد البيانات (Databases)

- SQL (Structured Query Language): لغة تُستخدم لإدارة قواعد البيانات، مثل MySQL وPostgreSQL. تُستخدم لاستعلام البيانات وإدارتها.

- NoSQL: مثل MongoDB، تُستخدم لتخزين البيانات غير الهيكلية وتعتبر مثالية للتطبيقات التي تحتاج إلى مرونة في البيانات.

٥. أطر العمل (Frameworks)

- Frontend Frameworks

- React: مكتبة JavaScript لبناء واجهات المستخدم.

- Vue.js: إطار عمل JavaScript لبناء واجهات المستخدم بشكل تفاعلي.

- Angular: إطار عمل شامل لتطوير تطبيقات الويب.

- Backend Frameworks :

- Express.js: إطار عمل بسيط وقوي لتطبيقات Node.js.

- Django: إطار عمل Python يوفر الكثير من الميزات المدمجة لتطوير الويب.

- Ruby on Rails: إطار عمل Ruby يركز على الإنتاجية وسرعة التطوير.

٦. تقنيات أخرى

- TypeScript: لغة برمجة تعتمد على JavaScript وتضيف ميزات مثل الأنواع الثابتة.

- Sass و LESS: لغات ما قبل المعالجة لـ CSS التي تساعد في تنظيم وتسهيل كتابة أنماط CSS.

٢, ٥ اختبارات مواقع الويب

اختبارات مواقع الويب هي عملية تقييم جودة وأداء الموقع الإلكتروني من جوانب متعددة. تشمل هذه الاختبارات عدة مجالات مثل الأداء، الأمان، تجربة المستخدم، والتوافق مع المعايير. إليك بعض الأنواع الشائعة لاختبارات مواقع الويب:

١. اختبارات الأداء :

- سرعة التحميل: قياس الوقت الذي يستغرقه تحميل الصفحة بالكامل.

- تحمل الضغط: اختبار كيفية أداء الموقع تحت ضغط عدد كبير من المستخدمين في نفس الوقت.

- استجابة الخادم: قياس الوقت الذي يستغرقه الخادم للاستجابة لطلبات المستخدمين.

٢. اختبارات الأمان :

- تقييم الثغرات: البحث عن نقاط الضعف في الموقع مثل SQL Injection أو XSS.

- اختبار التحقق من الهوية: التأكد من أن إجراءات التحقق من الهوية تعمل بشكل صحيح.

- تحليل الأمان: استخدام أدوات لتحليل الأمان مثل OWASP ZAP أو Nessus.

٣. اختبارات تجربة المستخدم (UX)

- اختبار قابلية الاستخدام: تقييم مدى سهولة استخدام الموقع من قبل الزوار.
- استطلاعات الرأي: جمع آراء المستخدمين حول تجربتهم في الموقع.
- اختبار A/B: مقارنة نسختين من صفحة معينة لمعرفة أيهما يحقق أداءً أفضل.

٤. اختبارات التوافق :

- التوافق مع المتصفحات: التأكد من أن الموقع يعمل بشكل صحيح على مختلف المتصفحات (Chrome، Firefox، Safari، إلخ).
- التوافق مع الأجهزة: اختبار أداء الموقع على أجهزة مختلفة مثل الهواتف الذكية والأجهزة اللوحية وأجهزة الكمبيوتر.
- ٥. اختبارات الوضوح والوظائف :
- اختبار الوظائف: التأكد من أن جميع الروابط والأزرار والنماذج تعمل كما هو متوقع.

- اختبار المحتوى: مراجعة المحتوى للتأكد من خلوه من الأخطاء اللغوية والنحوية.

أدوات لاختبار مواقع الويب

- Google PageSpeed Insights: لتحليل سرعة تحميل الصفحات.
- GTmetrix: لتحليل أداء الموقع وتقديم توصيات للتحسين.
- Selenium: لأتمتة اختبارات الوظائف.
- Lighthouse: أداة لتحليل الأداء وتجربة المستخدم والأمان.

٣ – التقنيات المستخدمة :

1.3 - Laravel version 10 :

Laravel هو إطار عمل PHP مفتوح المصدر يستخدم لتطوير تطبيقات الويب. يتميز بكونه سهل الاستخدام ومرن، ويحتوي على مجموعة واسعة من الميزات التي تسهل عملية التطوير.

الميزات الجديدة في Laravel 10 :

١. تحسين الأداء: تم تحسين الأداء العام للإطار، مما يجعل التطبيقات تعمل بشكل أسرع.
٢. تحديثات على نظام التوجيه: تم تحسين نظام التوجيه ليكون أكثر كفاءة وسهولة في الاستخدام.
٣. أداة Artisan: تم إضافة أوامر جديدة لأداة Artisan، مما يسهل إدارة التطبيق.
٤. تحسينات على Eloquent: تم إضافة ميزات جديدة لـ Eloquent ORM، مثل تحسينات في استعلامات قاعدة البيانات.
٥. دعم أفضل للاختبارات: تم تحسين أدوات الاختبار، مما يسهل كتابة اختبارات فعالة للتطبيقات.
٦. تحسينات على التوثيق: تم تحديث وثائق Laravel لتكون أكثر شمولاً وسهولة في الفهم.

المتطلبات الأساسية :

لتنصيب Laravel 10، تحتاج إلى:

- PHP 8.1 أو أعلى

- Composer (مدير الحزم لـ PHP)

- قاعدة بيانات (مثل MySQL، SQLite، PostgreSQL)

كيفية تثبيت Laravel 10 :

يمكنك تثبيت Laravel 10 باستخدام Composer. افتح سطر الأوامر واستخدم الأمر التالي:

```
composer create-project --prefer-dist laravel/laravel your-project-name
```

هيكل المشروع:

عند إنشاء مشروع Laravel جديد، ستحصل على هيكل مجلدات متسق يشمل:

- /app: يحتوي على منطق التطبيق.
- /config: يحتوي على ملفات تكوين التطبيق.
- /database: يحتوي على ملفات قاعدة البيانات والمهاجرات.
- /resources: يحتوي على الموارد مثل القوالب وملفات اللغة.
- /routes: يحتوي على ملفات التوجيه.
- /public: يحتوي على الملفات العامة مثل CSS وJS.

إعداد قاعدة البيانات :

لتكوين قاعدة البيانات، يمكنك تعديل ملف env. في جذر المشروع:

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=your_database_name
DB_USERNAME=your_username
DB_PASSWORD=your_password
```

إنشاء نموذج (Model) ومهاجرة (Migration) :

لإنشاء نموذج ومهاجرة باستخدام Artisan، يمكنك استخدام الأوامر التالية:

```
php artisan make:model Post -m
```

هذا الأمر سيقوم بإنشاء نموذج Post وملف مهاجرة له.

تنفيذ المهاجرات :

لتنفيذ المهام وإنشاء الجداول في قاعدة البيانات، استخدم الأمر:

```
php artisan migrate
```

إنشاء وحدة تحكم (Controller) :

لإنشاء وحدة تحكم جديدة، يمكنك استخدام الأمر:

```
php artisan make:controller PostController
```

إعداد التوجيه (Routing) :

يمكنك إعداد التوجيه في ملف `routes/web.php` على سبيل المثال:

```
use App\Http\Controllers\PostController;
```

```
Route::get('/posts', [PostController::class, 'index']);
```

2.3 - معمارية MVC :

معمارية (MVC) (Model-View-Controller) هي نمط تصميم شائع يُستخدم في تطوير البرمجيات، خاصةً في تطبيقات الويب. يهدف هذا النمط إلى فصل مكونات التطبيق إلى ثلاثة أجزاء رئيسية، مما يسهل إدارة وتعزيز قابلية الصيانة والتوسع. دعنا نستعرض كل مكون على حدة:

١. النموذج (Model):

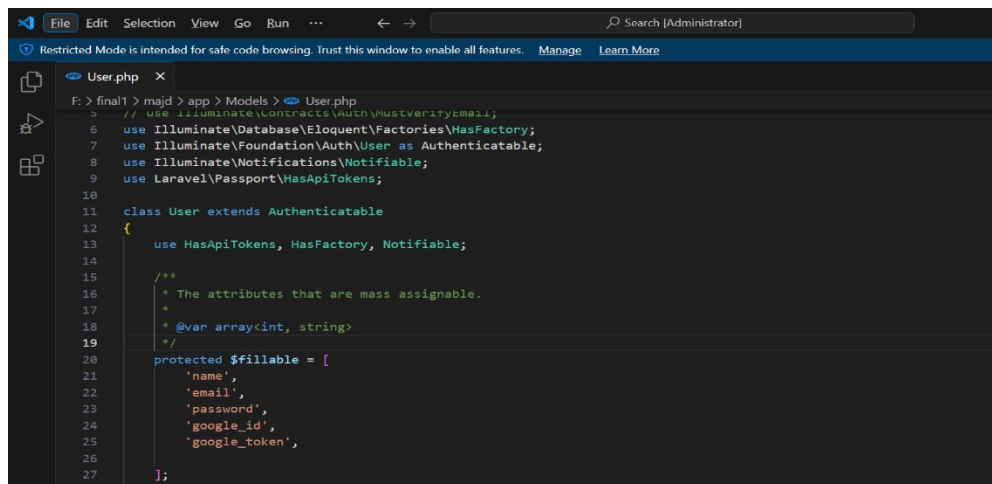
النموذج هو الجزء الذي يتعامل مع البيانات والمنطق التجاري للتطبيق. يشمل ذلك:

- البيانات: تمثل المعلومات التي يتعامل معها التطبيق، مثل بيانات المستخدمين أو المنتجات.

- العمليات: يحتوي على المنطق الذي يعالج البيانات، مثل عمليات الإضافة، التحديث، والحذف.

- التفاعل مع قاعدة البيانات: يتم من خلاله استرجاع البيانات أو حفظها في قاعدة البيانات.

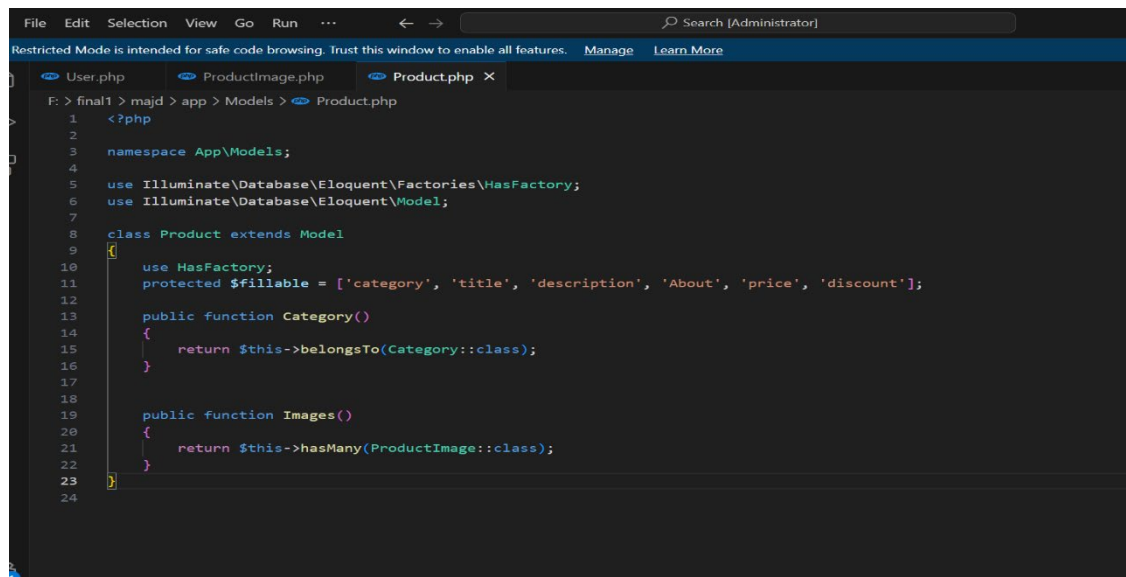
- هذا الكود هو جزء من نموذج (Model) في إطار عمل Laravel، وهو يستخدم لإنشاء نموذج يمثل المستخدمين في موقع الويب الخاص بنا.



الشكل (٦)

أولاً يتم استيراد بعض المكتبات الضرورية، مثل Authenticatable لتمكين ميزات المصادقة، و HasApiTokens لتمكين استخدام OAuth2 مع Laravel Passport ثم يُعرف كلاس User الذي يرث من Authenticatable، مما يعني أنه يمكن استخدامه لتسجيل الدخول والتحقق من الهوية .

- هذا الكود يمثل نموذج (Model) آخر في إطار عمل Laravel، وهو نموذج Product (فرصة العمل)



الشكل (٧)

أولاً : يتم استيراد المكتبتين الضروريتين:

- HasFactory: لتمكين استخدام المصانع (Factories) لإنشاء كائنات من النموذج بسهولة.

- Model: الكلاس الأساسي الذي يرث منه جميع النماذج في Laravel.

ثانياً: يتم تعريف علاقة belongsTo بين Product و Category.
تعني هذه العلاقة أن كل منتج ينتمي إلى فئة واحدة.

- هذا الكود يمثل نموذج (Model) في إطار عمل Laravel يُسمى Category

```
F:\final1 > majd > app > Models > Category.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 Category extends Model
9  {
10     use HasFactory;
11     protected $fillable = ['title', 'image'];
12 }
13
```

الشكل (٨)

٢. العرض (View)

العرض هو الجزء الذي يتعامل مع واجهة المستخدم. يُستخدم لعرض البيانات للمستخدمين وتقديم واجهة تفاعلية. تشمل مهام العرض:

- تقديم البيانات: عرض المعلومات المستخرجة من النموذج بطريقة مفهومة وجذابة.
- التفاعل مع المستخدم: استقبال المدخلات من المستخدم مثل النقرات أو الكتابات.

٣. المتحكم (Controller) :

المتحكم هو الجزء الذي يعمل كحلقة وصل بين النموذج والعرض. يتلقى المدخلات من العرض ويقوم بمعالجة هذه المدخلات بالتعاون مع النموذج. تشمل مهام المتحكم:

- معالجة المدخلات: يستقبل الأحداث من واجهة المستخدم (مثل النقرات) ويقرر كيفية الرد عليها.
- تحديث النموذج: يقوم بإجراء تغييرات على النموذج بناءً على المدخلات.
- تحديث العرض: بعد تغيير النموذج، يقوم المتحكم بتحديث العرض ليتناسب مع التغييرات الجديدة.

- هذا الكود هو جزء من وحدة تحكم (Controller) في تطبيق Laravel، ويختص بإدارة عمليات تسجيل الدخول وتسجيل الخروج للمستخدمين.

```

class AuthController extends Controller
{
    // Register Method
    public function Register(RegisterRequest $request)
    {
        $request->validated();
        $user = User::create([
            'name' => $request->name,
            'email' => $request->email,
            'password' => Hash::make($request->password),
        ]);
        $token = $user->createToken('token')->accessToken;
        $refreshToken = $user->createToken('authTokenRefresh')->accessToken;
        return response()->json([
            'user' => $user,
            'token' => $token,
        ], 200);
    }

    // Login Method
    public function Login(LoginRequest $request)
    {
        $request->validated();
    }
}

```

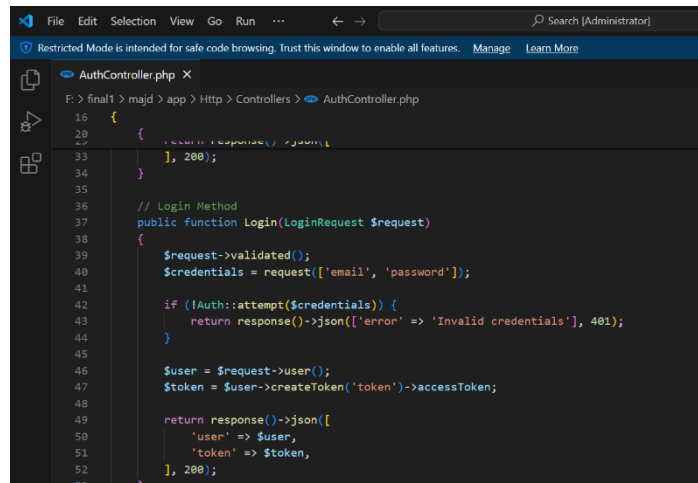
الشكل (٩)

- اولاً : يتم استيراد الفئات المطلوبة لاستخدامها في الكود. على سبيل المثال:
- LoginRequest و RegisterRequest: تستخدم للتحقق من صحة البيانات المدخلة عند تسجيل الدخول والتسجيل.
- User: نموذج المستخدم (Model).
- Auth: واجهة المصادقة.
- Hash: لتشفير كلمات المرور

class AuthController extends Controller;

- هذا السطر يحدد أن AuthController هو فئة تمتد من الفئة الأساسية Controller التي توفر وظائف أساسية لوحدات التحكم.

خطوات عملية (Login) :



```

16 {
17     // Login Method
18     public function Login(LoginRequest $request)
19     {
20         $request->validated();
21         $credentials = request(['email', 'password']);
22         if (!Auth::attempt($credentials)) {
23             return response()->json(['error' => 'Invalid credentials'], 401);
24         }
25         $user = $request->user();
26         $token = $user->createToken('token')->accessToken;
27         return response()->json([
28             'user' => $user,
29             'token' => $token,
30         ], 200);
31     }
32 }

```

الشكل (١٠)

خطوات عملية (Logout):



```

53 }
54 // Logout Method
55 public function logout(Request $request)
56 {
57     $request->user()->token()->revoke();
58     return response()->json(['message' => 'Successfully logged out'], 200);
59 }
60 }
61 }
62 }

```

الشكل (١١)

- خطوات إنشاء Category :



```

30 // Create Category Method
31 public function store(Request $request)
32 {
33     $category = new Category();
34     $request->validate([
35         'title' => 'required',
36         'image' => 'required|image'
37     ]);
38     $category->title = $request->title;
39     if ($request->hasFile('image')) {
40         $file = $request->file('image');
41         $filename = date('YmdHis') . '.' . $file->getClientOriginalExtension();
42         $path = 'images';
43         $file->move($path, $filename);
44         $category->image = url('/') . '/images/' . $filename;
45     }
46     $category->save();
47 }
48 }

```

الشكل (١٢)

- خطوات تعديل Category :

```

*/
public function edit(Category $category, $id, Request $request)
{
    $category = Category::findOrFail($id);
    $request->validate([
        'title' => 'required',
    ]);
    $category->title = $request->title;
    if ($request->hasFile('image')) {
        $oldpath = public_path() . '/images/' . substr($category['image'], strrpos($category['image'], '/') + 1);

        if (File::exists($oldpath)) {
            File::delete($oldpath);
        }
        $file = $request->file('image');
        $filename = date('YmdHis') . '.' . $file->getClientOriginalExtension();
        $category->image = url('/') . '/images/' . $filename;
        $path = 'images';
        $file->move($path, $filename);
    }
    $category->save();
}

```

الشكل (١٣)

- خطوات حذف Category :

```

92      */
93      public function destroy(Category $category, $id)
94      {
95          $category = Category::findOrFail($id);
96          $path = public_path() . '/images/' . substr($category['image'], strrpos($category['image'], '/') + 1);
97
98          if (File::exists($path)) {
99              File::delete($path);
100          }
101          $category->delete();
102      }
103  }
104

```

الشكل (١٤)

-خطوات انشاء فرصة عمل جديدة :

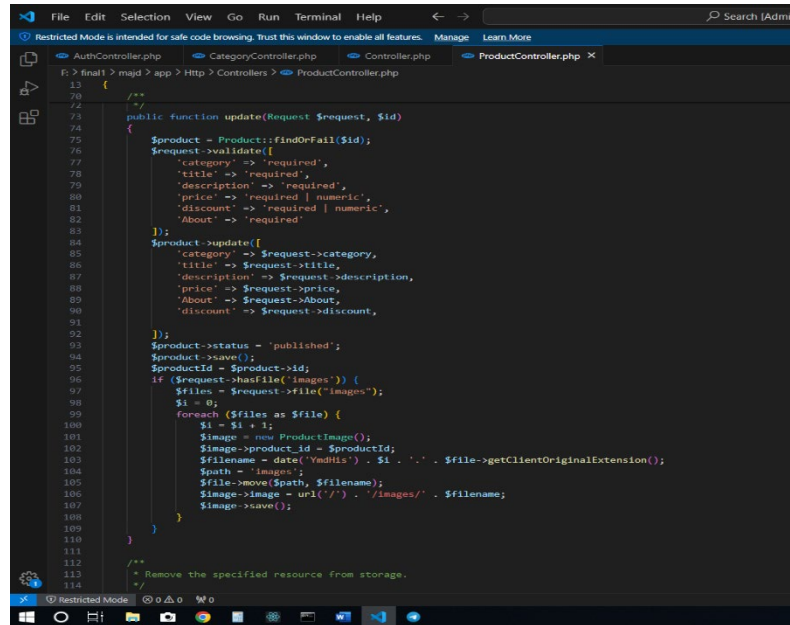
```

29
30      /**
31       * Store a newly created resource in storage.
32       */
33      public function store(Request $request)
34      {
35          $product = new Product();
36          $request->validate([
37              'title' => 'required',
38              'description' => 'required',
39              'price' => 'required | numeric',
40              'discount' => 'required | numeric',
41              'About' => 'required'
42          ]);
43          $productCreated = $product->create([
44              'category' => $request->category,
45              'title' => $request->title,
46              'description' => $request->description,
47              'price' => $request->price,
48              'About' => $request->About,
49              'discount' => $request->discount,
50          ]);
51          return $productCreated;
52      }

```

الشكل (١٥)

- خطوات تعديل فرصة عمل :



الشكل (١٦)

- خطوات حذف فرصة عمل :



الشكل (١٧)

- سنقوم بشرح ملف API الموجود في قسم Backend :

```

1 // Public Routes
2
3 Route::controller(AuthController::class)->group(function () {
4     Route::post('/register', 'register');
5     Route::post('/login', 'login');
6     Route::post('/passowrd', 'sendResetLink');
7     Route::post('/reset-password', 'reset');
8 });

```

الشكل (١٨)

1. Route::controller(AuthController::class):

- هذه السطر يستخدم لتعريف مجموعة من المسارات التي ترتبط بـ AuthController. هذا يعني أن جميع المسارات المعرفة في المجموعة ستستخدم الدوال الموجودة في AuthController.

2- Route::post('/register', 'register');

- هذا السطر يحدد مسار HTTP POST لعملية التسجيل (/register). عندما يتم إرسال طلب POST إلى هذا المسار، سيتم استدعاء الدالة register في AuthController.

3- Route::post('/login', 'login');

- هنا يتم تحديد مسار HTTP POST لعملية تسجيل الدخول (/login). عند إرسال طلب POST إلى هذا المسار، سيتم استدعاء الدالة login في AuthController.

4- Route::post('/passowrd', 'sendResetLink');

- هذا السطر يحدد مسار HTTP POST لإرسال رابط إعادة تعيين كلمة المرور (/passowrd). عند إرسال طلب POST إلى هذا المسار، سيتم استدعاء الدالة sendResetLink في AuthController.

5- Route::post('/reset-password', 'reset');

- هنا يتم تحديد مسار HTTP POST لإعادة تعيين كلمة المرور (/reset-password). عند إرسال طلب POST إلى هذا المسار، سيتم استدعاء الدالة reset في AuthController .

```
Route::middleware('checkAdmin')->controller(UsersController::class)->group(function () {  
    Route::get('/users', 'GetUsers');  
    Route::get('/user/{id}', 'getUser');  
    Route::post('/user/edit/{id}', 'editUser');  
    Route::post('/user/add', 'addUser');  
    Route::delete('/user/{id}', 'destroy');  
});
```

الشكل (١٩)

1. Route::get('/users', 'GetUsers');

- يستجيب لطلبات GET على /users.
- يقوم باستدعاء الدالة GetUsers في UsersController ، والتي هي مسؤولة عن إرجاع قائمة بجميع المستخدمين.

2. Route::get('/user/{id}', 'getUser');

- يستجيب لطلبات GET على /user/{id} حيث {id} هو معرف المستخدم.
- يستدعي الدالة getUser ، التي تعيد معلومات مستخدم.

3. Route::post('/user/edit/{id}', 'editUser');

- يستجيب لطلبات POST على /user/edit/{id}.
- يستدعي الدالة editUser ، والتي تستخدم لتحديث معلومات مستخدم معين.

4. Route::post('/user/add', 'addUser');

- يستجيب لطلبات POST على /user/add.
- يستدعي الدالة addUser ، والتي تستخدم لإضافة مستخدم جديد.

5. Route::delete('/user/{id}', 'destroy');

- يستجيب لطلبات DELETE على /user/{id}.
- يستدعي الدالة destroy ، والتي تستخدم لحذف مستخدم معين .

```
// Product Manger
Route::middleware('checkProductManager')->controller(CategoryController::class)->group(function () {
    Route::get('/categories', 'index');
    Route::get('/category/{id}', 'show');
    Route::post('/category/edit/{id}', 'edit');
    Route::post('/category/add', 'store');
    Route::delete('/category/{id}', 'destroy');
});
```

الشكل (٢٠)

1. Route::get('/categories', 'index');

- يستجيب لطلبات GET على /categories.
- يقوم باستدعاء الدالة index في CategoryController ، والتي تُستخدم لاسترجاع قائمة بجميع الفئات.

2. Route::get('/category/{id}', 'show');

- يستجيب لطلبات GET على /category/{id} حيث {id} هو معرف الفئة.
- يستدعي الدالة show ، التي تُستخدم لعرض معلومات فئة معينة بناءً على معرفها.

3. Route::post('/category/edit/{id}', 'edit');

- يستجيب لطلبات POST على /category/edit/{id}.
- يستدعي الدالة edit ، والتي تُستخدم لتحديث معلومات فئة معينة بناءً على معرفها.

4. Route::post('/category/add', 'store');

- يستجيب لطلبات POST على /category/add.
- يستدعي الدالة store ، والتي تُستخدم لإضافة فئة جديدة.

5. Route::delete('/category/{id}', 'destroy');

- يستجيب لطلبات DELETE على /category/{id}.
- يستدعي الدالة destroy ، والتي تُستخدم لحذف فئة معينة بناءً على معرفها.

- سنقوم بشرح بعض الاكواد الهامة في مجلد database :

```

18  */
19  public function definition()
20  {
21      return [
22          'name' => "admin",
23          'email' => 'admin@gmail.com',
24          'email_verified_at' => now(),
25          'role' => '1995',
26          'password' => Hash::make('admin123$%'), // password
27          'remember_token' => Str::random(10),
28      ];
29  }
30  
```

الشكل (٢١)

- هنا يتم تعريف بيانات admin الذي له ايميل خاص به وكلمة سر وصلاحيه مستقلة عن باقي مستخدمي النظام .

```

30
31  public function withToken()
32  {
33      return $this->afterCreating(function ($user) {
34          $user->createToken('token')->accessToken;
35      });
36  }
37  /**
38   * Indicate that the model's email address should be unverified.
39   */
40  public function unverified(): static
41  {
42      return $this->state(fn (array $attributes) => [
43          'email_verified_at' => null,
44      ]);
45  }
46  }
47  
```

الشكل (٢٢)

الدالة withToken() تقوم بإنشاء رمز وصول تلقائيًا للمستخدم عند إنشائه، مما يسهل عملية المصادقة في التطبيقات التي تستخدم واجهات برمجة التطبيقات.

- الدالة unverified() تستخدم لتحديد حالة نموذج المستخدم بحيث يكون البريد الإلكتروني غير مؤكد. يمكن استخدامها في حالات الاختبار أو عند الحاجة لإنشاء مستخدمين غير موثقين .

❖ تعريف قاعدة بيانات المستخدمين :

```
public function up(): void
{
    Schema::create('users', function (Blueprint $table) {
        $table->id();
        $table->string('name');
        $table->string('email')->unique();
        $table->timestamp('email_verified_at')->nullable();
        $table->string('password')->nullable();
        $table->string('role')->default('2001');
        $table->string('google_id')->nullable();
        $table->string('google_token')->nullable();
        $table->rememberToken();
        $table->timestamps();
    });
}
```

الشكل (٢٣)

❖ تعريف قاعدة بيانات الصنف Category :

```
Schema::create('categories', function (Blueprint $table) {
    $table->id();
    $table->text('title');
    $table->string('image');
    $table->timestamps();
});
```

الشكل (٢٤)

3.3 - React Js :

React.js هو مكتبة JavaScript لبناء واجهات المستخدم، تم تطويرها بواسطة Facebook. تُستخدم بشكل واسع لتطوير تطبيقات الويب التفاعلية، وتساعد المطورين على إنشاء واجهات مستخدم ديناميكية وسريعة.

المميزات الرئيسية لـ React.js :

١. المكونات (Components):

- React يعتمد على مفهوم المكونات، حيث يمكن تقسيم واجهة المستخدم إلى أجزاء صغيرة وقابلة لإعادة الاستخدام.

- كل مكون يمكن أن يحتوي على حالته الخاصة (State) وخصائصه (Props).

٢. DOM الافتراضي (Virtual DOM):

- React يستخدم DOM الافتراضي لتحسين أداء التطبيق. بدلاً من تحديث DOM الحقيقي مباشرة، يقوم React بتحديث DOM الافتراضي أولاً ثم يقارن التغييرات مع DOM الحقيقي ويطبق التغييرات اللازمة فقط.

٣. التفاعل مع البيانات (Data Binding):

- React يدعم الربط الأحادي الاتجاه للبيانات، مما يعني أن البيانات تتدفق في اتجاه واحد من المكون الأب إلى المكون الابن.

٤. JSX:

- JSX هو امتداد لكتابة JavaScript يسمح بكتابة كود HTML داخل JavaScript. يجعل كتابة المكونات أكثر سهولة ووضوحًا.

٥. الأداء:

- بفضل استخدام DOM الافتراضي وتقنيات التحسين، فإن React يوفر أداءً عاليًا حتى في التطبيقات الكبيرة والمعقدة.

الهيكل الأساسي لمشروع React:

عند إنشاء مشروع جديد باستخدام Create React App، ستحصل على الهيكل التالي:

```
my-app/
├── node_modules/
├── public/
├── | index.html
├── | favicon.ico
├── src/
├── | App.js
├── | index.js
```

— L App.css

كيفية تثبيت React:

يمكنك إنشاء مشروع جديد باستخدام Create React App عبر الأوامر التالية:

```
npx create-react-app my-app  
cd my-app  
npm start
```

إنشاء مكون بسيط :

لنفترض أنك تريد إنشاء مكون بسيط يعرض "Hello, World!". يمكنك إنشاء ملف Hello.js داخل مجلد src:

```
import React from 'react';  
function Hello () {  
  return <h1>Hello, World!</h1>; }  
export default Hello;
```

ثم يمكنك استخدامه في App.js:

```
import React from 'react';  
import Hello from './Hello';  
function App () {  
  return (  
    <div>  
      <Hello />  
    </div>  
  );  
}
```

```

        <div/>
    );}
export default App;

```

إدارة الحالة (State Management)

يمكنك استخدام useState hook لإدارة الحالة في مكونات الوظائف:

```

import React, { useState } from 'react';
function Counter () {
  const [count, setCount] = useState(0);
  return(
    <div>
      <p>You clicked {count} times</p>
      <button onClick={() => setCount(count + 1)}>
        Click me </button>
    </div>
  ) ;
}

```

تأثيرات جانبية (Side Effects)

يمكنك استخدام useEffect hook للتعامل مع التأثيرات الجانبية مثل جلب البيانات:

```

import React, { useEffect, useState } from 'react'
function DataFetching () {
  const [data, setData] = useState([]);
  useEffect(() => {
    fetch('https://api.example.com/data').

```

```
    then(response => response.json()).  
    then(data => setData(data)) ; },[]);  
.
```

[] تعني أن التأثير سيعمل مرة واحدة عند تحميل المكون

التوجيه (Routing) :

لإضافة التوجيه في تطبيق React، يمكنك استخدام مكتبة react-router-dom. قم بتنصيبها أولاً:

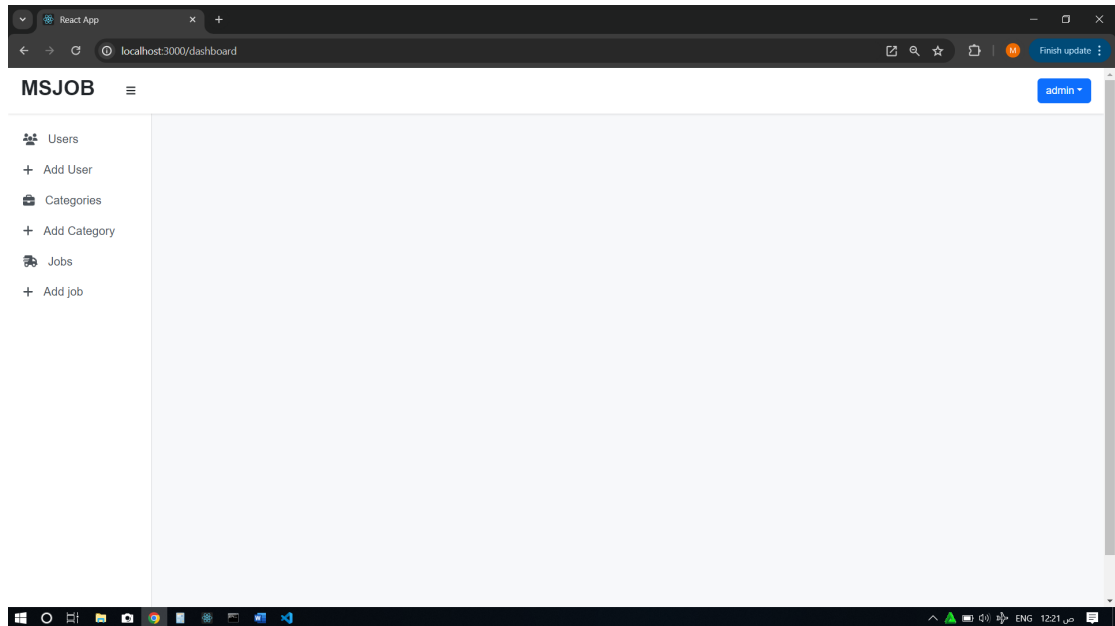
```
npm install react-router-dom
```

ثم يمكنك إعداد التوجيه كما يلي:

```
import { BrowserRouter as Router, Route, Switch } from 'react-router-dom';  
  
import Home from './Home';  
import About from './About';  
  
function App () {  
  return(  
    <Router>  
      <Switch>  
        <Route path="/" exact component={Home}>  
        </Route path="/about" component={About}>  
      </Switch>  
    </Router>  
  );  
}
```

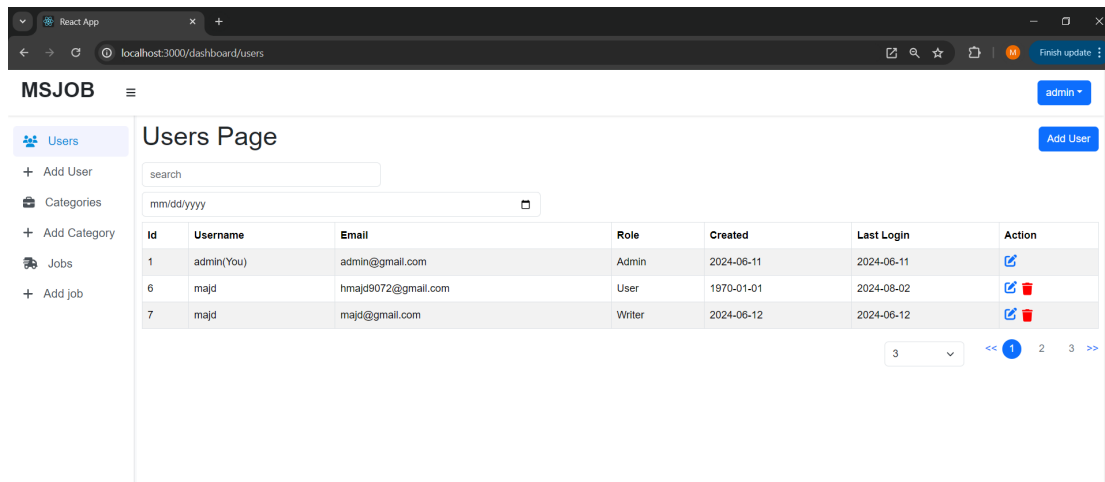
واجهات الموقع :

○ Dashboard Department



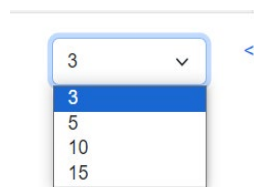
الشكل (٢٥)

○ Users

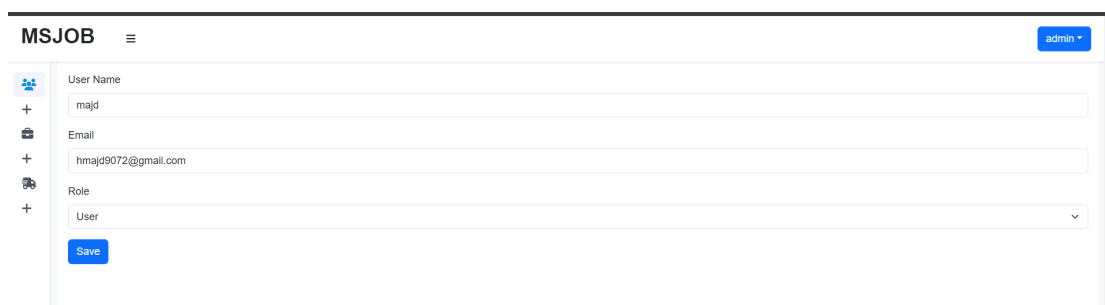


الشكل (٢٦)

- تظهر في هذه الواجهة أسماء المستخدمين المسجلين على الموقع حيث يحتوي الجدول السابق على اسم المستخدم وإيميله وصلاحيته (مستخدم عادي , writer, Admin) , ومتى تم انشاء هذه المستخدم مع إمكانية حذفه وتعديله , كما يوجد في اسفل الصفحة Pagenation لكي يقسم ال Users مع إمكانية تغيير عدد العناصر المعروضة

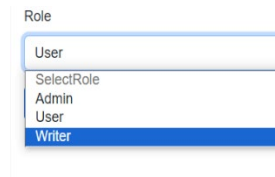


○ Update User



الشكل (٢٧)

- حيث يقوم بجلب بيانات المستخدم الحالية مع إمكانية التعديل من قبل Admin , مع إمكانية تعديل الصلاحية
- الصلاحيات المتوفرة :



الشكل (٢٨)

○ Add User

MSJOB

admin

User Name

name...

Email

name@example.com...

Password

Enter Password...

Role

SelectRole

Save

الشكل (٢٩)

○ Categories

MSJOB

admin

Categories Page

search

mm/dd/yyyy

| Id | Title | Image | Created | Updated | Action |
|----|--------------------------|-------|------------|------------|--------|
| 16 | Administrative Assistant | | 2024-06-20 | 2024-06-20 | |
| 17 | Nurse | | 2024-06-20 | 2024-06-20 | |
| 18 | Electrician | | 2024-06-20 | 2024-06-20 | |
| 19 | Mechanical Engineer | | 2024-06-20 | 2024-06-20 | |

3

<< 1 2 3 ... 10 11 12 13 >>

الشكل (٣٠)

○ Update Category

MSJOB

admin

Title

Nurse

Image

Choose File No file chosen

Save

الشكل (٣١)

○ Jobs

MSJOB

admin

Add Job

Jobs Page

search

mm/dd/yyyy

| ID | Image | Title | Description | Salary | Extra work | Created | Updated | Action |
|----|-------|----------------------|--|--------|------------|------------|------------|---|
| 71 | | Territory Management | Managing assigned sales territories, including planning and prioritizing customer visits, prospecting activities, and sales calls. part-time Lattaka | 4520 | 0 | 2024-06-20 | 2024-08-01 | Edit Delete |
| 72 | | Market Intelligence | Gathering information about competitors, market trends, and customer preferences to inform sales strategies and tactics. part-time AlHasska | 6950 | 0 | 2024-06-20 | 2024-08-01 | Edit Delete |
| 73 | | Customer Support: | Providing ongoing support to customers after a sale, addressing inquiries, resolving issues, and ensuring customer satisfaction. Full time Hama | 9865 | 0 | 2024-06-20 | 2024-08-01 | Edit Delete |
| 74 | | Employee Engagement | Creating initiatives to boost employee morale, satisfaction, and engagement through recognition programs, events, and feedback mechanisms. Full-time Damasus | 8595 | 0 | 2024-06-20 | 2024-08-01 | Edit Delete |

3

<< 1 2 3 ... 11 12 13 14 >>

الشكل (٣٢)

○ Update Job details

MSJOB

admin

Sales Representative

Title

Territory Management

Description

Managing assigned sales territories, including planning and prioritizing customer visits, prospecting activities, and sales calls. || part-time || Lattaka

Salary

4520

Extra Work

5

About

Email : hmajd@9072@gmail.com || from 10/8 to 20/8

Images

Upload Images

الشكل (٣٣)

-حيث يمكن رفع صورة مع كل فرصة عمل (بالمستقبل اذا اردنا تطوير الموقع سنقوم بوضع صورة مع كل فرصة عمل لتدل على RoadMap).

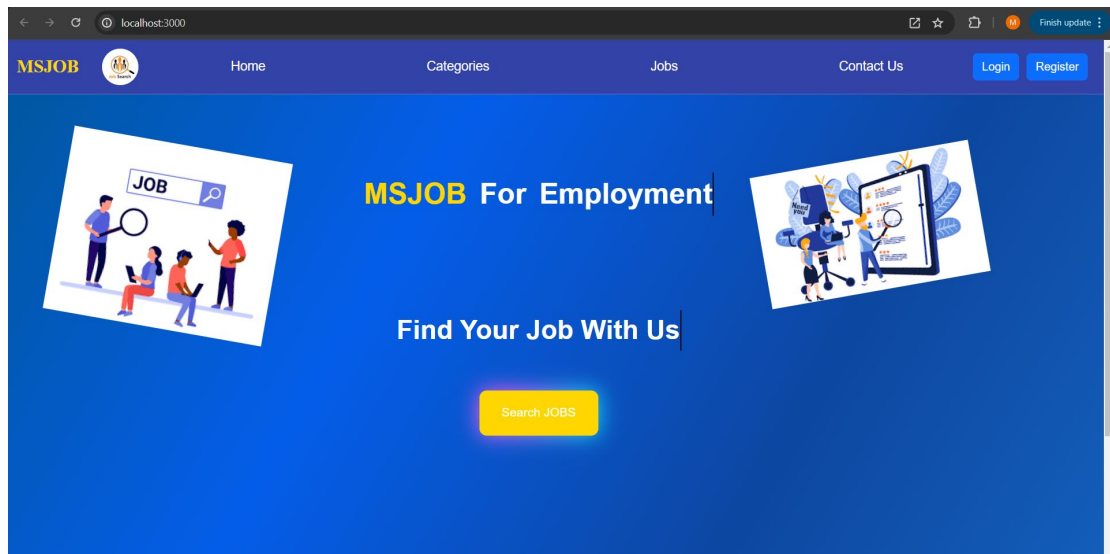
- أيضاً مع خيار البحث عن طريق الاسم او عن طريق تاريخ معين في الأقسام الثلاثة .



The image shows a web form with two input fields. The first field is labeled 'search' and the second field is labeled 'mm/dd/yyyy'. There is a small square icon to the right of the second field.

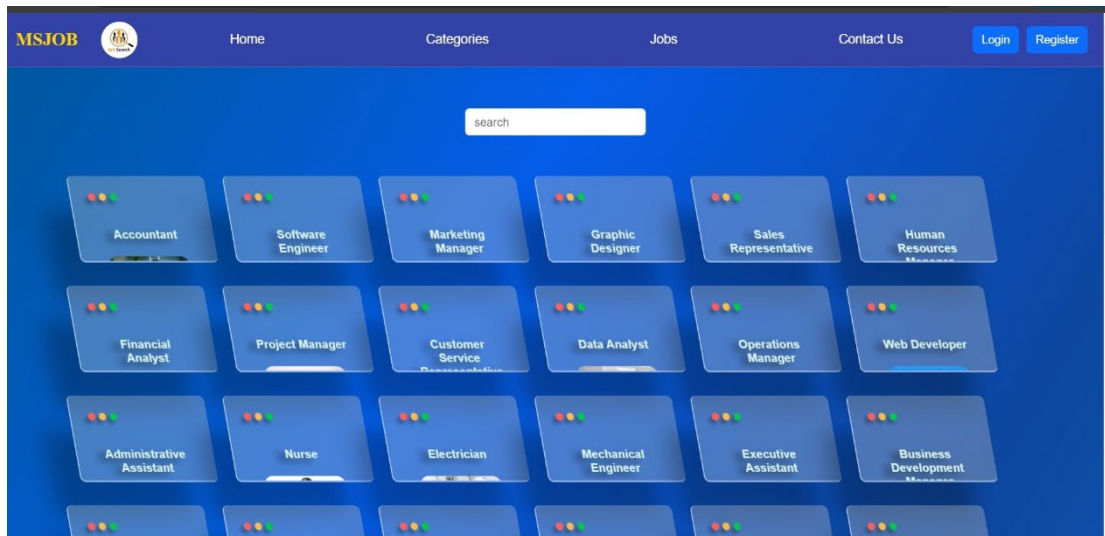
الشكل (٣٤)

○ Home page



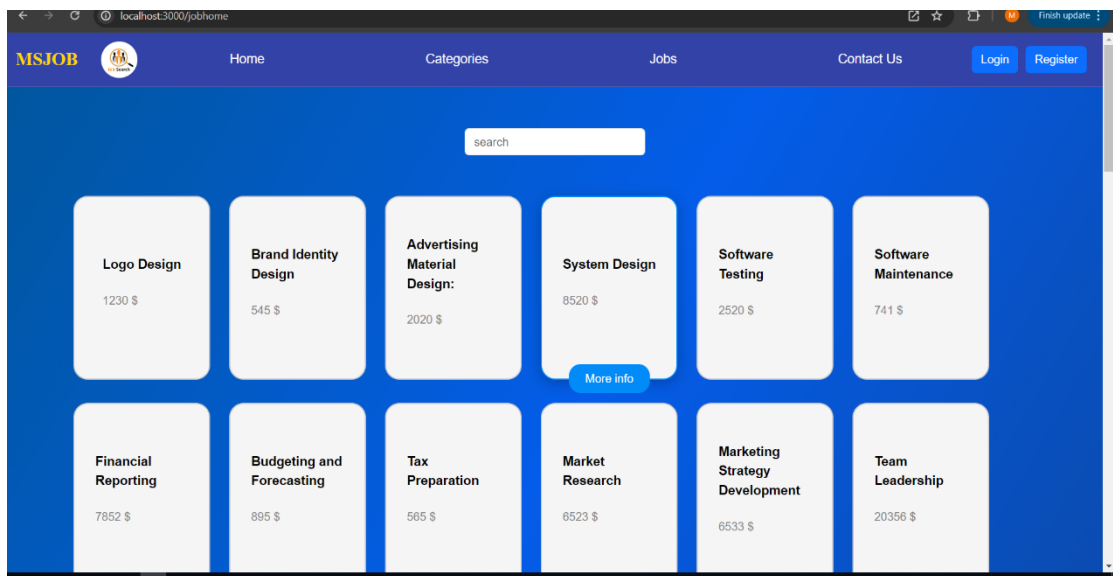
الشكل (٣٥)

○ Categories Page



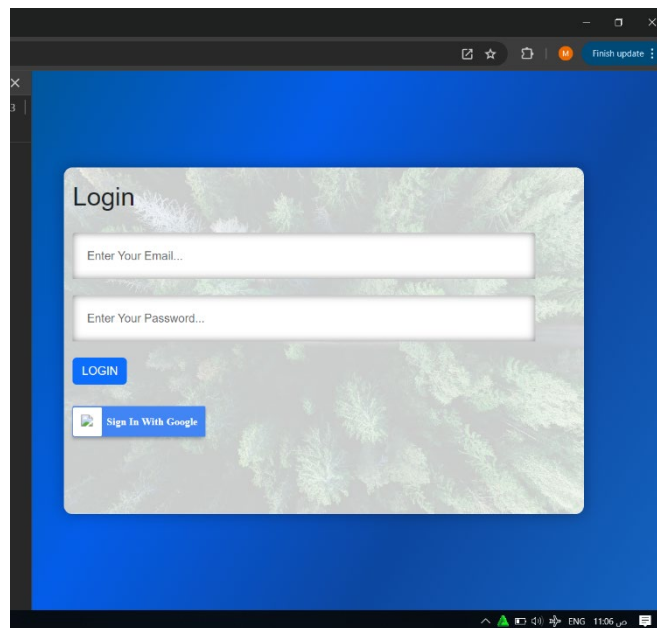
الشكل (٣٦)

○ Jobs Page



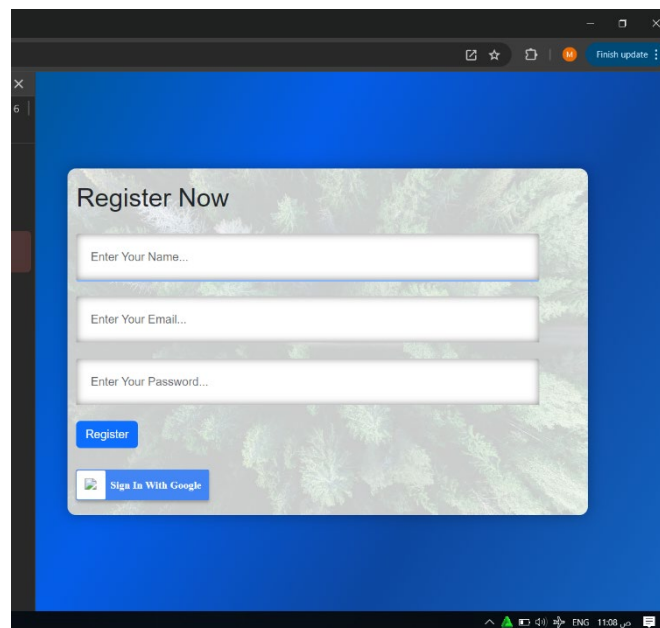
الشكل (٣٧)

- **Login Page**



الشكل (٣٨)

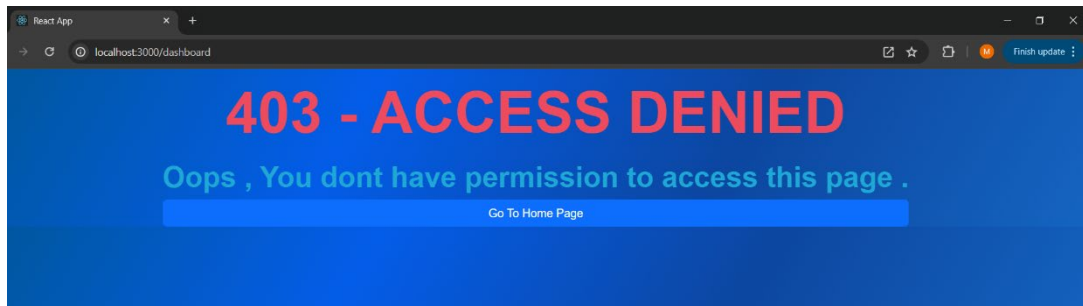
- **Register page**



الشكل (٣٩)

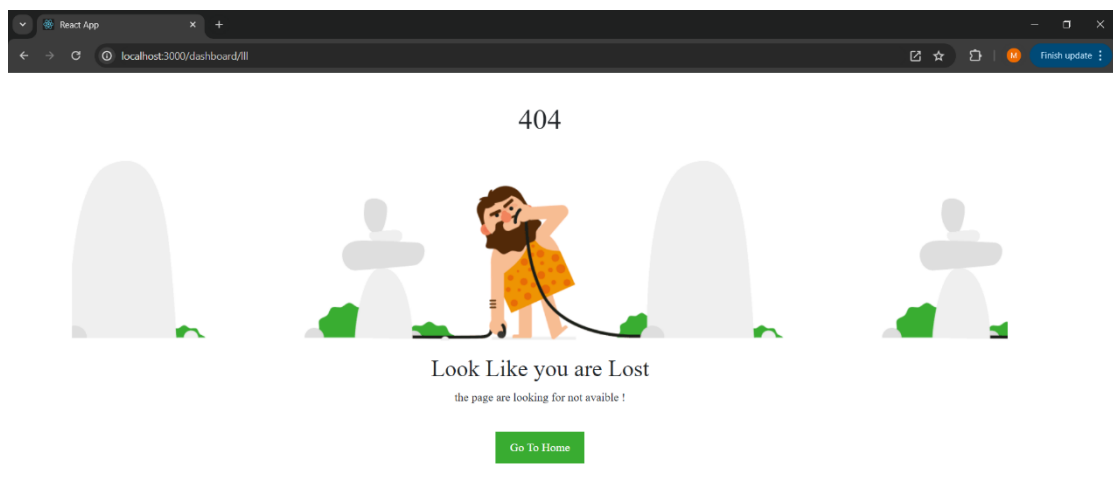
- اذا كان المستخدم مستخدم عادي وطلب الدخول الى Dashboard ستظهر له شاشة تعبر انه غير مصرح للدخول الى dashboard .

○ 403 Page



الشكل (٤٠)

○ 404 Page



الشكل (٤١)

٤ - الخاتمة:

في الختام، يمكن القول إن تكنولوجيا مواقع الويب تمثل حجر الزاوية في عالمنا الرقمي اليوم. فقد أحدثت ثورة في كيفية تواصلنا وتفاعلنا مع المعلومات والخدمات، مما جعلها جزءاً لا يتجزأ من حياتنا اليومية. من خلال التطورات المستمرة في تصميم المواقع، وأدوات البرمجة، وتقنيات تحسين تجربة المستخدم، أصبح بإمكان الأفراد والشركات على حد سواء إنشاء منصات مبتكرة تلبي احتياجاتهم وتوفر قيمة حقيقية للمستخدمين.

[1] Object Management Group,(2024)

"UML", <https://www.uml.org.com/>.

[2]Twitter (2024)

"bootstrap",<https://getbootstrap.com/docs/5.3/getting-started/introduction.com> .

[3]Laravel LLC ,(2024)

"Laravel", <https://laravel.com/>.

<https://laravel.com/docs/10.x/readme>.

[4] Microsoft (2024)

"MVC", <https://dotnet.microsoft.com> .

[5]Netscape Communications Corpoaration(2024).

"JavaScript", <https://www.javascript.com> .

[6]Meta Platforms(2024)

"React Js" , <https://reactjs.org>