



وزارة التعليم العالي

المعهد العالي للادارة وتكنولوجيا المعلومات

بكر الشيخ

مشروع التخرج شعبة علوم الحاسوب

Movie Reviews Project

إشراف

- أ.م.د / عمرو ابراهيم الشورى
- أ.م.د / ريهام مصطفى عيسى
- د / أميرة أحمد العطار
- د / منى محمد موافي

إعداد

- محمد عماد محمد مخلوف
- احمد سمير محمود احمد
- اسراء علي محمد الشورية
- ابراهيم خالد احمد محمد سيد احمد

شكر وتقدير

بدأنا بأكثر من يد وفاسينا أكثر من هم وعانيانا الكثير من الصعوبات وها نحن اليوم والحمد لله نطوي سهر الليالي وتعب الأيام وخلاصة مشوارنا بين دقى هذا العمل المتواضع امتنعت الكلمات عن التعبير .. ورق القلم أن يسبر .. ولكن أبى القلب ألا أن يبوح لما يخالجه من مشاعر تقىض شكرًا وامتنانا وثناء .. ثناء على من نذروا أنفسهم لخدمتنا امتنانا لما بذلوه حتى انتهاء هذا العمل المتواضع نتوجه بخالص الشكر لراعي الجغرافية ومرببها في المعهد ، إلى من علمنا التفاؤل والمضي إلى الأمام ، إلى من رعاانا وحافظ علينا ، إلى من وقف إلى جانبنا عندما ضللنا الطريق ...

الأستاذ الدكتور / محمد سامي راضي

عميد المعهد

شكر وتقدير

لابد لنا ونحن نخطو خطواتنا الأخيرة في الحياة الجامعية من وقفة نعود إلى أعوام قضيناها في رحاب الجامعة مع أساتذتنا الكرام الذين قدموا لنا الكثير باذلين بذلك جهودا كبيرة في بناء جيل الغد لتبعث الأمة من جديد ... وقبل أن نمضي نقدم أسمى آيات الشكر الامتنان والتقدير والمحبة إلى الذين حملوا أقدس رسالة في الحياة .. إلى الذين مهدوا لنا طريق العلم والمعرفة إلى جميع أساتذتنا الأفاضل ... "كن عالما .. فإن لم تستطع فكن متعلم ، فإن لم تستطع فأحب العلماء ، فإن لم تستطع فلا تبغضهم" .

ونخص بالشكر والتقدير إلى

الأستاذ الدكتور / عمرو الشورى

رئيس قسم الحاسوب الآلي

شكر وتقدير

في مثل هذه اللحظات يتوقف اليراع ليفكر قبل أن يخط الحروف ليجمعها في كلمات ..
تتبادر الحروف وعبثاً أن يحاول تجميعها في سطور سطورة كثيرة تمر في الخيال ولا يبقى لنا
في نهاية المطاف إلا قليلاً من الذكريات وصور تجمعنا برفاق كانوا إلى جانبنا .. فواجب علينا
شكراً لهم ووداعهم ونحن خطوا خطواتنا الأولى في غمار الحياة ونخص بالجزيل الشكر والعرفان
إلى كل من أشعل شمعة في دروب علمنا وإلى من وقف على المنابر وأعطى من حصيلة فكره
ينير دربنا إلى الأساتذة الكرام في معهدنا نتوجه بالشكر الجزيل إلى:

الأستاذة الدكتورة / ريهام عيسى

الذي تفضلت باشرافها على هذا البحث فجزاها الله عنا كل خير فلها منا كل التقدير والاحترام.

شكر وتقدير

لابد لنا ونحن نخطو خطواتنا الأخيرة في الحياة الجامعية من وقفة نعود إلى أعوام قضيناها في رحاب الجامعة مع أساتذتنا الكرام الذين قدموا لنا الكثير باذلين بذلك جهوداً كبيرة في بناء جيل الغد لتبعث الأمة من جديد ... وقبل أن نمضى نقدم أسمى آيات الشكر والامتنان والتقدير والمحبة إلى الذين حملوا أقدس رسالة في الحياة .. إلى الذين مهدوا لنا طريق العلم والمعرفة إلى جميع أساتذتنا الأفاضل ... "كن عالما .. فإن لم تستطع فكن متعلما ، فإن لم تستطع فأحب العلماء ، فإن لم تستطع فلا تبغضهم" .

ونخص بالشكر والتقدير إلى

الدكتورة / أميرة العطار

شكر وتقدير

لابد لنا ونحن نخطو خطواتنا الأخيرة في الحياة الجامعية من وقفة نعود إلى أعوام قضيناها في رحاب الجامعة مع أساتذتنا الكرام الذين قدموا لنا الكثير باذلين بذلك جهوداً كبيرة في بناء جيل الغد لتبعث الأمة من جديد ... وقبل أن نمضى نقدم أسمى آيات الشكر والامتنان والتقدير والمحبة إلى الذين حملوا أقدس رسالة في الحياة .. إلى الذين مهدوا لنا طريق العلم والمعرفة إلى جميع أساتذتنا الأفاضل ... "كن عالما .. فإن لم تستطع فكن متعلما ، فإن لم تستطع فأحب العلماء ، فإن لم تستطع فلا تبغضهم" .

ونخص بالشكر والتقدير إلى

الدكتورة / منى محمد موافي

الفهرس

الصفحة	الموضوع	م
	الفصل الأول	
10	مقدمة الدراسة	1
11	مشكلة الدراسة	2
11	أهداف الدراسة	3
12	أهمية الدراسة	4
12	حدود الدراسة	5
13	عينة الدراسة	6
14	مصطلحات الدراسة	7
	الفصل الثاني	
18	الذكاء الإصطناعي	8
23	تعلم الآلة	9
28	تحليل المشاعر	10
	الفصل الثالث	
33	التحليل	11
36	التصميم	12
38	التنفيذ	13
	الفصل الثالث	
136	الملخص	14
137	المصادر	15

الفصل الأول

مقدمة الدراسة

الفصل الأول

1. مقدمة الدراسة

2. مشكلة الدراسة

3. أهداف الدراسة

4. أهمية الدراسة

5. حدود الدراسة

6. عينة الدراسة

7. مصطلحات الدراسة

مقدمة الدراسة

لقد أصبح الذكاء الاصطناعي مصطلحاً شاملاً للتطبيقات التي تؤدي مهام مُعقدة كانت تتطلب في الماضي إدخالات بشرية مثل التواصل مع العملاء عبر الإنترن特 أو ممارسة لعبة الشطرنج. يستخدم هذا المصطلح غالباً بالتبادل مع مجالاته الفرعية، والتي تشمل التعلم الآلي (ML) والتعلم العميق. ومع ذلك، هناك اختلافات.. على سبيل المثال، يُركز التعلم الآلي على إنشاء أنظمة تتعلم أو تحسن من أدائها استناداً إلى البيانات التي تستهلكها. ومن المهم أن نلاحظ أنه على الرغم من أن كل سُبل التعلم الآلي ما هي إلا ذكاء اصطناعي، فإنه ليس كل ذكاء اصطناعي يُعد تعلمآ آلياً.

للحصول على القيمة الكاملة من الذكاء الاصطناعي، تقوم العديد من الشركات باستثمارات كبيرة في فرق علوم البيانات. إن علوم البيانات، التي تُعد مجالاً متعدد التخصصات يستخدم الأسلوب العلمية وأساليب أخرى لاستخلاص القيمة من البيانات، تجمع بين المهارات المستمدة من مجالات مثل الإحصاء وعلوم الكمبيوتر مع المعرفة العلمية لتحليل البيانات التي يتم جمعها من مصادر متعددة.

من أكثر الشركات نجاحاً هي الشركات التي تفهم عملائها جيداً وتفهم ما يقولونه وكيف يقولونه وما معناه. كلام العملاء يمكن أن نجد في reviews و comments و tweets وفي معظم المواقع الأخرى وفهمه مهم جداً. إن مجال ال sentiment analysis هو المجال الذي يهتم بدراسة كلام العملاء وتحليل معاني الكلام وهو مجال هام جداً وضروري تعلمه لكل من المبرمجين واصحاب الشركات. في هذا المشروع حاول مساعدة الشركات تسهيل استخدام مواقعهم للعملاء عن طريق تقييم تعليقاتهم تلقائياً حيث في باقي المواقع يقوم المستخدم بالتعليق ثم بالتقييم ولكن يمكن استخدام ال sentiment analysis بعد أن يقوم العميل بالتعليق وحساب التقييم تلقائياً اما بالايجاب او بالسلب.

مشكلة الدراسة

تقوم الدراسة الحالية بالعمل على:

- مساعدة أصحاب المنتجات معرفة ردود أفعال العملاء على منتجاتهم.

- تسهيل تقييم المنتج على العميل

- استخدام الذكاء الاصطناعي في تقييم التعليقات

- عمل python ML module و Laravel project integration

أهداف الدراسة

- بناء موقع يتمكن فيه ال administrator أن يقوم بإنشاء وإدارة المنتجات ويتمكن ال editor من إنشاء منتج ويتمكن ال viewer أو ال user أن يقوم بعمل review على المنتج.

- استخدام ال Machine Learning في جعل عملية تقييم المنتجات تلقائية بناءً على تعليق العميل.

- حساب متوسط التقييمات على المنتجات حتى يستطيع ال Administrator و ال Editor أن يتبع آراء العملاء على المنتجات الخاصة به.

- السماح للعملاء تصفح المنتجات والتعليق عليهم بحرية.

أهمية الدراسة

- التعليقات والتقييمات من أهم خصائص المواقع الإلكترونية ويجب على كل موقع معاصر يعرض منتجاته أن يسمح لعملائه بنظام سهل وسلس للتعليق والتقييم على المنتجات .
- باستخدام فرع مهم جداً من فروع ال Machine Learning وهو ما يسمى ب sentiment analysis يمكن جعل عملية التقييم تلقائية مما يزيد من جودة ال user experience او ما يعرف بال UX وأيضاً يسمح لأصحاب المنتجات متابعة منتجاتهم بطريقة أكثر فعالية وأكثر إنتاجية

حدود الدراسة

- بناء موقع إلكتروني لعرض المنتجات باستخدام التقنيات في ال front-end HTML, CSS, JavaScript, Jquery
- بناء لوحة تحكم تمكن أصحاب المنتجات من التحكم في المنتجات ومتابعة منتجاتهم باستخدام التقنيات في ال back-end PHP, MYSQL, Laravel .
- بناء نموذج تعلم الآلة باستخدام لغة Python .
- بناء API أو Application user interface باستخدام Flask واستخدامها في Laravel عن طريق استخدام HTTP protocol .

عينة الدراسة

Administrator .1

- a. إنشاء المستخدمين وحذفهم والتعديل عليهم
- b. الوصول للوحة التحكم
- c. إنشاء الأفلام وحذفها والتعديل عليها
- d. التقييم على الأفلام
- e. التغيير في محتوى الموقع
- f. إنشاء reviews على الأفلام وحذفها والتعديل عليها

Editor .2

- a. إنشاء الأفلام وحذفها والتعديل عليها
- b. التحكم في ال reviews على أفلامه
- c. عمل reviews على الأفلام
- d. الوصول لما يخصه فقط من لوحة التحكم

Viewer .3

- a. عمل reviews على الأفلام
- b. والتحكم في ال reviews الخاصة به

مصطلحات الدراسة

❖ **الخادم Server** : السيرفر هو عبارة عن جهاز حاسوب / كمبيوتر له القدرة على تحمل ظروف التشغيل باستمرار لفترات طويلة قد تصل لأكثر من عام، ويتم توصيل الخادم بالشبكة (أي كان نوعها أو حجمها - سواء شبكة داخلية أو شبكة الإنترن特)، ويتم تخصيص هذا الجهاز للقيام بمهام محددة لخدمة باقي الأجهزة المتصلة بنفس الشبكة، وطبقاً للمهام التي يختص بها جهاز الخادم / السيرفر يتم تحديد عتاد الجهاز ، فهناك مهام تحتاج لمعالجة الكثير من البيانات، وبالتالي يتم اختيار جهاز خادم يحتوي على أكثر من نواة - Core - في وحدة المعالجة المركزية، وهناك مهام تحتاج إلى الكثير من المساحات التخزينية، وبالتالي يتم استخدام خادم يحتوي على وحدات تخزين كبيرة، وهكذا يتم تحديد العتاد الخاص بجهاز الخادم، ومع التطور في تصنيع أجهزة الكمبيوتر، فإن أي جهاز حاسوب ذو عتاد جيد (مواصفات الجهاز - Hardware) يمكن أن يستخدم كخادم في أي شبكة.

❖ **الذكاء الاصطناعي Artificial intelligence** : الذكاء الاصطناعي هو فرع من علم الحاسوب. تُعرّف الكثير من المؤلفات الذكاء الاصطناعي، على أنه: «دراسة وتصميم العمليات الأذكياء»، والعميل الذكي هو نظام يستوعب بيئته ويتخذ المواقف التي تزيد من فرصته في النجاح في تحقيق مهمته أو مهمة فريقه. هذا التعريف، من حيث الأهداف والأفعال والتصور والبيئة يرجع إلى (Russell & Norvig 2003) وتشمل أيضاً التعريفات الأخرى المعرفة والتعلم معايير إضافية. صاغ عالم الحاسوب جون مكارثي هذا المصطلح بالأساس في عام 1956 وعرفه بنفسه بأنه «علم و الهندسة صناع الآلات الذكية». ويعرف أنديرياس كابلان ومايكل هاينلين الذكاء الاصطناعي بأنه «قدرة النظام على تفسير البيانات الخارجية بشكل صحيح، والتعلم من هذه البيانات، واستخدام تلك المعرفة لتحقيق أهداف ومهام محددة من خلال التكيف المرن».

❖ **تعلم الآلة Machine Learning** : هو أحد فروع الذكاء الاصطناعي التي تهتم بتصميم وتطوير خوارزميات وتقنيات تسمح للحواسيب بامتلاك خاصية «التعلم». بشكل عام هناك مستويين من التعلم: الاستقرائي والاستنتاجي. يقوم الاستقرائي باستنتاج قواعد وأحكام عامة من البيانات الضخمة. المهمة الأساسية للتعلم الآلي هو استخراج معلومات قيمة من البيانات، وبالتالي هو قريب جداً من التقىب في البيانات (بالإنجليزية: data mining) والإحصاء والمعلوماتية النظرية. يستخدم التعلم الآلي في العديد من المجالات من الهندسة إلى الطب.

❖ **تحليل المشاعر Sentiment Analysis** : هو استخدام معالجة اللغات الطبيعية، وعلم اللغة الحاسوبي والتحليل النصي من أجل الكشف عما يحمله النص من مشاعر سواء إيجابية أو سلبية أو محايضة تجاه موضوع النص. يُستخدم تحليل المشاعر على نطاق في مجالات التسويق وخدمة العملاء ومجالات أخرى. يهدف تحليل المشاعر بشكل عام إلى تحديد مشاعر المتحدث أو الكاتب تجاه موضوع ما أو تحديد المشاعر الغالبة على وثيقة ما. يمكن أن تعبّر هذه المشاعر عن رأي المؤلف (انظر نظرية التقييم النصي) أو حالة شعورية (أي الحالة الوجدانية المؤلف أثناء الكتابة) أو شعور مُتعمد توصيله (أي التأثير الوجداني الذي يريد المؤلف أن يُوقعه بنفس القارئ).

الفصل الثاني

الإطار النظري

الفصل الثاني

الإطار النظري

1. الذكاء الاصطناعي

2. تعلم الآلة

3. تحليل المشاعر

الذكاء الاصطناعي

تعريفه : يعرف الذكاء الاصطناعي على أنه تقنية تحاكي الذكاء البشري على أداء المهام ويمكنه بشكل مباشر متكرر تحسين نفسه استنادا إلى المعلومات التي يجمعها.

أهمية :

- يؤثر الذكاء الاصطناعي على مستقبل كل قطاع صناعي وعلى كل إنسان على هذا الكوكب كما ويدعى المحرك الأساسي لجميع التقنيات الناشئة مثل جمع البيانات الضخمة والروبوتات وإنترنت الأشياء، ومن المتوقع أن يلعب الذكاء الاصطناعي دوراً أكبر خلال السنوات القادمة.
- وبفضل الذكاء الاصطناعي يتطور مجال الرعاية الصحية بسرعة متزايدة ويرافق ذلك زيادة كبيرة في كمية البيانات والتحديات في ما يخص التكلفة ونتائج المرضى لذلك تم استخدام تطبيقات الذكاء الاصطناعي للحد من هذه الصعوبات، كما ويتم استخدام الذكاء الصناعي لتفادي إجراء الفحوصات المخبرية الروتينية غير الضرورية، وتضييق دائرة التحاليل المخبرية التي قد يحتاجها المريض، وتحسين سير العمل السريري، والتنبؤ بالأمراض المكتسبة من المستشفيات.
- كما يزيد الذكاء الاصطناعي من كفاءة الأعمال وسرعة تنفيذها ويزيد من قيمتها ويساهم في تطور الأعمال باستمرار، كما يزيد من عدد المتفاعلين مع هذه الأعمال بسبب التطور المستمر للأدوات والبرمجيات المتعلقة بها.
- والذكاء الاصطناعي أهمية في حياتنا اليومية فقد أحدث استخدام تطبيقات الذكاء الاصطناعي ثورة كبيرة في مجال صناعة السيارات حيث يستخدم برنامج القيادة الذاتية من جوجل الذي يستخدم تقنيات الذكاء الاصطناعي لتقليل نسبة الحوادث وتحفيظ الازدحام المروري، وتستخدم تطبيقات الذكاء الاصطناعي في موقع التجارة الإلكترونية للحصول على صورة واضحة لسلوك العملاء في عمليات الشراء عبر الموقع وتقديم التوصيات، وتستخدم أيضاً شبكات التواصل الاجتماعي تطبيقات الذكاء

الاصطناعي مثل فيسبوك للكشف عن وجود اختراق لصور المستخدم.

نشأته : إن أول عمل جوهري في مجال الذكاء الاصطناعي قام به عالم الرياضيات ورائد الحاسوب البريطاني آلان تورينج، حيث أعلن تورينج في عام 1950 أنه في يوم من الأيام سيكون هناك آلية يمكنها معاشرة الذكاء البشري بكل طريقة وإثبات ذلك من خلال اجتياز اختبار متخصص، وفي هذا الاختبار سيتمكن طرح أسئلة متطابقة عشوائية على جهاز حاسوب وإنسان مخفي عن الأنظار، وإذا نجح الحاسوب فلن يتمكن السائل من تمييز الآلة عن الشخص بالإجابات. وبحلول أوائل القرن الحادي والعشرين، لم يقترب أي برنامج للذكاء الاصطناعي من اجتياز اختبار تورينج، ومع ذلك فقد حققت بعض البرامج مستويات أداء الخبراء البشريين في أداء بعض المهام المحددة، ويمكن العثور على الذكاء الاصطناعي بهذا المعنى المحدود في تطبيقات متنوعة مثل التشخيص الطبي وترجمة اللغات وتصميم الحاسوب والتعرف على الصوت أو الكتابة اليدوية.

ونظراً لأن الهدف النهائي للذكاء الاصطناعي هو إنشاء أجهزة حاسوب يمكنها التفكير كما يفعل البشر فقد اقترح بعض مؤيدي الذكاء الاصطناعي أنه يجب تصميم أجهزة الحاسوب على غرار الدماغ البشري والذي يتكون أساساً من شبكة من الخلايا العصبية، وتم تطوير أول شبكة عصبية اصطناعية في عام 1954 عندما كان هدف الذكاء الاصطناعي القوي نظام يقترب من الذكاء البشري يتقاسمها الكثيرون، وفي أوائل العقد الأول من القرن الحادي والعشرين كانت الشبكات العصبية الاصطناعية قادرة على مجموعة من المهام المعقّدة بما في ذلك التعرف على الوجوه والأشياء الأخرى من البيانات المرئية لكن التفاؤل بشأن تحقيق ذكاء اصطناعي قوي أفسح المجال لنقدير الصعوبات الشديدة التي ينطوي عليها الأمر.

وقد أكد بعض باحثي الذكاء الاصطناعي أن الذكاء الحقيقي ينطوي ببساطة على القدرة على العمل في بيئه حقيقة، فكان هذا النهج المعروف باسم الذكاء الاصطناعي الجديد رائداً في مختبر الذكاء الاصطناعي التابع لمعهد ماساتشوستس للتكنولوجيا بواسطة العالم الأسترالي رودني بروكس، وأحد الأمثلة الشهيرة للذكاء الاصطناعي الجديد هو الروبوت المحمول هيربيرت من بروكس والذي صُمم للتجول في مساحة مكتبية وجمع علب الصودا الفارغة والتخلص منها، ومنذ أن تم الكشف عن هيربرت في أواخر الثمانينيات صمم

بروكس وطلابه روبوتات أخرى لتطهير حقول الألغام واستكشاف المريخ بالإضافة إلى روبوت بشري اسمه Cog والذي تزيد معرفته بشكل متزايد من خلال تفاعلاته مع البيئة.

تطوره :

الحيوانات الآلية ذات الذكاء الاصطناعي: على الرغم من أن الحيوانات الآلية الحقيقة جميلة، ولكن وفقاً لمطوري برامج الحيوانات الآلية فلها بعض المشكلات، حيث يجب الاعتناء بها ورعايتها وإطعامها وتنظيمها. ستكون الحيوانات الآلية التي تعمل بالذكاء الاصطناعي روبوتات تبدو وتشعر وتتصرف كحيوان حقيقي ولكنها تقضي على مثل هذه المشكلات التي يواجهها أصحابها. ومن المتوقع أن تكون الحيوانات الآلية التي يقودها الذكاء الاصطناعي متاحة على نطاق واسع في السوق بحلول عام 2025. يتعرف الذكاء الاصطناعي على المشاعر: يمكن للروبوت الذي تم بناؤه في أواخر التسعينيات ويسمى Kismet التعرف على المشاعر من خلال لغة الجسد البشري ونبرة الصوت.

يمكن للذكاء الاصطناعي إصلاح نفسه: أعاد روبوت بناء نفسه بعد أن لاحظ أن أدائه قد انخفض بعد أن فقد اثنين من أرجله السنتين. لم يعرف الروبوت ما هي المشكلة ولكنه أصلحها عن طريق التجربة والخطأ. يمكن للذكاء الاصطناعي أن يحل محل القوى العاملة البشرية: حوالي 1,160,000 شخص عاطلون عن العمل في كندا وحدها. على الرغم من أن الذكاء الاصطناعي يساعد في تقليل تكاليف الأعمال، إلا أنه من المقرر أن يوجد بعض المشكلات المهمة. فوفقاً لصحيفة The Guardian، ستواجه (85٪) من وظائف خدمة العملاء أعلى تهديداً لها بسبب الذكاء الاصطناعي بحلول عام 2022.

قد الذكاء الاصطناعي أكثر ذكاءً من البشر: يمكن للذكاء الاصطناعي تعلم أي شيء بسرعة، مما يعني أن ذكاءه آخذ في الازدياد. في عام 2013، كان لدى الذكاء الاصطناعي نفس ذكاء طفل يبلغ من العمر 4 سنوات. بحلول عام 2029، سيكون الذكاء الاصطناعي نفس مستوى ذكاء البشر البالغين. يمكن للبشر تطوير علاقة شاعرية مع الذكاء الاصطناعي: يعتقد ديفيد ليفي وهو طالب في هذا المجال أن الزواج بين البشر والروبوتات سيصبح قانونياً بحلول عام 2050. في الوقت الحالي، الروبوتات ليست متقدمة بما يكفي لهذا الغرض. للذكاء الاصطناعي جنسيات وجوازات سفر: حصلت صوفيا وهي إنسان آلي شبيه بالبشر على

الجنسية المضمونة في المملكة العربية السعودية. لقد أثار الجدل حيث تسأَل الناس عما إذا كان يجب أن تتمتع الروبوتات بحقوق أم لا.

الذكاء الاصطناعي يمكنه الكتابة بمفرده: كتب برنامج آلي مقالاً عن زلزال في كاليفورنيا على موقع لوس أنجلوس تايمز على الإنترنت، وجمع البيانات من جهاز قياس الزلازل. يُعرف الذكاء الاصطناعي على الناس عن طريق الصوت: تستطيع برامج الذكاء الاصطناعي التعرف على الصوت وتمييز المتحدث من بينآلاف أو ملايين البشر بسبب ما يُسمى "البصمة الصوتية".

أهدافه :

- توفر إمكانية الحوسبة عالية الأداء بسهولة وبأسعار معقولة. إن وفرة قدرة الحوسبة في مجال الأعمال في السحابة، مكن من الوصول السهل للقدرة على الحوسبة بأداء عالي وبأسعار معقولة. وقبل هذا التطور، كانت بيئات الحوسبة الوحيدة المتاحة للذكاء الاصطناعي غير قائمة على السحابة وتحتاج إلى تكاليف باهضة.
- وجود كميات كبيرة من البيانات المتاحة للتعلم. يحتاج الذكاء الاصطناعي إلى التعلم من خلال الكثير من البيانات لإجراء التنبؤات الصحيحة. وقد أدى ظهور أدوات مختلفة لجمع البيانات المصنفة، بالإضافة إلى تمكن المؤسسات من تخزين هذه البيانات ومعالجتها بسهولة وبتكلفة ميسورة سواء البيانات الهيكيلية أو غير الهيكيلية، إلى تمكن المزيد من المؤسسات من إنشاء خوارزميات الذكاء الاصطناعي وتدربيها.
- توفر تقنية الذكاء الاصطناعي التطبيقي ميزة تنافسية. تدرك الشركات بشكل متزايد الميزة التنافسية لتطبيق رؤى الذكاء الاصطناعي على أهداف الأعمال وجعلها أولوية على مستوى الأعمال. على سبيل المثال، يمكن أن تساعد التوصيات المستهدفة التي تقدمها تقنية الذكاء الاصطناعي على اتخاذ قرارات أفضل بشكل أسرع. كما يمكن للعديد من ميزات وقدرات الذكاء الاصطناعي أن تؤدي إلى خفض التكاليف وتقليل المخاطر وتسريع وقت الوصول إلى السوق وغير ذلك الكثير.

خصائص الذكاء الاصطناعي :

- استخدام الذكاء الاصطناعي لحل المشاكل المعروضة في غياب المعلومة الكاملة.
- القدرة على التفكير والإدراك.
- القدرة على اكتشاف المعرفة وتطبيقاتها.
- القدرة على التعلم والفهم من التجارب والخبرات السابقة.
- القدرة على استخدام الخبرات القديمة وتوظيفها في مواقف جديدة.
- القدرة على استخدام التجربة والخطأ لاستكشاف الأمور المختلف عليها.
- القدرة على الاستجابة السريعة للمواقف والظروف الجديدة.
- القدرة على التعامل مع الحالات الصعبة والمعقدة.
- القدرة على التطور والإبداع وفهم الأمور المرئية وإدراكتها.
- القدرة على تقديم المعلومة لإسناد القرارات الإدارية.

تعلم الآلة

تعريفه : التعلم الآلي هو علم تطوير الخوارزميات والنمذج الإحصائية التي تستخدمها أنظمة الكمبيوتر لأداء المهام بدون تعليمات واضحة، اعتماداً على الأنماط والاستدلال بدلاً من ذلك. وتستخدم أنظمة الكمبيوتر لوغاريمات التعلم الآلي لمعالجة كميات كبيرة من البيانات السابقة والتعرف على أنماط البيانات. وهذا يسمح لها بتوقع النتائج بصورة أكثر دقة من مجموعة بيانات مدخلة معينة. على سبيل المثال، يمكن لعلماء البيانات تدريب تطبيق طبي على تشخيص مرض السرطان من صور الأشعة السينية عن طريق تخزين ملايين الصور المفحوصة والتشخيصات المقابلة.

أهدافه :

- **التصنيع :** يستطيع التعلم الآلي دعم الصيانة التنبؤية ومراقبة الجودة والأبحاث الابتكارية في قطاع التصنيع. وتساعد تكنولوجيا التعلم الآلي أيضاً الشركات في تحسين الحلول اللوجستية، بما في ذلك الأصول وسلسلة الإمداد وإدارة المخزون. على سبيل المثال، عملاق التصنيع 3M تستخدم AWS Machine Learning لابتكار ورق الصنفراة. تعمل لوغاريمات التعلم الآلي على تمكين الباحثين في 3M من تحليل كيف يمكن للتغيرات الطفيفة في الشكل والحجم والاتجاه أن تحسن إمكانية الكشط والاستدامة. وتغذى تلك الاقتراحات عملية التصنيع.
- **الرعاية الصحية وعلوم الحياة :** إن انتشار المستشعرات والأجهزة التي يمكن ارتداؤها قد أنشأ كمًا هائلًا من البيانات الصحية. وستطيع برامج التعلم الآلي تحليل هذه المعلومات ودعم الأطباء في التشخيص والعلاج في الوقت الفعلي. يطور باحثو التعلم الآلي الحلول التي تكتشف الأورام السرطانية وتشخيص أمراض العين، ما يؤثر بشكل بالغ في نتائج الصحة البشرية. على سبيل المثال، قامت AWS Machine Learning باستخدام Cambia Health Solutions لدعم الشركات

الناشرة في قطاع الرعاية الصحية حيث تستطيع أتمتة علاج النساء الحوامل وتخصيصه.

- **الخدمات المالية :** تحسن مشروعات التعلم الآلي في القطاع المالي تحليلات المخاطر والتنظيم.

وتحسن تكنولوجيا التعلم الآلي السماح للمستثمرين بالتعرف على الفرص الجديدة عن طريق تحليل حركات سوق الأسهم أو تقييم صناديق التحوط أو معايرة المحافظ المالية. بالإضافة إلى ذلك، يمكن للتعلم الآلي المساعدة في التعرف على عملاء القروض عالي المخاطر والتخفيف من علامات الاحتيال. تستخدم رائدة البرمجيات المالية Intuit نظام AWS Machine Learning، Amazon Textract لإنشاء إدارة مالية أكثر تخصيصاً ومساعدة المستخدمين النهائيين في تحسين وضعهم المالي.

- **البيع بالتجزئة :** يمكن استخدام التعلم الآلي في قطاع تجارة التجزئة من أجل تحسين خدمة العملاء

وإدارة الأسهم وزيادة المبيعات والتسويق متعدد القنوات. على سبيل المثال، خفضت Amazon (AFT Fulfillment) تكاليف بنيتها الأساسية بمعدل 40% باستخدام نموذج تعلم آلة الغرض منه تحديد المخزون الذي يكون في غير موضعه. ويساعدها ذلك في الوفاء بوعده Amazon المتمثل في سرعة توفر العنصر للعملاء ووصوله في الموعد المناسب، وذلك على الرغم من معالجة ملايين الشحنات العالمية سنوياً.

- **الوسائل الإعلامية والترفيهية :** تتحول الشركات الترفيهية إلى التعلم الآلي لتحسين فهمها لجماهيرها

المستهدفة وتقديم محتوى جذاب ومخصص عند الطلب. وتنشر لوغاریتمات التعلم الآلي لمساعدة في تصميم إعلانات الأعمال الفنية والإعلانات الأخرى، لتزويد العملاء بتوصيات مخصصة للمحتوى، وتبسيط عملية الإنتاج.

نشارته :

أُستخدم مصطلح «التعلم الآلي» لأول مرة في عام 1959 من قبل آرثر صموئيل، وهو أمريكي عمل في شركة آي بي إم. في السبعينيات، كان التعلم الآلي يستخدم في الغالب تصنيف الأنماط.

اهميته :

يساعد التعلم الآلي الأعمال عن طريق دفع عجلة النمو وفتح سبل إيرادات جديدة وحل مشكلات صعبة. البيانات هي القوة المحركة الحاسمة وراء صنع القرار التجاري، ولكن في الأحوال التقليدية، كانت تستخدم الشركات البيانات من مصادر مختلفة، مثل ملاحظات العملاء والموظفين والوضع المالي. وتعمل أبحاث التعلم الآلي على أتمتة هذه العملية وتحسينها. عن طريق استخدام البرمجيات التي تحل كميات هائلة من البيانات بسرعات عالية، تستطيع الأعمال تحقيق النتائج بصورة أسرع.

خصائصه :

- القدرة على أداء تصور البيانات الآلي: يتم إنشاء كمية هائلة من البيانات من قبل الشركات وعامة الناس على أساس منتظم. من خلال تصور العلاقات البارزة في البيانات ، لا يمكن للشركات اتخاذ قرارات أفضل فحسب ، بل يمكنها أيضًا بناء الثقة. يقدم التعلم الآلي عدًّا من الأدوات التي توفر مقتطفات منسقة من البيانات التي يمكن تطبيقها على كل من البيانات المنظمة وغير المنظمة. بمساعدة منصات تصور البيانات المؤتمتة سهلة الاستخدام في التعلم الآلي ، يمكن للشركات الحصول على ثروة من الرؤى الجديدة في محاولة لزيادة الإنتاجية في عملياتها.
- أتمتة في أفضل حالاتها: تمثل إحدى أكبر خصائص التعلم الآلي في قدرته على أتمتة المهام المتكررة وبالتالي زيادة الإنتاجية. يستخدم عدد كبير من المؤسسات بالفعل الأعمال الورقية التي تعتمد على التعلم الآلي وأتمتة البريد الإلكتروني. في القطاع المالي ، على سبيل المثال ، هناك حاجة إلى عدد كبير من المهام المتكررة والمتعلقة بالبيانات والتي يمكن التنبؤ بها. لهذا السبب ، يستخدم هذا القطاع أنواعًا مختلفة من حلول التعلم الآلي إلى حد كبير. يجعل مهام المحاسبة أسرع وأكثر ثانية وأكثر دقة. تتضمن بعض الجوانب التي تمت معالجتها بالفعل من خلال التعلم الآلي معالجة

الاستفسارات المالية بمساعدة روبوتات المحادثة ، والتنبؤ ، وإدارة النفقات ، وتبسيط الفواتير ، وأتمتة التسويات المصرفية.

• القدرة على الارتقاء بالكفاءة: إلى المستوى التالي عند الدمج مع إنترنت الأشياء كما لم يحدث من قبل. بفضل الضجيج الهائل المحيط بإنترنت الأشياء ، شهد التعلم الآلي ارتفاعاً كبيراً في شعبيته. تم تصنيف إنترنت الأشياء منطقة ذات أهمية استراتيجية من قبل العديد من الشركات. وقد أطلق العديد من الآخرين مشاريع تجريبية لقياس إمكانات إنترنت الأشياء في سياق العمليات التجارية. لكن الحصول على مزايا مالية من خلال إنترنت الأشياء ليس بالأمر السهل. من أجل تحقيق النجاح ، تحتاج الشركات ، التي تقدم خدمات وخدمات استشارية لإنترنت الأشياء ، إلى تحديد المجالات التي ستتغير بوضوح مع تنفيذ استراتيجيات إنترنت الأشياء. فشلت العديد من هذه الشركات في معالجتها. في هذا السيناريو ، ربما يكون التعلم الآلي هو أفضل تقنية يمكن استخدامها لتحقيق مستويات أعلى من الكفاءة. من خلال دمج التعلم الآلي مع إنترنت الأشياء ، يمكن للشركات تعزيز كفاءة عمليات الإنتاج بأكملها.

• القدرة على تغيير سوق الرهن العقاري : إنها حقيقة مفادها أن تعزيز درجة ائتمان إيجابية يتطلب عادةً الانضباط والوقت والكثير من التخطيط المالي لكثير من المستهلكين. عندما يتعلق الأمر بالمقرضين ، فإن درجة ائتمان المستهلك هي واحدة من أكبر مقاييس الجدارة الائتمانية التي تتخطى على عدد من العوامل بما في ذلك تاريخ الدفع ، وإجمالي الديون ، وطول تاريخ الائتمان وما إلى ذلك ، ولكن أليس من الرائع أن يكون هناك مقياس مبسط وقياس أفضل؟ بمساعدة التعلم الآلي ، يمكن للمقرضين الآن الحصول على صورة أكثر شمولاً للمستهلك. يمكنهم الآن التنبؤ بما إذا كان العميل منفقاً منخفضاً أو منفقاً مرتفعاً وفهم نقطة تحول في الإنفاق. بصرف النظر عن الإقراض العقاري ، تستخدم المؤسسات المالية نفس الأساليب لأنواع أخرى من القروض الاستهلاكية.

• تحليل دقيق للبيانات: تقليدياً ، كان تحليل البيانات يشمل دائماً طريقة التجربة والخطأ ، وهو نهج يصبح مستحيلاً عندما نعمل مع مجموعات بيانات كبيرة وغير متجانسة. يأتي التعلم الآلي كأفضل حل لجميع هذه المشكلات من خلال تقديم بدائل فعالة لتحليل كميات هائلة من البيانات. من خلال تطوير خوارزميات فعالة وسريعة ، بالإضافة إلى نماذج تعتمد على البيانات معالجة البيانات في الوقت الفعلي ، يمكن للتعلم الآلي إنشاء تحليل ونتائج دقيقة.

- ذكاء الأعمال في أفضل حالاتها : يمكن لخصائص التعلم الآلي ، عند دمجها مع العمل التحليلي للبيانات الضخمة ، أن تولد مستويات قصوى من ذكاء الأعمال بمساعدة العديد من الصناعات المختلفة التي تتخذ مبادرات إستراتيجية. من البيع بالتجزئة إلى الخدمات المالية إلى الرعاية الصحية وغير ذلك الكثير – أصبح التعلم الآلي بالفعل أحد أكثر التقنيات فعالية لتعزيز العمليات التجارية. سواء كنت مقتنعاً أم لا ، فإن خصائص تعلم الآلة المذكورة أعلاه قد ساهمت بشكل كبير في جعله أحد أهم اتجاهات التكنولوجيا – فهو يكمن وراء عدد كبير من الأشياء التي نستخدمها هذه الأيام دون حتى التفكير فيها.
- مشاركة العملاء كما لم يحدث من قبل : بالنسبة لأي عمل تجاري ، تتمثل إحدى الطرق الأكثر أهمية لدفع المشاركة وتعزيز ولاء العلامة التجارية وإقامة علاقات طويلة الأمد مع العملاء في إجراء محادثات هادفة مع قاعدة العملاء المستهدفة. يلعب التعلم الآلي دوراً مهماً في تمكين الشركات والعلامات التجارية من إجراء محادثات أكثر قيمة فيما يتعلق بمشاركة العملاء. حل التكنولوجيا عبارات وكلمات وجمل وتعابير وتنسيقات محتوى معينة لها صدى لدى أفراد معينين من الجمهور. يمكنك التفكير في بيانتس التي تستخدم التعلم الآلي بنجاح لتخصيص الاقتراحات لمستخدميها. إنه يستخدم التكنولوجيا لمصدر المحتوى الذي سيكون المستخدمون مهتمين به ، بناءً على الكائنات التي قاموا بتثبيتها بالفعل.

Sentiment analysis

: تعريفه

تحليل المشاعر ينطوي على عملية تحليل النص الرقمي لتحديد ما إذا كانت النغمة العاطفية للرسالة إيجابية أو سلبيةً أو محايدةً. تمتلك اليوم الشركات كميات كبيرة من البيانات النصية، مثل رسائل البريد الإلكتروني، ونصوص الدردشة دعم العملاء، تعليقات وسائل التواصل الاجتماعي، والمراجعات. يمكن لأدوات تحليل المشاعر مسح هذه النصوص ضوئياً لتمكن تلقائياً من تحديد موقف كاتبها من موضوع ما. تستخدم الشركات رؤى تحليل المشاعر لتحسين خدمة العملاء وتعزيز سمعة العلامة التجارية.

: أهميته

يعد تحليل المشاعر، المعروف أيضاً باسم التقييم عن الآراء، أداة مهمة لذكاء الأعمال تساعد الشركات في تحسين منتجاتها وخدماتها. نقدم أدناه بعض مزايا تحليل المشاعر.

- **تقديم رؤى موضوعية :** يمكن للشركات تجنب التحيز الشخصي المرتبط بالمقيمين من البشر باستخدام أدوات تحليل المشاعر القائمة على الذكاء الاصطناعي (AI). ونتيجةً لذلك، تحصل الشركات على نتائج متسقة وموضوعية عند تحليل آراء العملاء. على سبيل المثال، تأمل الجملة التالية: أنا مندهش من سرعة المعالج ولكنني أشعر بالإحباط لأنه يسخن بسرعة. قد يرفض المسوقون

الجزء المحبط من التقييم و يتيحون بشكل إيجابي تجاه أداء المعالج. ومع ذلك، تقوم أدوات تحليل المشاعر الدقيقة بفرز النص وتصنيف لاستخلاص المشاعر بموضوعية.

- **بناء منتجات وخدمات أفضل :** نظام تحليل المشاعر يساعد الشركات في تحسين منتجاتها وخدماتها بناءً على ملاحظات العملاء الحقيقة والمحددة. تقنيات الذكاء الاصطناعي تحدد الأشياء أو المواقف الواقعية (تسمى الكيانات) التي يربطها العملاء بمشاعر سلبية. من المثال أعلاه، يركز مهندسو المنتجات على تحسين إمكانية إدارة حرارة المعالج لأن برنامج تحليل النص المرتبط بمحبطة (سلبية) مع المعالج (الكيان) ويُسخن (الكيان).
- **تحليل على نطاق واسع :** تقوم الشركات باستمرار بالتنقيب عن المعلومات من كمية هائلة من البيانات غير المهيكلة، مثل رسائل البريد الإلكتروني، ونصوص برامج الدردشة الآلية، والاستبيانات، وسجلات إدارة علاقات العملاء، والملاحظات حول المنتجات. تسمح أدوات تحليل المشاعر المستندة إلى السحابة للشركات بتوسيعة نطاق عملية الكشف عن مشاعر العملاء في البيانات النصية بتكلفة معقولة.
- **النتائج في الوقت الفعلي :** يجب أن تكون الشركات سريعة الاستجابة للأزمات المحتملة أو اتجاهات السوق في المشهد سريع التغير اليوم. يعتمد المسوقون على برمجيات تحليل المشاعر لمعرفة ما يشعر به العملاء تجاه العلامة التجارية للشركة ومنتجاتها وخدماتها في الوقت الفعلي واتخاذ إجراءات فورية بناءً على النتائج التي توصلوا إليها. يمكنهم تكوين البرمجيات لإرسال تنبيهات عند اكتشاف المشاعر السلبية من كلمات أساسية محددة.
- **حالات الإستخدام :** تستخدم الشركات تحليل المشاعر في استخلاص خطط قابلة للتنفيذ حول الذكاء والنمذاج في مجالات مختلفة.
- **تحسين خدمة العملاء :** تستخدم الشركات تحليل المشاعر في استخلاص خطط قابلة للتنفيذ حول الذكاء والنمذاج في مجالات مختلفة.
- **تحسين خدمة العملاء :** تستخدم فرق دعم العملاء أدوات تحليل المشاعر لتخصيص الردود بناءً على الحالة المزاجية للمحادثة. يتم رصد الأمور العاجلة بواسطة برامج الدردشة الآلية القائمة على الذكاء الاصطناعي (AI) مع القدرة على تحليل المشاعر وتصعيدها إلى موظفي الدعم.

- **مراقبة العلامة التجارية :** تراقب المنظمات باستمرار الإشارات والثرة التي تحدث حول علاماتها التجارية على وسائل التواصل الاجتماعي والمنتديات والمدونات والمقالات الإخبارية وغيرها من المساحات الرقمية. تتيح تقنيات تحليل المشاعر لفريق العلاقات العامة أن يكون على دراية بالقصص الجارية ذات الصلة. يمكن للفريق تقييم الحالة المزاجية الأساسية لمعالجة الشكاوى أو الاستفادة من الاتجاهات الإيجابية.
- **أبحاث السوق :** يساعد نظام تحليل المشاعر الشركات في تحسين عرض منتجاتها من خلال معرفة ما يصلح وما لا يصلح. يمكن للمسوقين تحليل التعليقات على موقع التقييمات عبر الإنترن特، والاستجابات للاستبيانات، ونشرات وسائل التواصل الاجتماعي لاكتساب رؤى أعمق حول مميزات منتج محددة. إنهم ينقلون النتائج إلى مهندسي المنتجات الذين يتذكرون وفقاً لذلك.
- **تتبع أداء الحملة :** يستخدم المسوقون أدوات تحليل المشاعر للتأكد من أن حملتهم الإعلانية تحقق الاستجابة المتوقعة. يقومون بتتبع المحادثات على منصات التواصل الاجتماعي ويتذكرون من أن المشاعر العامة مشجعة. إذا كان ملخص المشاعر أقل من التوقعات، يقوم المسوقون بتعديل الحملة بناءً على تحليلات البيانات في الوقت الفعلي.

الفصل الثالث

إجراءات البحث

الفصل الثالث

إجراءات البحث

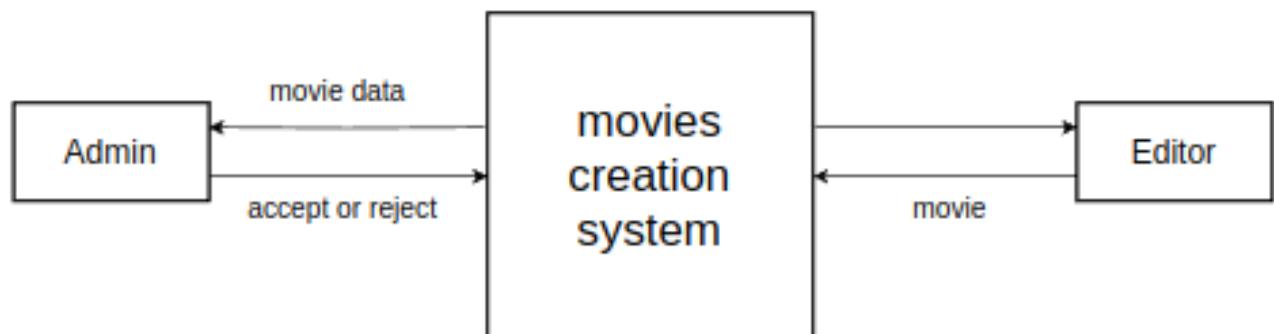
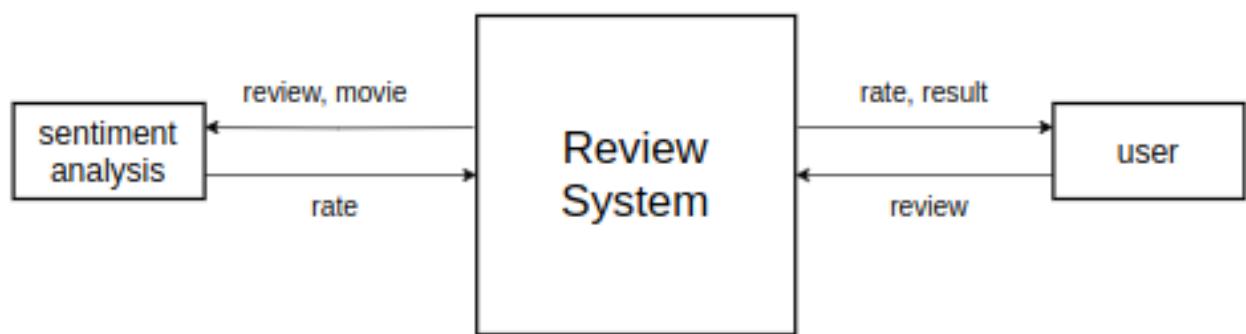
1. التحليل

2. التصميم

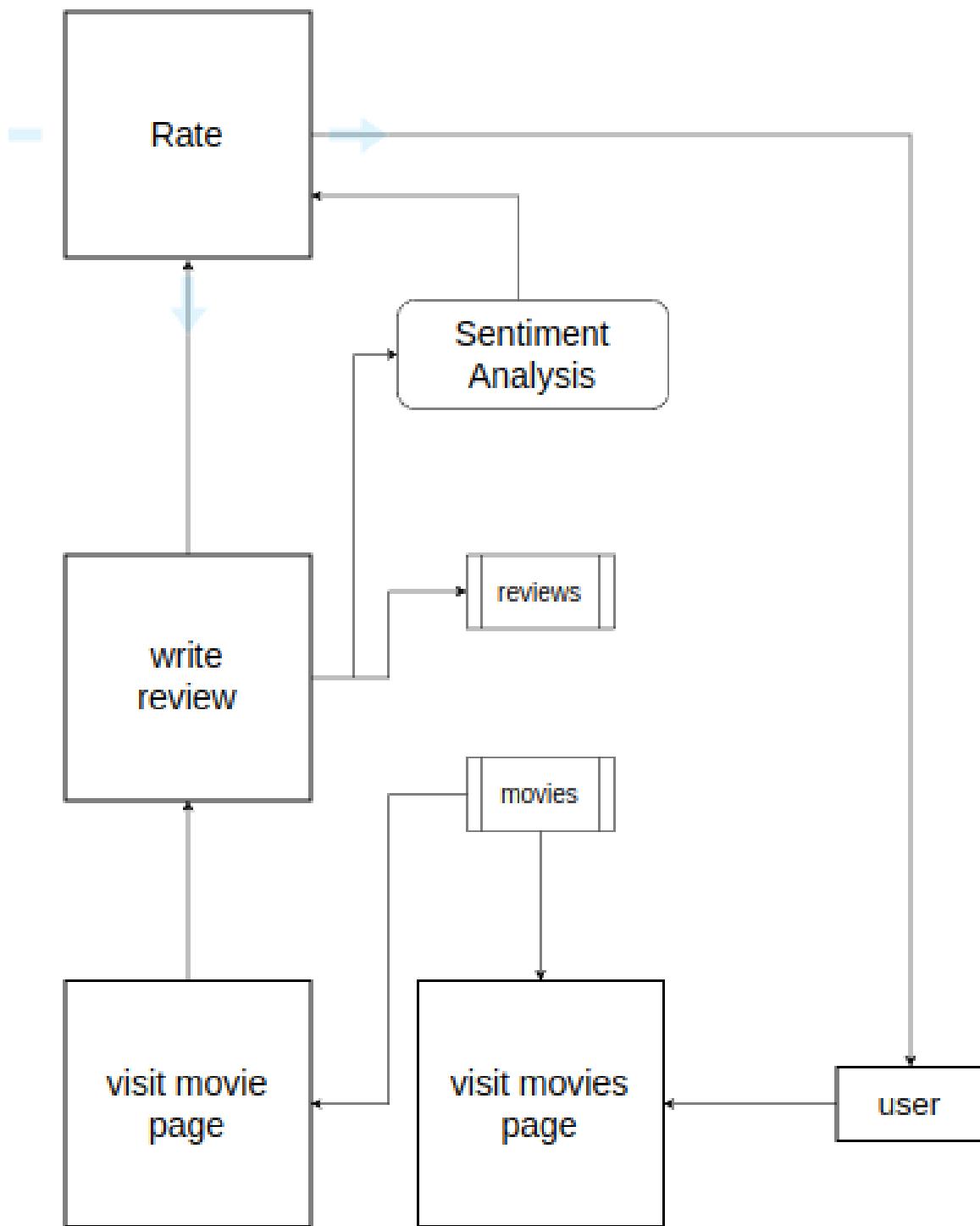
3. التنفيذ

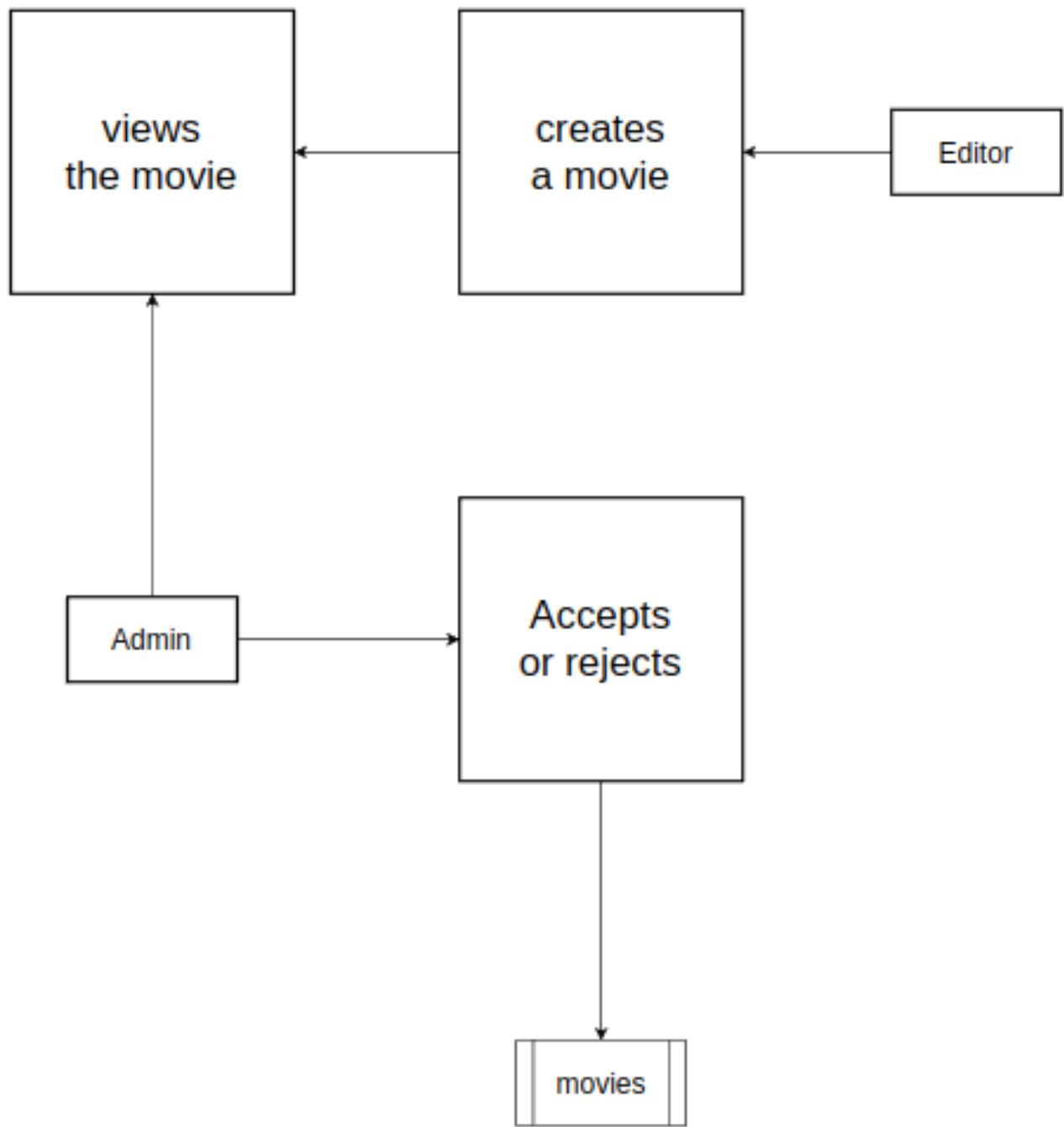
تحليل النظام

Context diagram level 0



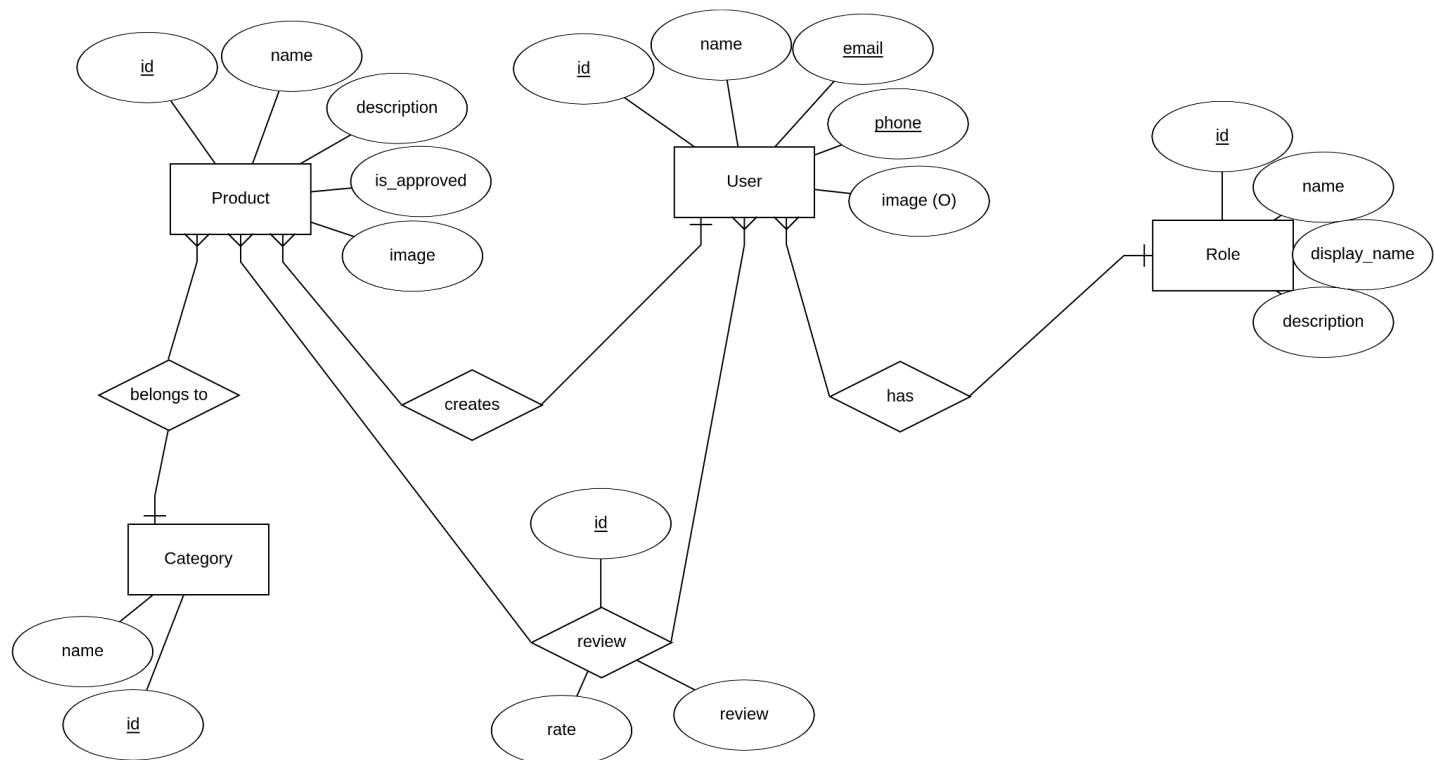
Context diagram level 1



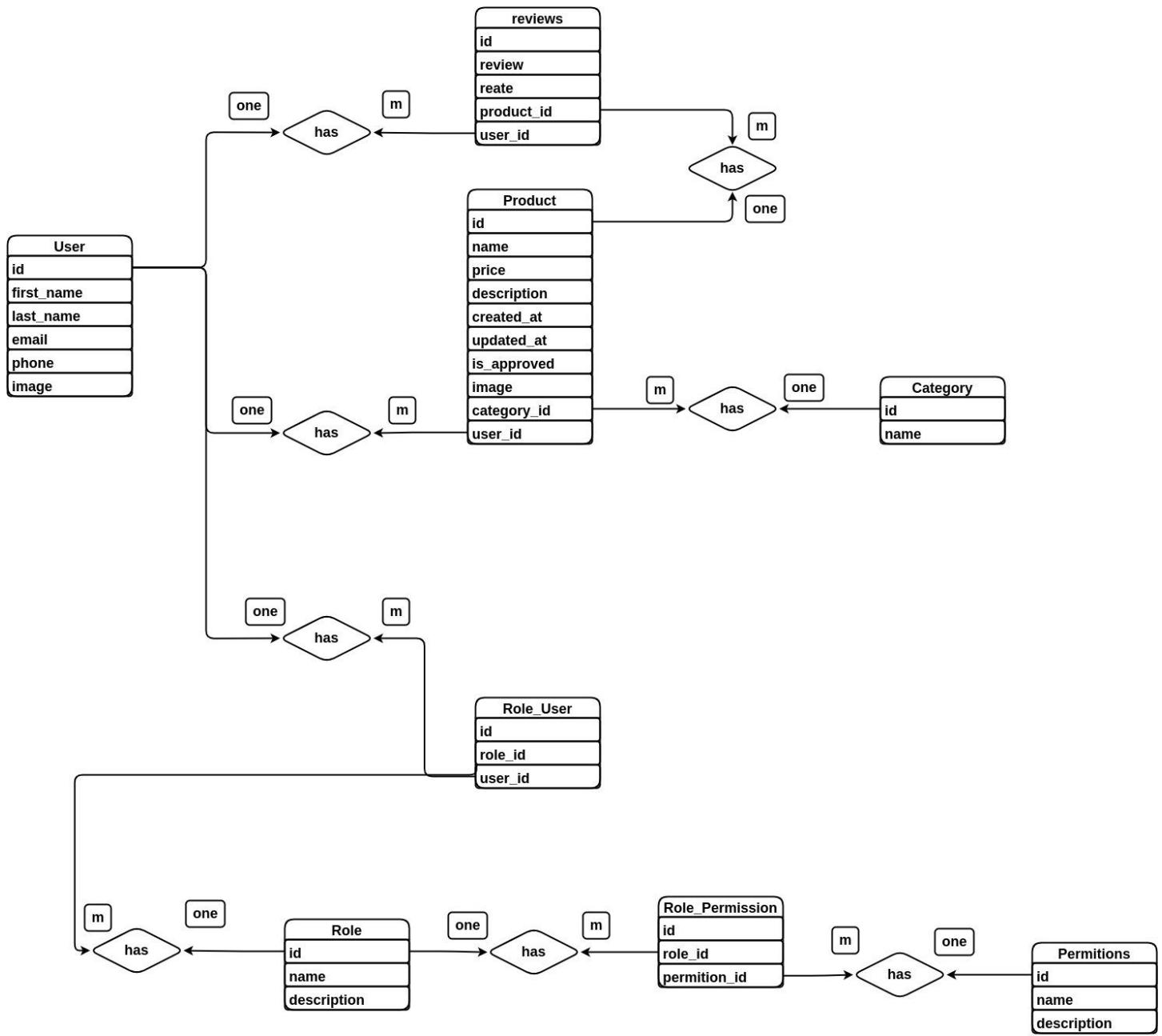


تصميم النظام

ERD Entity Relationship Diagram

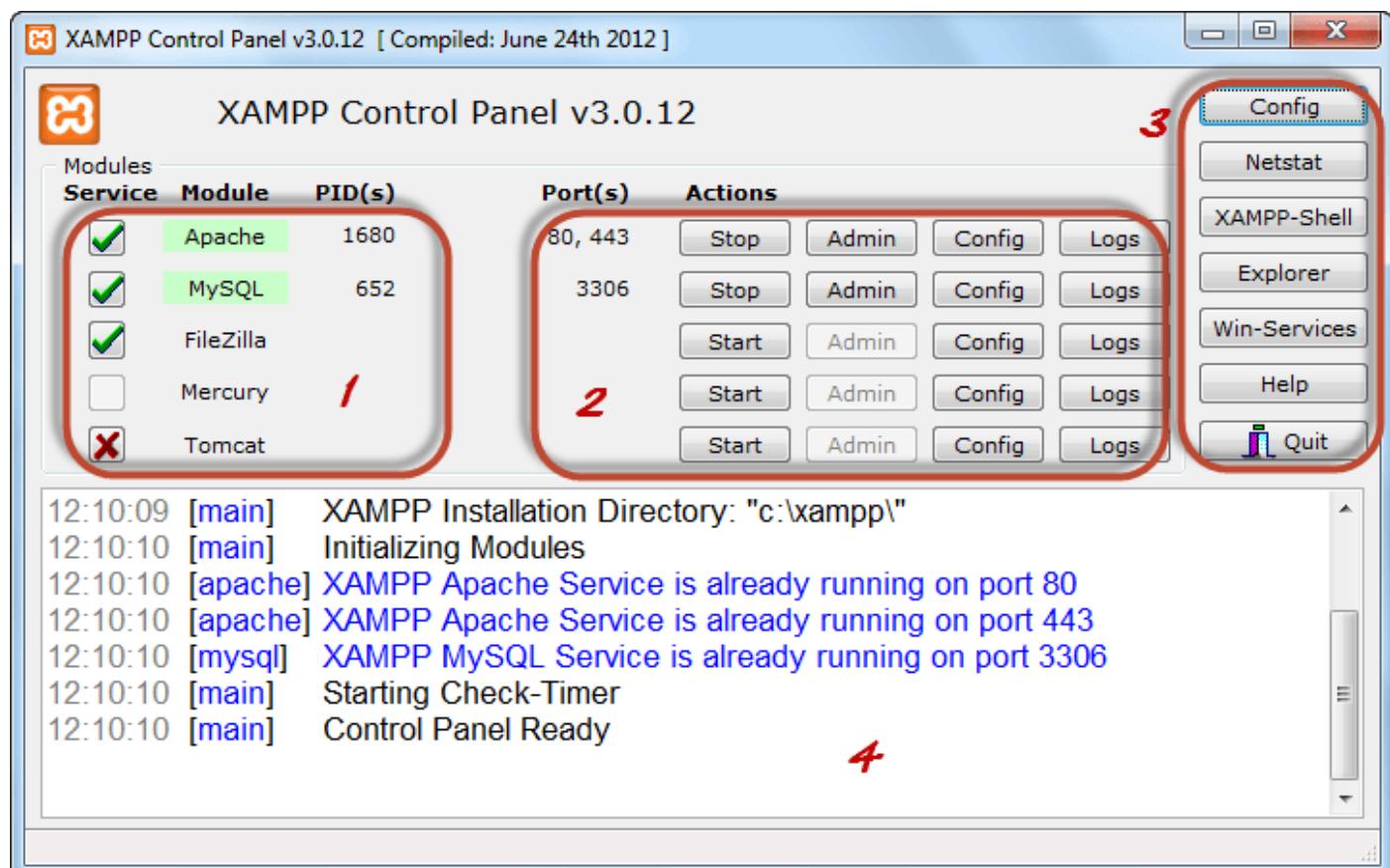


Database Design



التنفيذ

Xampp



1. السيرفرات وحالتهم

2. العمليات على السيرفرات

3. الاعدادات

٤. شاشة طباعة النتائج والأخطاء

- ❖ نقوم بفتح أي متصفح ويب ولكن google chrome
 - ❖ نقوم بكتابة هذا الرابط للدخول لقاعدة البيانات <http://localhost/phpmyadmin>
 - ❖ نضغط على new لإنشاء قاعدة بيانات جديدة
 - ❖ نقوم بتسهييلها graduate

Composer

نحتاج برنامج composer لتسطيب مشروع laravel جديد

<https://getcomposer.org>

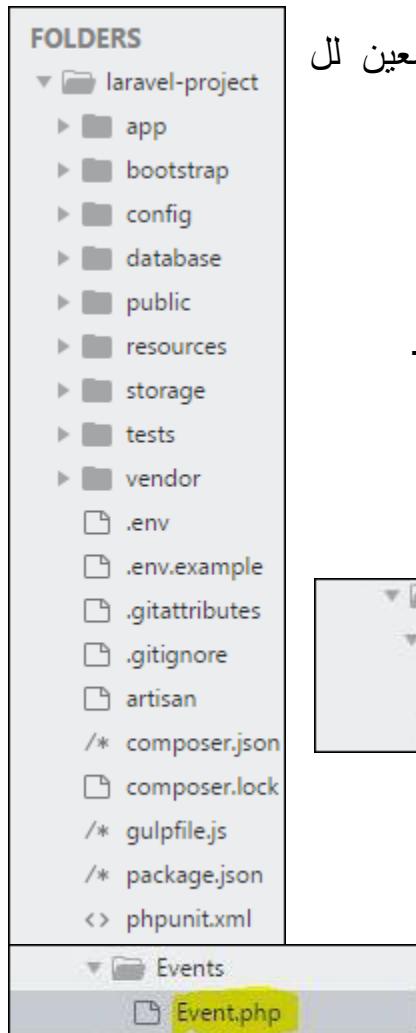
نقوم بتحميله وتنصيبيه

ولتتأكد من تنصيبه نفتح ال CMD ونكتب

Laravel

الذهاب إلى ال documentation الخاصة ب laravel وفتح ال CMD الأمر التالي

```
composer create-project laravel/laravel example-app
```



بعد تسطيب المشروع ستجد ان laravel تقوم بناء structure معين لل directories أو folders

: إنه مجلد التطبيق ويتضمن كود المصدر بالكامل للمشروع. • App

Artisan تتضمن :Console •

.Laravel Commands اللازمة لـ

يتضمن دليلاً يسمى Kernel.php حيث

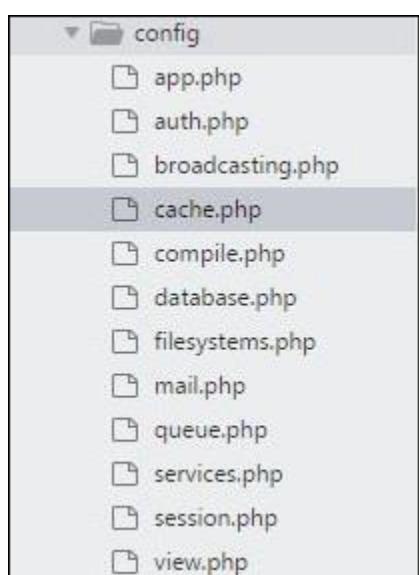
يتم التصريح عن جميع الأوامر

: يتضمن هذا المجلد جميع أحداث المشروع. تستخدم Events •

الأحداث لبدء الأنشطة وإظهار الأخطاء أو عمليات التحقق

اللازمة وتوفير قدر أكبر من المرونة.

- .Exceptions : يحتوي هذا المجلد على جميع الطرق اللازمة للتعامل مع ال Exceptions •
يحتوي أيضًا على ملف handle.php الذي يعالج جميع RequestsExceptions .
- Controllers ، Middlewares على مجلد Http على مجلدات فرعية ، •
نظرًا لأن Laravel يتبع نمط تصميم MVC ، فإن هذا المجلد يتضمن Models و Requests . Requests و Views و Controllers لها مكان آخر محدد.
- Jobs : يحتفظ دليل الوظائف بالأنشطة الموجودة في قائمة الانتظار لتطبيق Laravel •
- Listeners : يعتمد Events على ال Listeners و يتضامنون طرقًا تُستخدم للتعامل مع ال Exceptions وال Events .
- .Authorization هي فئات PHP التي تتضمن منطق ال Policies : Policies •
يتضمن هذا المجلد جميع مزودي الخدمة المطلوبين لتسجيل ال Events للخوادم •
 الأساسية وتهيئة تطبيق Laravel .
- Bootstrap : يحتوي هذا المجلد على كافة البرامج النصية الخاصة بتمهيد التطبيق. يحتوي على مجلد فرعي وهو ذاكرة التخزين المؤقت ، والذي يتضمن جميع الملفات المرتبطة بالتخزين المؤقت لتطبيق ويب. يمكنك أيضًا العثور على ملف app.php ، الذي يهيئ البرامج النصية اللازمة للتمهيد .



- Config : يشتمل مجلد الضبط على تكوينات متنوعة ومعلمات مرتبطة بها مطلوبة للتشغيل السلس لتطبيق Laravel. يتم عرض الملفات المختلفة المضمنة في مجلد التكوين كما هو موضح في الصورة هنا. تعمل أسماء الملفات وفقًا للوظيفة المرتبطة بها.

• Database : يحتوي على Database

Seeds : هذا يحتوي على الفئات المستخدمة لقاعدة بيانات الاختبار.

Migrations : يساعد هذا المجلد في استعلامات ترحيل قاعدة البيانات المستخدمة في تطبيق الويب.

Factories : يستخدم هذا المجلد لإنشاء عدد كبير من سجلات البيانات.

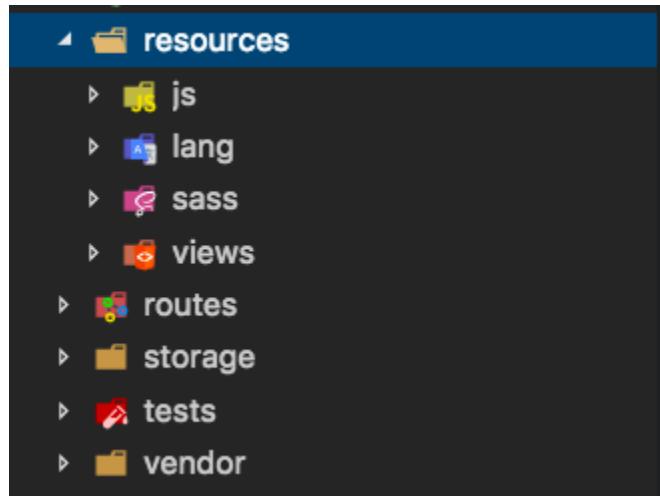
• Public : يحتوي على Public

htaccess : يعطي هذا الملف إعدادات الخادم.

.assets : تعتبر هذه الملفات javascript and css

index.php : هذا الملف مطلوب لتهيئة تطبيق الويب.

Resources : يحتوي على الملفات التي تعزز تطبيق الويب الخاص بك.



Assets : يتضمن مجلد الأصول ملفات

LESS و SCSS ، المطلوبة
لتصميم تطبيق الويب.

Lang : يتضمن هذا المجلد التكوين
للترجمة أو الاستيعاب.

Views : طرق العرض هي ملفات أو
قوالب HTML التي تتفاعل مع
المستخدمين النهائيين وتلعب دوراً أساسياً

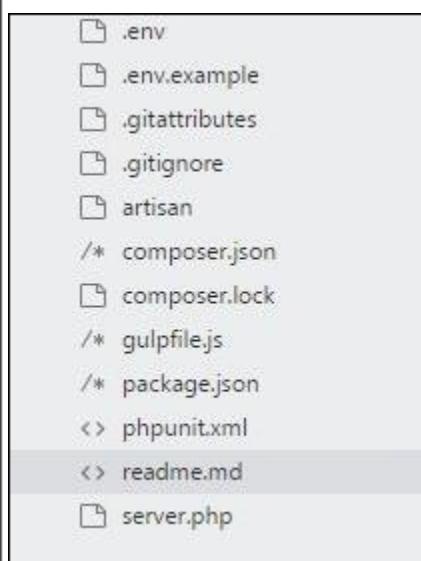
في بنية MVC.

• Storage : هذا هو المجلد الذي يخزن جميع السجلات والملفات

Laravel . الضرورية التي تحتاجها بشكل متكرر عند تشغيل مشروع

App : يحتوي هذا المجلد على الملفات التي تم استدعاؤها على
التوالي.

Framework : يحتوي على الجلسات وذاكرة التخزين المؤقت
وطرق العرض التي يتم استدعاؤها بشكل متكرر.



Logs : يتم تعقب كافة الاستثناءات وسجلات الأخطاء في هذا المجلد الفرعي.

Tests • end to end tests و configuration tests و unit tests : يحتوي على الـ

Vendor • dependencies : يحتوي على كل الـ

نقوم ببناء كل ال migrations الخاصة بالمشروع حيث اننا قمنا بتصميم ال database في خطوة system design k package نقوم ببنائها كلها ولكن آخر migration هي خاصه ب اسمها lara-trust سنقوم بالتحدث عنها لاحقا

```
▽ migrations
  2014_10_12_000000_create_users_table.php
  2014_10_12_100000_create_password_resets_table.php
  2019_08_19_000000_create_failed_jobs_table.php
  2019_12_14_000001_create_personal_access_tokens_table.php
  2022_02_16_154201_create_categories_table.php
  2022_02_16_234731_create_products_table.php
  2022_02_17_000558_create_reviews_table.php
  2022_03_14_014439_laratrust_setup_tables.php
```



user migration

```
1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 return new class extends Migration
8 {
9     /**
10      * Run the migrations.
11      *
12      * @return void
13      */
14     public function up()
15     {
16         Schema::create('users', function (Blueprint $table) {
17             $table->id();
18             $table->string('name');
19             $table->string('email')->unique();
20             $table->timestamp('email_verified_at')->nullable();
21             $table->string('password');
22             $table->string('phone')->nullable();
23             $table->text('image')->nullable();
24             $table->rememberToken();
25             $table->timestamps();
26         });
27     }
28
29     /**
30      * Reverse the migrations.
31      *
32      * @return void
33      */
34     public function down()
35     {
36         Schema::dropIfExists('users');
37     }
38 };
39
```

reset password migration

هذا الملف يكون جاهز في مشروع laravel ويستخدم للسماح لـ user من تغيير ال password

```
1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  return new class extends Migration
8  {
9      /**
10      * Run the migrations.
11      *
12      * @return void
13      */
14     public function up()
15     {
16         Schema::create('password_resets', function (Blueprint $table) {
17             $table->string('email')->index();
18             $table->string('token');
19             $table->timestamp('created_at')->nullable();
20         });
21     }
22
23     /**
24     * Reverse the migrations.
25     *
26     * @return void
27     */
28     public function down()
29     {
30         Schema::dropIfExists('password_resets');
31     }
32 };
33
```

هذا الملف يكون جاهز في مشروع laravel ويستخدم لتخزين الـ failed jobs migration فشلت في التنفيذ

```
1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 return new class extends Migration
8 {
9     /**
10      * Run the migrations.
11      *
12      * @return void
13      */
14     public function up()
15     {
16         Schema::create('failed_jobs', function (Blueprint $table) {
17             $table->id();
18             $table->string('uuid')->unique();
19             $table->text('connection');
20             $table->text('queue');
21             $table->longText('payload');
22             $table->longText('exception');
23             $table->timestamp('failed_at')->useCurrent();
24         });
25     }
26
27     /**
28      * Reverse the migrations.
29      *
30      * @return void
31      */
32     public function down()
33     {
34         Schema::dropIfExists('failed_jobs');
35     }
36 };
37
```

personal access tokens



```
1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  return new class extends Migration
8  {
9      /**
10     * Run the migrations.
11     *
12     * @return void
13     */
14    public function up()
15    {
16        Schema::create('personal_access_tokens', function (Blueprint $table) {
17            $table->id();
18            $table->morphs('tokenable');
19            $table->string('name');
20            $table->string('token', 64)->unique();
21            $table->text('abilities')->nullable();
22            $table->timestamp('last_used_at')->nullable();
23            $table->timestamps();
24        });
25    }
26
27    /**
28     * Reverse the migrations.
29     *
30     * @return void
31     */
32    public function down()
33    {
34        Schema::dropIfExists('personal_access_tokens');
35    }
36};
37
```

categories جدول ال



```
1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 return new class extends Migration
8 {
9     /**
10      * Run the migrations.
11      *
12      * @return void
13      */
14     public function up()
15     {
16         Schema::create('categories', function (Blueprint $table) {
17             $table->id();
18             $table->string('name');
19             $table->timestamps();
20         });
21     }
22
23     /**
24      * Reverse the migrations.
25      *
26      * @return void
27      */
28     public function down()
29     {
30         Schema::dropIfExists('categories');
31     }
32 };
33 
```

جدول ال products

```
1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 return new class extends Migration
8 {
9     /**
10      * Run the migrations.
11      *
12      * @return void
13      */
14     public function up()
15     {
16         Schema::create('products', function (Blueprint $table) {
17             $table->id();
18             $table->string('name');
19             $table->text('description');
20             $table->boolean('is_approved')->nullable();
21             $table->text('image')->nullable();
22             $table->unsignedBigInteger('category_id')->nullable();
23             $table->unsignedBigInteger('user_id')->nullable();
24             $table->timestamps();
25
26             $table->foreign('category_id')->references('id')->on('categories')->onDelete('set null');
27             $table->foreign('user_id')->references('id')->on('users')->onDelete('set null');
28         });
29     }
30
31     /**
32      * Reverse the migrations.
33      *
34      * @return void
35      */
36     public function down()
37     {
38         Schema::dropIfExists('products');
39     }
40 };
41
```

reviews جدول ال



```
1  <?php
2
3  use Illuminate\Database\Migrations\Migration;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Support\Facades\Schema;
6
7  return new class extends Migration
8  {
9      /**
10      * Run the migrations.
11      *
12      * @return void
13      */
14     public function up()
15     {
16         Schema::create('reviews', function (Blueprint $table) {
17             $table->id();
18             $table->text('review');
19             $table->boolean('rate')->default(0);
20             $table->unsignedBigInteger('product_id')->nullable();
21             $table->unsignedBigInteger('user_id')->nullable();
22             $table->timestamps();
23
24             $table->foreign('product_id')->references('id')->onDelete('set null');
25             $table->foreign('user_id')->references('id')->on('users')->onDelete('set null');
26         });
27     }
28
29     /**
30     * Reverse the migrations.
31     *
32     * @return void
33     */
34     public function down()
35     {
36         Schema::dropIfExists('reviews');
37     }
38 };
39
```

جدول الـ roles و الـ permissions
هذا الجدول يتم بناؤه تلقائيا من الـ package

```

1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 class LaratrustSetupTables extends Migration
8 {
9     /**
10      * Run the migrations.
11      *
12      * @return void
13      */
14     public function up()
15     {
16         // Create table for storing roles
17         Schema::create('roles', function (Blueprint $table) {
18             $table->bigIncrements('id');
19             $table->string('name')->unique();
20             $table->string('display_name')->nullable();
21             $table->string('description')->nullable();
22             $table->timestamps();
23         });
24
25         // Create table for storing permissions
26         Schema::create('permissions', function (Blueprint $table) {
27             $table->bigIncrements('id');
28             $table->string('name')->unique();
29             $table->string('display_name')->nullable();
30             $table->string('description')->nullable();
31             $table->timestamps();
32         });
33
34         // Create table for associating roles to users and teams (Many To Many Polymorphic)
35         Schema::create('role_user', function (Blueprint $table) {
36             $table->unsignedBigInteger('role_id');
37             $table->unsignedBigInteger('user_id');
38             $table->string('user_type');
39
40             $table->foreign('role_id')->references('id')->on('roles')
41                 ->onDelete('cascade');
42
43             $table->primary(['user_id', 'role_id', 'user_type']);
44         });
45
46         // Create table for associating permissions to users (Many To Many Polymorphic)
47         Schema::create('permission_user', function (Blueprint $table) {
48             $table->unsignedBigInteger('permission_id');
49             $table->unsignedBigInteger('user_id');
50             $table->string('user_type');
51
52             $table->foreign('permission_id')->references('id')->on('permissions')
53                 ->onDelete('cascade');
54
55             $table->primary(['user_id', 'permission_id', 'user_type']);
56         });
57
58         // Create table for associating permissions to roles (Many-to-Many)
59         Schema::create('permission_role', function (Blueprint $table) {
60             $table->unsignedBigInteger('permission_id');
61             $table->unsignedBigInteger('role_id');
62
63             $table->foreign('permission_id')->references('id')->on('permissions')
64                 ->onDelete('cascade');
65             $table->foreign('role_id')->references('id')->on('roles')
66                 ->onDelete('cascade');
67
68             $table->primary(['permission_id', 'role_id']);
69         });
70     }
71
72     /**
73      * Reverse the migrations.
74      *
75      * @return void
76      */
77     public function down()
78     {
79         Schema::dropIfExists('permission_user');
80         Schema::dropIfExists('permission_role');
81         Schema::dropIfExists('permissions');
82         Schema::dropIfExists('role_user');
83         Schema::dropIfExists('roles');
84     }
85 }

```

[/https://laratrust.santigarcor.me](https://laratrust.santigarcor.me)

نقوم بتنزيل laratrust

```
composer require santigarcor/laratrust
```

sh

نقوم بنشر اعدادات ال package

```
php artisan vendor:publish --tag="laratrust"
```

sh

```
php artisan laratrust:setup
```

sh

```
composer dump-autoload
```

sh

ونجد الملف موجود في ال migrations

set up the .env file

ال اسم نضع

وال database

اذا وجد password

```
1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=base64:wVqUHbT1u+NWjUu6ALZNDZkcxF3Gt63090nP9v1wyWY=
4 APP_DEBUG=true
5 APP_URL=http://localhost
6
7 LOG_CHANNEL=stack
8 LOG_DEPRECATIONS_CHANNEL=null
9 LOG_LEVEL=debug
10
11 DB_CONNECTION=mysql
12 DB_HOST=127.0.0.1
13 DB_PORT=3306
14 DB_DATABASE=graduate
15 DB_USERNAME=mohammed
16 DB_PASSWORD=123
17
18 BROADCAST_DRIVER=log
19 CACHE_DRIVER=file
20 FILESYSTEM_DISK=local
21 QUEUE_CONNECTION=sync
22 SESSION_DRIVER=file
23 SESSION_LIFETIME=120
24
25 MEMCACHED_HOST=127.0.0.1
26
27 REDIS_HOST=127.0.0.1
28 REDIS_PASSWORD=null
29 REDIS_PORT=6379
30
31 MAIL_MAILER=smtp
32 MAIL_HOST=mailhog
33 MAIL_PORT=1025
34 MAIL_USERNAME=null
35 MAIL_PASSWORD=null
36 MAIL_ENCRYPTION=null
37 MAIL_FROM_ADDRESS=null
38 MAIL_FROM_NAME="${APP_NAME}"
39
40 AWS_ACCESS_KEY_ID=
41 AWS_SECRET_ACCESS_KEY=
42 AWS_DEFAULT_REGION=us-east-1
43 AWS_BUCKET=
44 AWS_USE_PATH_STYLE_ENDPOINT=false
45
46 PUSHER_APP_ID=
47 PUSHER_APP_KEY=
48 PUSHER_APP_SECRET=
49 PUSHER_APP_CLUSTER=mt1
50
51 MIX_PUSHER_APP_KEY="${PUSHER_APP_KEY}"
52 MIX_PUSHER_APP_CLUSTER="${PUSHER_APP_CLUSTER}"
```

ثم نقوم بفتح ال cmd في مسار المشروع ونستخدم

php artisan migrate

وهذا ال command سيقوم ببناء كل جداول ال database

نقوم ببناء ال factories والتي تستخدم لوضع داتا وهمية للتجربة في ال database

category factory

```
1  <?php
2
3  namespace Database\Factories;
4
5  use Illuminate\Database\Eloquent\Factories\Factory;
6
7  /**
8   * @extends \Illuminate\Database\Eloquent\Factories\Factory<\App\Models\Category>
9   */
10 class CategoryFactory extends Factory
11 {
12     /**
13      * Define the model's default state.
14      *
15      * @return array
16      */
17     public function definition()
18     {
19         return [
20             'name'=>$this->faker->name()
21         ];
22     }
23 }
24
```

product factory

```
● ● ●  
1 <?php  
2  
3 namespace Database\Factories;  
4  
5 use Illuminate\Database\Eloquent\Factories\Factory;  
6  
7 /**  
8  * @extends \Illuminate\Database\Eloquent\Factories\Factory<\App\Models\Product>  
9 */  
10 class ProductFactory extends Factory  
11 {  
12     /**  
13      * Define the model's default state.  
14      *  
15      * @return array  
16      */  
17     public function definition()  
18     {  
19         return [  
20             'name'=>$this->faker->name(),  
21             'description'=>$this->faker->text(),  
22             'is_approved'=>$this->faker->boolean(),  
23             'category_id'=>rand(1,9),  
24             'user_id'=>rand(1,9)  
25         ];  
26     }  
27 }  
28
```

user factory

```
1 <?php
2
3 namespace Database\Factories;
4
5 use Illuminate\Database\Eloquent\Factories\Factory;
6 use Illuminate\Support\Str;
7
8 /**
9  * @extends \Illuminate\Database\Eloquent\Factories\Factory<\App\Models\User>
10 */
11 class UserFactory extends Factory
12 {
13     /**
14      * Define the model's default state.
15      *
16      * @return array
17      */
18     public function definition()
19     {
20         return [
21             'name' => $this->faker->name(),
22             'email' => $this->faker->unique()->safeEmail(),
23             'phone' => $this->faker->unique()->phoneNumber(),
24             'email_verified_at' => now(),
25             'password' => '$2y$10$92IXUNpkj00rOQ5byMi.Ye4oKoEa3Ro9llC/.og/at2.uheWG/igi', // password
26             'remember_token' => Str::random(10),
27         ];
28     }
29
30     /**
31      * Indicate that the model's email address should be unverified.
32      *
33      * @return \Illuminate\Database\Eloquent\Factories\Factory
34      */
35     public function unverified()
36     {
37         return $this->state(function (array $attributes) {
38             return [
39                 'email_verified_at' => null,
40             ];
41         });
42     }
43 }
44
```

نقوم ببناء **seeder** وهو ما يستخدم لتشغيل ال factories

ثم نقوم بتشغيل ال seeder عن طريق ال command

`php artisan db:seed`

وكلما اردنا ان نقوم بتعديل في تركيب ال database

نقوم بتشغيل ال command

`php artisan migrate:fresh`

ثم نقوم بـ

`php artisan db:seed`

أو يمكن تشغيل الإثنين عن طريق

`php artisan migrate:fresh --seed`

ثم نقوم بناء ال routes في ملف routes/web.php ال الخاصة بالداشبورد

```
● ● ●
1 Route::group(['prefix' =>'admin' , 'as'=>'admin.' , 'middleware' => ['admin','auth']],function (){
2     Route::get('/',[HomeController::class , 'index'])->name('index');
3
4     Route::resource('category',CategoryController::class);
5     Route::resource('user',UserController::class);
6     Route::resource('product',ProductController::class);
7     Route::resource('review',ReviewController::class);
8 });
9
```

هذا ال routes تقوم بتوصيل ال url الى دالة معينة في ال controller

كلمة endpoints معناها انها تقوم ببناء 7 resource

Index .1

Create .2

Store .3

Edit .4

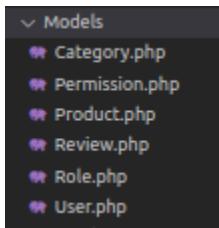
Update .5

Delete .6

Show .7

ال routes الخاصة بالواجهة

```
1 Route::group(['prefix' => '/', 'as' => 'home.'],function(){
2     Route::get('/',[FrontHomeController::class,'index'])->name('index');
3     Route::get('/about',[FrontHomeController::class,'about'])->name('about');
4     Route::get('registerPage',[FrontAuthController::class,'registerPage'])->name('registerPage')->middleware('guest');
5     Route::post('register',[FrontAuthController::class,'register'])->name('register')->middleware('guest');
6     Route::get('loginPage',[FrontAuthController::class,'loginPage'])->name('loginPage')->middleware('guest');
7     Route::post('login',[FrontAuthController::class,'login'])->name('login')->middleware('guest');
8     Route::get('logout',[FrontAuthController::class,'logout'])->name('logout')->middleware('auth');
9
10    Route::group(['prefix' => 'editor', 'as' => 'editor.', 'middleware' => 'editor'],function() {
11        Route::get('/profile',[EditorController::class,'profile'])->name('profile');
12        Route::get('/edit/{user}',[EditorController::class,'edit'])->name('edit');
13        Route::post('/update/{user}',[EditorController::class,'update'])->name('update');
14        Route::get('/movie/create',[EditorController::class,'movieCreate'])->name('movieCreate');
15        Route::post('/movie/store',[EditorController::class,'movieStore'])->name('movieStore');
16        Route::get('/movie/edit/{product}',[EditorController::class,'movieEdit'])->name('movieEdit');
17        Route::put('/movie/update/{product}',[EditorController::class,'movieUpdate'])->name('movieUpdate');
18        Route::delete('/movie/delete/{product}',[EditorController::class,'movieDelete'])->name('movieDelete');
19    });
20
21    Route::group(['prefix' => 'viewer', 'as' => 'viewer.', 'middleware' => 'viewer'], function(){
22        Route::get('profile',[ViewerController::class , 'profile'])->name('profile');
23        Route::get('edit/{user}',[ViewerController::class , 'edit'])->name('edit');
24        Route::post('update/{user}',[ViewerController::class , 'update'])->name('update');
25    });
26
27    Route::group(['prefix' => 'movies' , 'as' => 'movies.'], function(){
28        Route::get('index',[MovieController::class,'index'])->name('index');
29        Route::get('show/{product}',[MovieController::class,'show'])->name('show');
30    });
31
32
33    Route::group(['prefix' => 'review' , 'as' => 'review.','middleware' => 'auth'],function(){
34        Route::post('create',[FrontReviewController::class,'create'])->name('create');
35    });
36});
```



ثم نقوم بناء ال **models** ال **controller** هي حلقة الوصل بين ال **models** وال **database**

ال **category** الخاص بال **model**

```
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 $guarded =[];
12 }
13
```

ال **trait** المسمى ب **HasFactory** لكي تعمل ال **factories** ضروري

ال **guarded** معناه أنه لا يوجد اي **column** محمي ضد أن تقوم بالتخزين فيه في ال **database**

ال permission بال model الخاص

```
1 <?php
2
3 namespace App\Models;
4
5 use Laratrust\Models\LaratrustPermission;
6
7 class Permission extends LaratrustPermission
8 {
9     public $guarded = [];
10 }
11
```

الـ model بالـ product الخاص



```
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7
8 class Product extends Model
9 {
10     use HasFactory;
11     protected $guarded = [];
12
13     public function category()
14     {
15         return $this->belongsTo(Category::class, 'category_id');
16     }
17
18     public function user()
19     {
20         return $this->belongsTo(User::class, 'user_id');
21     }
22
23     public function reviews()
24     {
25         return $this->hasMany(Review::class, 'product_id');
26     }
27
28     public function scopeIsApproved($query)
29     {
30         return $query->where('is_approved', true);
31     }
32 }
33 }
```

أيضا نقوم بتعريف ال relations العلاقة اذا كانت one to one نستخدم الدالة belongsTo أو كانت many to many أو hasMany نستخدم الدالةhasMany يقوم بعمل امتداد .review ال الخاص بال model query

```
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7
8 class Review extends Model
9 {
10     use HasFactory;
11     protected $guarded =[];
12
13     public function product(){
14         return $this->belongsTo(Product::class,'product_id');
15     }
16
17     public function user(){
18         return $this->belongsTo(User::class,'user_id');
19     }
20 }
```

الـ role الخاص بالـ model



```
1 <?php
2
3 namespace App\Models;
4
5 use Laratrust\Models\LaratrustRole;
6
7 class Role extends LaratrustRole
8 {
9     public $guarded = [];
10 }
11
```

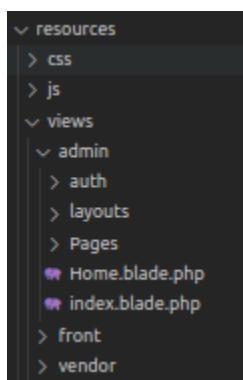
الـ userـ modelـ بالـ الخاصـ

```
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Contracts\Auth\MustVerifyEmail;
6 use Illuminate\Database\Eloquent\Factories\HasFactory;
7 use Illuminate\Foundation\Auth\User as Authenticatable;
8 use Illuminate\Notifications\Notifiable;
9 use Laravel\Sanctum\HasApiTokens;
10 use Laratrust\Traits\LaratrustUserTrait;
11
12 class User extends Authenticatable
13 {
14     use LaratrustUserTrait;
15     use HasApiTokens, HasFactory, Notifiable;
16
17     /**
18      * The attributes that are mass assignable.
19      *
20      * @var array<int, string>
21      */
22     protected $guarded = [];
23
24     /**
25      * The attributes that should be hidden for serialization.
26      *
27      * @var array<int, string>
28      */
29     protected $hidden = [
30         'password',
31         'remember_token',
32     ];
33
34     /**
35      * The attributes that should be cast.
36      *
37      * @var array<string, string>
38      */
39     protected $casts = [
40         'email_verified_at' => 'datetime',
41     ];
42 }
```

نقوم ببناء ال controllers

Home Controller

```
1 <?php
2
3 namespace App\Http\Controllers\Admin;
4
5 use App\Http\Controllers\Controller;
6 use App\Models\Category;
7 use App\Models\Product;
8 use App\Models\Review;
9 use App\Models\User;
10
11 class HomeController extends Controller
12 {
13     public function index(){
14         $categoryCount = Category::count();
15         $productCount = Product::count();
16         $userCount = User::count();
17         $reviewCount = Review::count();
18         return view('admin.Home',compact('categoryCount','productCount','userCount','reviewCount'));
19     }
20 }
21
```



هذا ال controller كما هو موضح من قبل يحتوي على مجموعة من الدوال التي توصل من endpoint معينة تم تعريفها في ال routes وتوصل إلى view معين ال view هو باختصار كود الواجهة الذي تتم عليه عملية ال render من كود html .blade , وفي laravel نستخدم template language قويه جدا اسمها .css , .js admin.home هذا ال controller فيه داله واحده توصل إلى ال view الذي مساره

admin.home view ↴

```
1 @extends('admin.index')
2
3 @section('content')
4     <div class="row">
5         <div class="col-lg-3 col-6">
6             <!-- small box -->
7             <div class="small-box bg-info">
8                 <div class="inner">
9                     <h3>{$categoryCount}</h3>
10
11                     <p>Categories</p>
12                 </div>
13                 <div class="icon">
14                     <i class="ion ion-bag"></i>
15                 </div>
16                 <a href="{{route('admin.category.index')}}" class="small-box-footer">More info <i class="fas fa-arrow-circle-right"></i></a>
17             </div>
18         <!-- ./col -->
19         <div class="col-lg-3 col-6">
20             <!-- small box -->
21             <div class="small-box bg-success">
22                 <div class="inner">
23                     <h3>{$productCount}</h3>
24
25                     <p>Movies Count</p>
26                 </div>
27                 <div class="icon">
28                     <i class="ion ion-stats-bars"></i>
29                 </div>
30                 <a href="{{route('admin.product.index')}}" class="small-box-footer">More info <i class="fas fa-arrow-circle-right"></i></a>
31             </div>
32         </div>
33     <!-- ./col -->
34     <div class="col-lg-3 col-6">
35         <!-- small box -->
36         <div class="small-box bg-warning">
37             <div class="inner">
38                 <h3>{$userCount}</h3>
39
40                     <p>User Registrations</p>
41                 </div>
42                 <div class="icon">
43                     <i class="ion ion-person-add"></i>
44                 </div>
45                 <a href="{{route('admin.user.index')}}" class="small-box-footer">More info <i class="fas fa-arrow-circle-right"></i></a>
46             </div>
47         </div>
48     <!-- ./col -->
49     <div class="col-lg-3 col-6">
50         <!-- small box -->
51         <div class="small-box bg-danger">
52             <div class="inner">
53                 <h3>{$reviewCount}</h3>
54
55                     <p>Review Count</p>
56                 </div>
57                 <div class="icon">
58                     <i class="ion ion-pie-graph"></i>
59                 </div>
60                 <a href="{{route('admin.review.index')}}" class="small-box-footer">More info <i class="fas fa-arrow-circle-right"></i></a>
61             </div>
62         </div>
63     <!-- ./col -->
64
65     </div>
66     @endsection
67
```

Auth Controller

```
1 <?php
2
3 namespace App\Http\Controllers\Admin;
4
5 use App\Http\Controllers\Controller;
6 use App\Http\Requests\loginRequest;
7 use App\Models\User;
8 use Illuminate\Http\Request;
9 use Illuminate\Support\Facades\Auth;
10 use RealRashid\SweetAlert\Facades\Alert;
11
12 class AuthController extends Controller
13 {
14
15     public function loginPage()
16     {
17         return view('admin.auth.login');
18     }
19
20     public function login(LoginRequest $request)
21     {
22         $cred = $request->only('email', 'password');
23         $user = User::where('email', $request->email)->first();
24
25         if ($user->hasRole('admin')) {
26             if (Auth::attempt($cred)) {
27                 return redirect(route('admin.index'));
28             }
29         }
30         Alert::error('error', 'You are not allowed');
31         return redirect()->back();
32     }
33
34
35     public function logout()
36     {
37         Auth::logout();
38
39         return redirect()->back();
40     }
41 }
```

• دالة الـ loginPage : تفتح صفحة الـ login

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <meta charset="utf-8">
5     <meta http-equiv="X-UA-Compatible" content="IE=edge">
6     <title>AdminLTE 3 | Log in</title>
7     <!-- Tell the browser to be responsive to screen width -->
8     <meta name="viewport" content="width=device-width, initial-scale=1">
9
10    <!-- Font Awesome -->
11    <link rel="stylesheet" href="{{asset('AdminLET/plugins/fontawesome-free/css/all.min.css')}}">
12    <!-- Ionicons -->
13    <link rel="stylesheet" href="https://code.ionicframework.com/ionicons/2.0.1/css/ionicons.min.css">
14    <!-- iCheck bootstrap -->
15    <link rel="stylesheet" href="{{asset('AdminLET/plugins/iCheck-bootstrap/icheck-bootstrap.min.css')}}">
16    <!-- Theme style -->
17    <link rel="stylesheet" href="{{asset('AdminLET/dist/css/adminlte.min.css')}}">
18    <!-- Google Font: Source Sans Pro -->
19    <link href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,400i,700" rel="stylesheet">
20 </head>
21 <body class="hold-transition login-page">
22 <div class="login-box">
23     <div class="login-logo">
24         <a href=""><b>Admin</b>LTE</a>
25     </div>
26     <!-- /.login-logo -->
27     <div class="card">
28         <div class="card-body login-card-body">
29             <p class="login-box-msg">Sign in to start your session</p>
30
31             <form method="post" action="{{route('admin.login')}}">
32                 @csrf
33                 <div class="input-group mb-3">
34                     <input type="email" class="form-control" name="email" placeholder="Email">
35                     <div class="input-group-append">
36                         <div class="input-group-text">
37                             <span class="fas fa-envelope"></span>
38                         </div>
39                     </div>
40                 </div>
41                 <div class="input-group mb-3">
42                     <input type="password" class="form-control" name="password" placeholder="Password">
43                     <div class="input-group-append">
44                         <div class="input-group-text">
45                             <span class="fas fa-lock"></span>
46                         </div>
47                     </div>
48                 </div>
49                 <div class="row">
50                     <div class="col-6">
51                         <button type="submit" class="btn btn-primary btn-block">Sign In</button>
52                     </div>
53                     <!-- /.col -->
54                 </div>
55             </form>
56
57             <!-- /.login-card-body -->
58         </div>
59     </div>
60 </div>
61 <!-- /.login-box -->
62
63 <!-- jQuery -->
64 <script src="{{asset('AdminLET/plugins/jquery/jquery.min.js')}}"></script>
65 <!-- Bootstrap 4 -->
66 <script src="{{asset('AdminLET/plugins/bootstrap/js/bootstrap.bundle.min.js')}}"></script>
67 <!-- AdminLTE App -->
68 <script src="{{asset('AdminLET/dist/js/adminlte.min.js')}}"></script>
69
70 @include('sweetalert::alert')
71
72 </body>
73 </html>
74
```

- دالة ال **login** لعمل تسجيل دخول وتسجيل بيانات المستخدم في ال **session**
- دالة ال **logout** لتسجيل الخروج وحذف ال **session**

Category Controller

```
● ● ●  
1  <?php  
2  
3  namespace App\Http\Controllers\Admin;  
4  use App\Http\Controllers\Controller;  
5  use App\Models\Category;  
6  use App\Http\Requests\StoreCategoryRequest;  
7  use App\Http\Requests\UpdateCategoryRequest;  
8  use RealRashid\SweetAlert\Facades\Alert;  
9  
10 class CategoryController extends Controller  
11 {  
12     public function index()  
13     {  
14         $categories =Category::paginate(5);  
15         return view('admin.Pages.Category.index',compact('categories'));  
16     }  
17  
18     public function create()  
19     {  
20         return view('admin.Pages.Category.create');  
21     }  
22  
23     public function store(StoreCategoryRequest $request)  
24     {  
25         Category::create($request->validated());  
26         Alert::success('Success Title', 'Category Was Created');  
27         return redirect(route('admin.category.index'));  
28     }  
29 }
```



```
1 public function edit(Category $category)
2 {
3     return view('admin.Pages.Category.edit', compact('category'));
4 }
5
6 public function update(UpdateCategoryRequest $request, Category $category)
7 {
8     $category->update($request->validated());
9     Alert::success('Success Title', 'Category Was Updated');
10    return redirect(route('admin.category.index'));
11 }
12
13 public function destroy(Category $category)
14 {
15     $category->delete();
16     Alert::success('Success Title', 'Category Was Deleted');
17     return redirect(route('admin.category.index'));
18 }
19 }
20 }
```

```

1 @extends('admin.index')
2
3 @section('content')
4     <div class="card">
5         <div class="card-header">
6             <h3 class="card-title">Categories</h3>
7             <a href="{{route('admin.category.create')}}" class="btn btn-success float-right" Create New Category</a>
8         </div>
9         <!-- /.card-header -->
10        <div class="card-body">
11            <div id="example2_wrapper" class="dataTables_wrapper dt-bootstrap4">
12                <div class="row"><div class="col-sm-12 col-md-6"></div>
13                <div class="col-sm-12 col-md-6"></div>
14            </div>
15            <div class="row">
16                <div class="col-sm-12">
17                    <table id="example2" class="table table-bordered table-hover dataTable dtr-inline" role="grid" aria-describedby="example2_info">
18                        <thead>
19                            <tr role="row">
20                                <th class="sorting asc" tabindex="0" aria-controls="example2" rowspan="1" colspan="1" aria-sort="ascending" aria-label="Rendering engine: activate to sort column descending">#</th>
21                                <th class="sorting" tabindex="0" aria-controls="example2" rowspan="1" colspan="1" aria-label="Browser: activate to sort column ascending">Name</th>
22                                <th class="sorting" tabindex="0" aria-controls="example2" rowspan="1" colspan="1" aria-label="Browser: activate to sort column ascending">Edit</th>
23                                <th class="sorting" tabindex="0" aria-controls="example2" rowspan="1" colspan="1" aria-label="Browser: activate to sort column ascending">Delete</th>
24
25
26                            </thead>
27                            <tbody>
28                                @foreach($categories as $category)
29                                    <tr role="row" class="odd">
30                                        <td tabindex="0" class="sorting_1">{{ $category->id }}</td>
31                                        <td>{{ $category->name }}</td>
32                                        <td>
33                                            <form method="get" action="{{ route('admin.category.edit', $category) }}">
34                                                @csrf
35                                                <button type="submit" class="btn btn-success">
36                                                    <i class="fa fa-edit"></i>
37                                                </button>
38                                            </form>
39                                        </td>
40                                        <td>
41                                            <form method="POST" action="{{ route('admin.category.destroy', $category) }}">
42                                                @csrf
43                                                @method('DELETE')
44                                                <button type="submit" class="btn btn-danger">
45                                                    <i class="fa fa-trash"></i>
46                                                </button>
47                                            </form>
48                                        </td>
49                                    </tr>
50                                @endforeach
51
52                            </tbody>
53
54                        </table>
55                    </div>
56                <div class="row">
57                    <div class="col-sm-12 col-md-5">
58                        <div class="dataTables_info" id="example2_info" role="status" aria-live="polite">Showing 1 to 10 of 57 entries</div>
59                    </div>
60                    <div class="col-sm-12 col-md-7">
61                        <div class="dataTables_paginate paging_simple_numbers" id="example2_paginate">
62                            {{ $categories->links() }}
63                        </div>
64                    </div>
65                </div>
66            </div>
67        </div>
68    </div>
69    <!-- /.card-body -->
70 </div>
71 @endsection
72

```

● دالة **create** تقوم بفتح صفحة إنشاء التصنيف

```
1 @extends('admin.index')
2
3 @section('content')
4     <div class="col-md-12">
5         <div class="card card-primary">
6             <div class="card-header">
7                 <h3 class="card-title">Create New Category</h3>
8             </div>
9
10            <form method="POST" action="{{route('admin.category.store')}}">
11                @csrf
12                <div class="card-body">
13                    <div class="form-group">
14                        <label for="exampleInputEmail1">Name</label>
15                        <input type="text" name="name" class="form-control" id="exampleInputEmail1">
16                    </div>
17                    @error('name')
18                        <p class="text-danger"> {{$message}}</p>
19                    @enderror()
20                </div>
21                <div class="card-footer">
22                    <button type="submit" class="btn btn-primary">Submit</button>
23                </div>
24            </form>
25
26        </div>
27        <!-- /.card -->
28
29    </div>
30
31 @endsection
32
```

- دالة الـ **edit** تقوم بفتح صفحة تعديل التصنيف

```

1  @extends('admin.index')
2
3  @section('content')
4      <div class="col-md-12">
5          <div class="card card-primary">
6              <div class="card-header">
7                  <h3 class="card-title">Create New Category</h3>
8              </div>
9
10             <form method="POST" action="{{route('admin.category.update',$category)}}>
11                 @csrf
12                 @method('PUT')
13                 <div class="card-body">
14                     <div class="form-group">
15                         <label for="exampleInputEmail1">Name</label>
16                         <input type="text" name="name" class="form-control" value="{{$category->name}}" id="exampleInputEmail1">
17                     </div>
18                     @error('name')
19                         <p class="text-danger"> {{ $message }}</p>
20                     @enderror()
21                 </div>
22                 <div class="card-footer">
23                     <button type="submit" class="btn btn-primary">Submit</button>
24                 </div>
25             </form>
26
27         </div>
28         <!-- /.card -->
29
30     </div>
31
32     @endsection
33

```

- دالة الـ **store** تقوم بالتخزين في ال **database** في جدول التصنيفات
- دالة الـ **update** تقوم بالتعديل على التصنيف الموجود في ال **database**
- دالة الـ **destroy** تقوم بحذف التصنيف من ال **database**

Product Controller

```
● ○ ●  
1  <?php  
2  
3  namespace App\Http\Controllers\Admin;  
4  
5  use App\Http\Controllers\Controller;  
6  use App\Models\Category;  
7  use App\Models\Product;  
8  use App\Http\Requests\StoreProductRequest;  
9  use App\Http\Requests\UpdateProductRequest;  
10 use App\Models\User;  
11 use Illuminate\Support\Facades\File;  
12 use RealRashid\SweetAlert\Facades\Alert;  
13  
14 use function PHPUnit\Framework\fileExists;  
15  
16 class ProductController extends Controller  
17 {  
18     public function index()  
19     {  
20         $products = Product::paginate(5);  
21         return view('admin.Pages.product.index', compact('products'));  
22     }  
23  
24     public function create()  
25     {  
26         $categories = Category::get();  
27         $users = User::get();  
28         return view('admin.Pages.product.create', compact('categories', 'users'));  
29     }  
30  
31     public function store(StoreProductRequest $request)  
32     {  
33         $data = $request->validated();  
34         if ($request->file('image')) {  
35             $file = $request->file('image');  
36             $filename = date('YmdHi') . $file->getClientOriginalName();  
37             $file->move(public_path('images'), $filename);  
38             $data['image'] = $filename;  
39         }  
40         $data['user_id'] = $request->user_id;  
41         Product::create($data);  
42         Alert::success('Success Title', 'Product Was Created');  
43         return redirect()->back();  
44     }  
}
```

```
1  public function edit(Product $product)
2  {
3      $categories = Category::get();
4      $users = User::get();
5      return view('admin.Pages.product.edit', compact('categories', 'users', 'product'));
6  }
7
8  public function update(UpdateProductRequest $request, Product $product)
9  {
10     $data = $request->all();
11     if ($request->file('image')) {
12         if (fileExists(public_path('images/' . $product->image))) {
13             File::delete(public_path('images/' . $product->image));
14         }
15         $file = $request->file('image');
16         $filename = date('YmdHi') . $file->getClientOriginalName();
17         $file->move(public_path('images'), $filename);
18         $data['image'] = $filename;
19     }
20     $data['is_approved'] = isset($data['is_approved']) ? $data['is_approved'] : 0;
21     $product->update($data);
22     Alert::success('Success Title', 'Product Was Updated');
23     return redirect(route('admin.product.index'));
24 }
25
26 public function destroy(Product $product)
27 {
28     $product->delete();
29     Alert::success('Success Title', 'Product Was Deleted');
30     return redirect(route('admin.product.index'));
31 }
32 }
```

• تفتح جداول المنتجات أو الأفلام : Index

```
1 @extends('admin.index')
2
3 @section('content')
4     <div class="card">
5         <div class="card-header">
6             <h3 class="card-title">Products Table</h3>
7
8         <a href="{{route('admin.product.create')}}" class="btn btn-success float-right ">
9             Create New Product
10        </a>
11    </div>
12    <!-- /.card-header -->
13    <div class="card-body">
14        <div id="example2_wrapper" class="dataTables_wrapper dt-bootstrap4">
15            <div class="row"><div class="col-sm-12 col-md-6"></div>
16                <div class="col-sm-12 col-md-6"></div>
17            </div>
18            <div class="row">
19                <div class="col-sm-12">
20                    <table id="example2" class="table table-bordered table-hover dataTable dtr-inline" role="grid" aria-describedby="example2_info">
21                        <thead>
22                            <tr role="row">
23                                <th class="sorting_asc" tabindex="0" aria-controls="example2" rowspan="1" colspan="1" aria-sort="ascending" aria-label="Rendering engine: activate to sort column descending">#</th>
24                                <th class="sorting" tabindex="0" aria-controls="example2" rowspan="1" colspan="1" aria-label="Browser: activate to sort column ascending">Name</th>
25                                <th class="sorting" tabindex="0" aria-controls="example2" rowspan="1" colspan="1" aria-label="Browser: activate to sort column ascending">Is Approved</th>
26                                <th class="sorting" tabindex="0" aria-controls="example2" rowspan="1" colspan="1" aria-label="Browser: activate to sort column ascending">Category</th>
27                                <th class="sorting" tabindex="0" aria-controls="example2" rowspan="1" colspan="1" aria-label="Browser: activate to sort column ascending">User</th>
28                                <th class="sorting" tabindex="0" aria-controls="example2" rowspan="1" colspan="1" aria-label="Browser: activate to sort column ascending">Image</th>
29
30                                <th class="sorting" tabindex="0" aria-controls="example2" rowspan="1" colspan="1" aria-label="Browser: activate to sort column ascending">Edit</th>
31                                <th class="sorting" tabindex="0" aria-controls="example2" rowspan="1" colspan="1" aria-label="Browser: activate to sort column ascending">Delete</th>
32
33
34                            </thead>
35                            <tbody>
36                                @foreach($products as $product)
37                                    <tr role="row" class="odd">
38                                        <td tabindex="0" class="sorting_1">{{ $product->id }}</td>
39                                        <td>{{ $product->name }}</td>
40                                        @if($product->isApproved == false)
41                                            <td class="text-danger">Denied</td>
42                                        @else
43                                            <td class="text-success">Approved</td>
44                                        @endif
45                                        <td>{{ $product->category->name ?? "" }}</td>
46                                        <td>{{ $product->user->name ?? "" }}</td>
47                                        <td>
48                                            
49                                        </td>
50                                        <td>
51                                            <form method="get" action="{{ route('admin.product.edit', $product) }}>
52                                                @csrf
53                                                <button type="submit" class="btn btn-success">
54                                                    <i class="fa fa-edit">/</i>
55                                                </button>
56                                            </form>
57                                        </td>
58                                        <td>
59                                            <form method="POST" action="{{ route('admin.product.destroy', $product) }}>
60                                                @csrf
61                                                @method('DELETE')
62                                                <button type="submit" class="btn btn-danger">
63                                                    <i class="fa fa-trash">/</i>
64                                                </button>
65                                            </form>
66                                        </td>
67                                    </tr>
68                                @endforeach
69
70                            </tbody>
71
72                    </table>
73                </div>
74            <div class="row">
75                <div class="col-sm-12 col-md-5">
76                    <div class="dataTables_info" id="example2_info" role="status" aria-live="polite">Showing 1 to 10 of 57 entries</div>
77                </div>
78                <div class="col-sm-12 col-md-7">
79                    <div class="dataTables_paginate paging_simple_numbers" id="example2_paginate">
80                        {{ $products->links() }}
81                    </div>
82                </div>
83            </div>
84        </div>
85    </div>
86    <!-- /.card-body -->
87 </div>
88 @endsection
89
```

● دالة **create** تقوم بفتح صفحة إنشاء المنتج

```
1  @extends('admin.index')
2
3  @section('content')
4      <div class="col-md-12">
5          <div class="card card-primary">
6              <div class="card-header">
7                  <h3 class="card-title">Create New Product</h3>
8              </div>
9
10             <form method="POST" action="{{route('admin.product.store')}}" enctype="multipart/form-data">
11                 @csrf
12                 <div class="card-body">
13                     <div class="form-group">
14                         <label for="name">Name</label>
15                         <input type="text" name="name" class="form-control" id="name">
16                         @error('name')
17                             <p class="text-danger"> {{ $message }} </p>
18                         @enderror()
19
20
21                         <label for="image">Image</label>
22                         <input type="file" name="image" class="form-control" id="image">
23                         @error('image')
24                             <p class="text-danger"> {{ $message }} </p>
25                         @enderror()
26
27                         <div class="form-group" >
28                             <textarea class="ckeditor form-control" name="description" ></textarea>
29                             @error('description')
30                                 <p class="text-danger"> {{ $message }} </p>
31                             @enderror()
32                         </div>
33
34                         <div class="form-check">
35                             <input type="checkbox" class="form-check-input" name="is_approved" value="1" id="exampleCheck1">
36                             <label class="form-check-label" for="exampleCheck1">Is Approved</label>
37                         </div>
38
39                         <div class="form-group">
40                             <label>Category</label>
41                             <select class="form-control" name="category_id">
42                                 @foreach($categories as $category)
43                                     <option value="{{ $category->id }}">{{ $category->name }}</option>
44                                 @endforeach()
45                             </select>
46                         </div>
47                         @error('category_id')
48                             <p class="text-danger"> {{ $message }} </p>
49                         @enderror()
50
51                         <div class="form-group">
52                             <label>User</label>
53                             <select class="form-control" name="user_id">
54                                 @foreach($users as $user)
55                                     <option value="{{ $user->id }}">{{ $user->name }}</option>
56                                 @endforeach()
57                             </select>
58
59                         </div>
60
61                         </div>
62                         @error('user_id')
63                             <p class="text-danger"> {{ $message }} </p>
64                         @enderror()
65                     </div>
66
67                     <div class="card-footer">
68                         <button type="submit" class="btn btn-primary">Submit</button>
69                     </div>
70                 </form>
71
72             </div>
73             <!-- /.card -->
74
75         </div>
76
77     @endsection
78
79     <script>
80         $(document).ready(function() {
81             $('#summernote').summernote();
82         });
83     </script>
84
85
```

● دالة الـ **edit** تقوم بفتح صفحة تعديل المنتج

```
1 @extends('admin.index')
2
3 @section('content')
4     <div class="col-md-12">
5         <div class="card card-primary">
6             <div class="card-header">
7                 <h3 class="card-title">Edit Product</h3>
8             </div>
9
10            <form method="POST" action="{{route('admin.product.update', $product)}}" enctype="multipart/form-data">
11                @csrf
12                @method('PUT')
13                <div class="card-body">
14                    <div class="form-group">
15                        <label for="exampleInputEmail1">Name</label>
16                        <input type="text" name="name" value="{{($product->name)}}" class="form-control" id="exampleInputEmail1">
17                        @error('name')
18                            <p class="text-danger"> {{($message)}}</p>
19                        @enderror()
20
21
22                        <label for="image">Image</label>
23                        <input type="file" name="image" class="form-control" id="image">
24                        @error('image')
25                            <p class="text-danger"> {{($message)}}</p>
26                        @enderror()
27
28                    <div class="form-group">
29                        <textare class="ckeditor form-control" name="description">{{($product->description)}}</textare>
30                        @error('description')
31                            <p class="text-danger"> {{($message)}}</p>
32                        @enderror()
33
34
35
36                    <div class="form-group">
37                        <input type="checkbox" class="form-check-input" name="is_approved" {{($product->is_approved == true ? "checked" : "")}} value="1" id="exampleCheck1">
38                        <label class="form-check-label" for="exampleCheck1">Is Approved</label>
39
40
41                    <div class="form-group">
42                        <label>Category</label>
43                        <select class="form-control" name="category_id" value="{{($product->category_id)}}>
44                            @foreach($categories as $category)
45                                <option value="{{($category->id)}}>{{($category->name)}}</option>
46                            @endforeach()
47                        </select>
48
49                        @error('category_id')
50                            <p class="text-danger"> {{($message)}}</p>
51                        @enderror()
52
53                    <div class="form-group">
54                        <label>User</label>
55                        <select class="form-control" name="user_id" value="{{($product->user_id)}}>
56                            @foreach($users as $user)
57                                <option value="{{($user->id)}}>{{($user->name)}}</option>
58                            @endforeach()
59                        </select>
60
61
62
63                    </div>
64                    @error('user_id')
65                        <p class="text-danger"> {{($message)}}</p>
66                    @enderror()
67
68                    <div class="card-footer">
69                        <button type="submit" class="btn btn-primary">Submit</button>
70                    </div>
71                </form>
72
73            </div>
74            <!-- /.card -->
75
76        </div>
77
78    @endsection
79
```

- دالة الـ **store** تقوم بالتخزين في ال **database** في جدول المنتجات
- دالة الـ **update** تقوم بالتعديل على المنتج الموجود في ال **database**
- دالة الـ **destroy** تقوم بحذف المنتج من ال **database**

Review Controller

```
1 <?php
2
3 namespace App\Http\Controllers\Admin;
4
5 use App\Models\Product;
6 use App\Models\User;
7 use App\Models\Review;
8 use App\Http\Controllers\Controller;
9 use App\Http\Requests\StoreReviewRequest;
10 use App\Http\Requests\UpdateReviewRequest;
11 use RealRashid\SweetAlert\Facades\Alert;
12
13 class ReviewController extends Controller
14 {
15     public function index()
16     {
17         $reviews = Review::paginate(5);
18         return view('admin.Pages.review.index', compact('reviews'));
19     }
20
21     public function create()
22     {
23         $products = Product::get();
24         $users = User::get();
25         return view('admin.Pages.review.create', compact('products', 'users'));
26     }
27
28     public function store(StoreReviewRequest $request)
29     {
30         Review::create($request->all());
31         Alert::success('Success Title', 'Review Was Created');
32         return redirect(route('admin.review.index'));
33     }
34
35     public function edit(Review $review)
36     {
37         $products = Product::get();
38         $users = User::get();
39         return view('admin.Pages.review.edit', compact('products', 'users', 'review'));
40     }
41
42     public function update(UpdateReviewRequest $request, Review $review)
43     {
44         $review->update($request->validated());
45         Alert::success('Success Title', 'Review Was Updated');
46         return redirect(route('admin.review.index'));
47     }
48
49     public function destroy(Review $review)
50     {
51         $review->delete();
52         Alert::success('Success Title', 'Review Was Deleted');
53         return redirect(route('admin.review.index'));
54     }
55 }
```

```

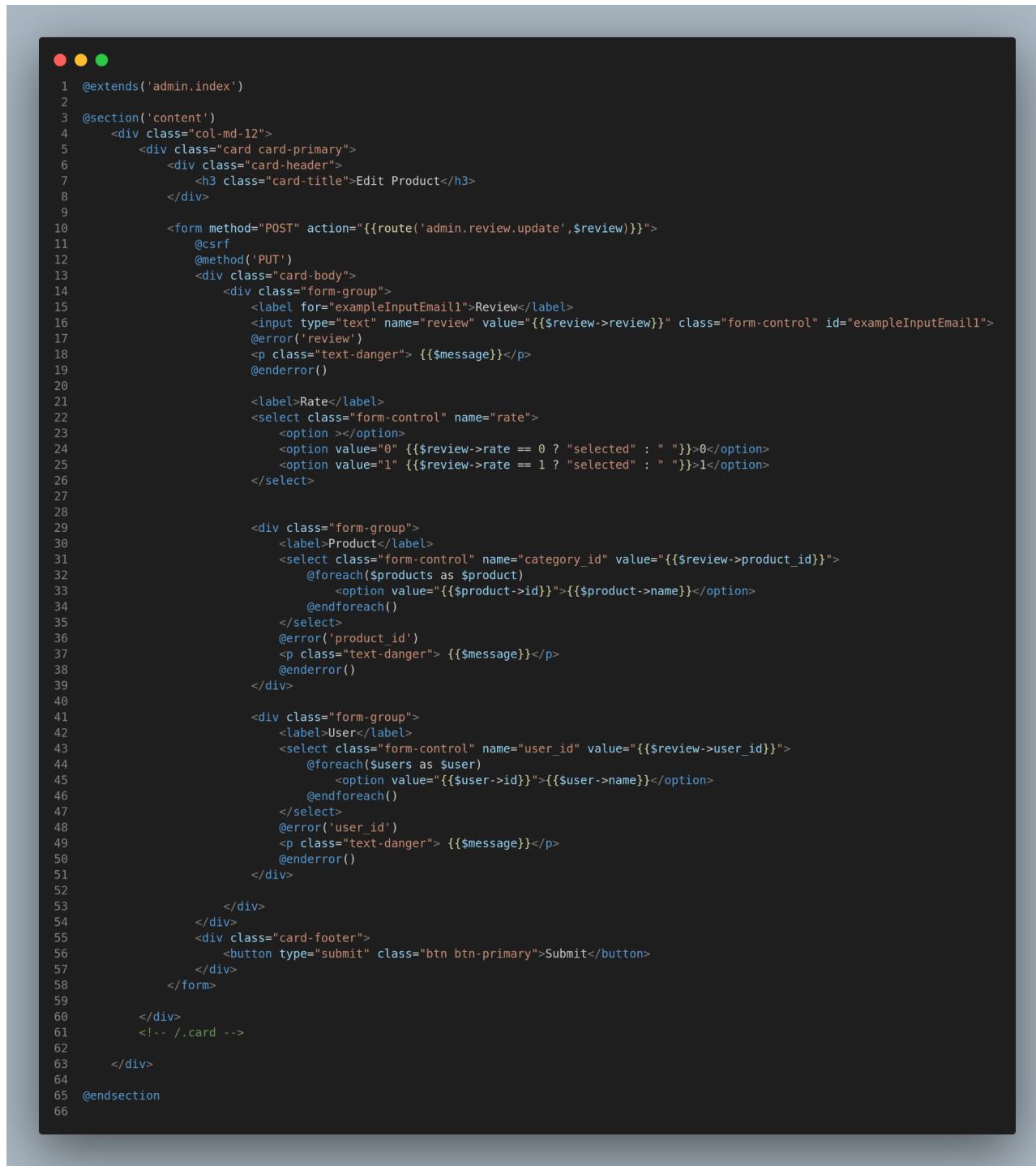
1 @extends('admin.index')
2
3 @section('content')
4     <div class="card">
5         <div class="card-header">
6             <h3 class="card-title">Categories</h3>
7             <a href="{{route('admin.review.create')}}" class="btn btn-success float-right"> Create New Review</a>
8         </div>
9         <!-- /.card-header -->
10        <div class="card-body">
11            <div id="example2_wrapper" class="dataTables_wrapper dt-bootstrap4">
12                <div class="row"><div class="col-sm-12 col-md-6"></div>
13                <div class="col-sm-12 col-md-6"></div>
14            </div>
15            <div class="row">
16                <div class="col-sm-12">
17                    <table id="example2" class="table table-bordered table-hover dataTable dtr-inline" role="grid" aria-describedby="example2_info">
18                        <thead>
19                            <tr role="row">
20                                <th class="sorting asc" tabindex="0" aria-controls="example2" rowspan="1" colspan="1" aria-sort="ascending" aria-label="Rendering engine: activate to sort column descending">#</th>
21                                <th class="sorting" tabindex="0" aria-controls="example2" rowspan="1" colspan="1" aria-label="Browser: activate to sort column ascending">Review</th>
22                                <th class="sorting" tabindex="0" aria-controls="example2" rowspan="1" colspan="1" aria-label="Browser: activate to sort column ascending">Product</th>
23                                <th class="sorting" tabindex="0" aria-controls="example2" rowspan="1" colspan="1" aria-label="Browser: activate to sort column ascending">User</th>
24                                <th class="sorting" tabindex="0" aria-controls="example2" rowspan="1" colspan="1" aria-label="Browser: activate to sort column ascending">Edit</th>
25                                <th class="sorting" tabindex="0" aria-controls="example2" rowspan="1" colspan="1" aria-label="Browser: activate to sort column ascending">Delete</th>
26
27
28                        </thead>
29                        <tbody>
30                            @foreach($reviews as $review)
31                                <tr role="row" class="odd">
32                                    <td tabindex="0" class="sorting_1" rowspan="1">{{ $review->id }}
33                                    <td>{{ $review->review }}</td>
34
35                                    <td>{{ $review->product->name }}</td>
36                                    <td>{{ $review->user->name }}</td>
37
38                                    <td>
39                                        <form method="get" action="{{ route('admin.review.edit', $review) }}">
40                                            @csrf
41                                            <button type="submit" class="btn btn-success">
42                                                <i class="fa fa-edit"></i>
43                                            </button>
44                                        </form>
45                                    </td>
46                                    <td>
47                                        <form method="POST" action="{{ route('admin.review.destroy', $review) }}">
48                                            @csrf
49                                            @method('DELETE')
50                                            <button type="submit" class="btn btn-danger">
51                                                <i class="fa fa-trash"></i>
52                                            </button>
53                                        </form>
54                                    </td>
55                                </tr>
56                            @endforeach
57
58                        </tbody>
59
60                    </table>
61                </div>
62            <div class="row">
63                <div class="col-sm-12 col-md-5">
64                    <div class="dataTables_info" id="example2_info" role="status" aria-live="polite">Showing 1 to 10 of 57 entries</div>
65                </div>
66                <div class="col-sm-12 col-md-7">
67                    <div class="dataTables_paginate paging_simple_numbers" id="example2_paginate">
68                        {{ $reviews->links() }}
69                    </div>
70                </div>
71            </div>
72        </div>
73    </div>
74    <!-- /.card-body -->
75 </div>
76 </div>
77 @endsection
78

```

● دالة **create** تقوم بفتح صفحة إنشاء التعليق

```
1  @extends('admin.index')
2
3  @section('content')
4      <div class="col-md-12">
5          <div class="card card-primary">
6              <div class="card-header">
7                  <h3 class="card-title">Create New Review</h3>
8              </div>
9
10             <form method="POST" action="{{route('admin.review.store')}}">
11                 @csrf
12                 <div class="card-body">
13                     <div class="form-group">
14                         <label for="exampleInputEmail1">Review</label>
15                         <input type="text" name="review" class="form-control" id="exampleInputEmail1">
16                         @error('name')
17                             <p class="text-danger"> {{ $message }} </p>
18                         @enderror()
19
20                         <label>Rate</label>
21                         <select class="form-control" name="rate">
22                             <option ></option>
23                             <option value="0">0</option>
24                             <option value="1">1</option>
25                         </select>
26
27                         <div class="form-group">
28                             <label>Product</label>
29                             <select class="form-control" name="product_id">
30                                 @foreach($products as $product)
31                                     <option value="{{ $product->id }}>{{ $product->name }}</option>
32                                 @endforeach()
33                             </select>
34                             @error('product_id')
35                             <p class="text-danger"> {{ $message }} </p>
36                             @enderror()
37                         </div>
38
39                         <div class="form-group">
40                             <label>User</label>
41                             <select class="form-control" name="user_id">
42                                 @foreach($users as $user)
43                                     <option value="{{ $user->id }}>{{ $user->name }}</option>
44                                 @endforeach()
45                             </select>
46                             @error('user_id')
47                             <p class="text-danger"> {{ $message }} </p>
48                             @enderror()
49                         </div>
50
51                 </div>
52
53             </div>
54
55             <div class="card-footer">
56                 <button type="submit" class="btn btn-primary">Submit</button>
57             </div>
58         </form>
59     </div>
60
61     </div>
62     <!-- /.card -->
63
64 </div>
65 @endsection
66
```

- دالة الـ **edit** تقوم بفتح صفحة تعديل التعليق



```

1  @extends('admin.index')
2
3  @section('content')
4      <div class="col-md-12">
5          <div class="card card-primary">
6              <div class="card-header">
7                  <h3 class="card-title">Edit Product</h3>
8              </div>
9
10             <form method="POST" action="{{route('admin.review.update',$review)}}>
11                 @csrf
12                 @method('PUT')
13                 <div class="card-body">
14                     <div class="form-group">
15                         <label for="exampleInputEmail1">Review</label>
16                         <input type="text" name="review" value="{{$review->review}}" class="form-control" id="exampleInputEmail1">
17                         @error('review')
18                             <p class="text-danger"> {{$message}}</p>
19                         @enderror()
20
21                         <label>Rate</label>
22                         <select class="form-control" name="rate">
23                             <option ></option>
24                             <option value="0" {{$review->rate == 0 ? "selected" : ""}}>0</option>
25                             <option value="1" {{$review->rate == 1 ? "selected" : ""}}>1</option>
26                         </select>
27
28
29                         <div class="form-group">
30                             <label>Product</label>
31                             <select class="form-control" name="category_id" value="{{$review->product_id}}>
32                                 @foreach($products as $product)
33                                     <option value="{{$product->id}}>{{$product->name}}</option>
34                                 @endforeach()
35                             </select>
36                             @error('product_id')
37                             <p class="text-danger"> {{$message}}</p>
38                             @enderror()
39                         </div>
40
41                         <div class="form-group">
42                             <label>User</label>
43                             <select class="form-control" name="user_id" value="{{$review->user_id}}>
44                                 @foreach($users as $user)
45                                     <option value="{{$user->id}}>{{$user->name}}</option>
46                                 @endforeach()
47                             </select>
48                             @error('user_id')
49                             <p class="text-danger"> {{$message}}</p>
50                             @enderror()
51                         </div>
52
53                         </div>
54                     </div>
55                     <div class="card-footer">
56                         <button type="submit" class="btn btn-primary">Submit</button>
57                     </div>
58                 </form>
59
60             </div>
61             <!-- /.card -->
62
63         </div>
64
65     @endsection
66

```

- دالة الـ **store** تقوم بالتخزين في ال **database** في جدول التعليقات
- دالة الـ **update** تقوم بالتعديل على التعليق الموجود في ال **database**
- دالة الـ **destroy** تقوم بحذف التعليق من ال **database**

User Controller

```
1 <?php
2
3 namespace App\Http\Controllers\Admin;
4
5 use App\Http\Controllers\Controller;
6
7 use App\Http\Requests\StoreUserRequest;
8 use App\Http\Requests\UpdateUserRequest;
9 use App\Models\User;
10 use Illuminate\Support\Facades\File;
11 use Illuminate\Support\Facades\Hash;
12 use RealRashid\SweetAlert\Facades\Alert;
13
14 use function PHPUnit\Framework\fileExists;
15
16 class UserController extends Controller
17 {
18     public function index()
19     {
20         $users = User::paginate(5);
21         return view('admin.Pages.user.index', compact('users'));
22     }
23
24     public function create()
25     {
26         return view('admin.Pages.user.create');
27     }
28
29     public function store(StoreUserRequest $request)
30     {
31         if ($request->file('image')) {
32             $file = $request->file('image');
33             $filename = date('YmdHi') . $file->getClientOriginalName();
34             $file->move(public_path('images'), $filename);
35         }
36
37         $user = User::create([
38             'name' => $request->name,
39             'email' => $request->email,
40             'password' => Hash::make($request->password),
41             'image' => $filename ?? '',
42         ]);
43         if ($request->type == 'viewer') {
44             $user->attachRole('viewer');
45         } else if ($request->type == 'editor') {
46             $user->attachRole('editor');
47         } else if ($request->type == 'admin') {
48             $user->attachRole('admin');
49         }
50         Alert::success('Success Title', 'User Was Created');
51         return redirect(route('admin.user.index'));
52     }
}
```

```
1
2     public function edit(User $user)
3     {
4         return view('admin.Pages.user.edit', compact('user'));
5     }
6
7     public function update(UpdateUserRequest $request, User $user)
8     {
9
10        $data = $request->validated();
11        if ($request->file('image')) {
12            if (fileExists(public_path('images/' . $user->image))) {
13                File::delete(public_path('images/' . $user->image));
14            }
15            $file = $request->file('image');
16            $filename = date('YmdHi') . $file->getClientOriginalName();
17            $file->move(public_path('images'), $filename);
18            $data['image'] = $filename;
19        }
20        $user->update($data);
21        if ($request->type == 'viewer') {
22            $user->attachRole('viewer');
23        } else if ($request->type == 'editor') {
24            $user->attachRole('editor');
25        } else if ($request->type == 'admin') {
26            $user->attachRole('admin');
27        }
28        Alert::success('Success Title', 'User Was Updated');
29        return redirect(route('admin.user.index'));
30    }
31
32    public function destroy(User $user)
33    {
34        $user->delete();
35        Alert::success('Success Title', 'User Was Deleted');
36        return redirect(route('admin.user.index'));
37    }
38 }
```

• Index : تفاصيل المستخدمين

```
1 @extends('admin.index')
2
3 @section('content')
4 <div class="card">
5     <div class="card-header">
6         <h3 class="card-title">Users Table</h3>
7         <a href="{{route('admin.user.create')}}" class="btn btn-success float-right"> Create New User</a>
8     </div>
9     <!-- /.card-header -->
10    <div class="card-body">
11        <div id="example2_wrapper" class="dataTables_wrapper dt-bootstrap4">
12            <div class="row"><div class="col-sm-12 col-md-6"></div>
13            <div class="col-sm-12 col-md-6"></div>
14        </div>
15        <div class="row">
16            <div class="col-sm-12">
17                <table id="example2" class="table table-bordered table-hover dataTable dtr-inline" role="grid" aria-describedby="example2_info">
18                    <thead>
19                        <tr role="row">
20                            <th class="sorting_asc" tabindex="0" aria-controls="example2" rowspan="1" colspan="1" aria-sort="ascending" aria-label="Rendering engine: activate to sort column descending">#</th>
21                            <th class="sorting" tabindex="0" aria-controls="example2" rowspan="1" colspan="1" aria-label="Browser: activate to sort column ascending">Name</th>
22                            <th class="sorting" tabindex="0" aria-controls="example2" rowspan="1" colspan="1" aria-label="Browser: activate to sort column ascending">Email</th>
23                            <th class="sorting" tabindex="0" aria-controls="example2" rowspan="1" colspan="1" aria-label="Browser: activate to sort column ascending">Phone</th>
24                            <th class="sorting" tabindex="0" aria-controls="example2" rowspan="1" colspan="1" aria-label="Browser: activate to sort column ascending">Type</th>
25                            <th class="sorting" tabindex="0" aria-controls="example2" rowspan="1" colspan="1" aria-label="Browser: activate to sort column ascending">Image</th>
26                            <th class="sorting" tabindex="0" aria-controls="example2" rowspan="1" colspan="1" aria-label="Browser: activate to sort column ascending">Edit</th>
27                            <th class="sorting" tabindex="0" aria-controls="example2" rowspan="1" colspan="1" aria-label="Browser: activate to sort column ascending">Delete</th>
28
29
30                    </thead>
31                    <tbody>
32                        @foreach($users as $user)
33                            <tr role="row" class="odd">
34                                <td tabindex="0" class="sorting_1" rowspan="1">{{ $user->id }}
35                                <td>{{ $user->name }}</td>
36                                <td>{{ $user->email }}</td>
37                                <td>{{ $user->phone }}</td>
38                                <td>{{ $user->roles()->first()->name ?? "no-role" }}</td>
39                                <td>
40                                    
41                                </td>
42                                <td>
43                                    <form method="get" action="{{route('admin.user.edit',$user)}}>
44                                        @csrf
45                                        <button type="submit" class="btn btn-success">
46                                            <i class="fa fa-edit"></i>
47                                        </button>
48                                    </form>
49                                </td>
50                                <td>
51                                    <form method="POST" action="{{route('admin.user.destroy',$user)}}>
52                                        @csrf
53                                        @method('DELETE')
54                                        <button type="submit" class="btn btn-danger">
55                                            <i class="fa fa-trash"></i>
56                                        </button>
57                                    </form>
58                                </td>
59                            </tr>
60                        @endforeach
61
62                    </tbody>
63
64                </table>
65            </div>
66        <div class="row">
67            <div class="col-sm-12 col-md-5">
68                <div class="dataTables_info" id="example2_info" role="status" aria-live="polite">Showing 1 to 10 of 57 entries</div>
69            </div>
70            <div class="col-sm-12 col-md-7">
71                <div class="dataTables_paginate paging_simple_numbers" id="example2_paginate">
72                    {{ $users->links() }}
73                </div>
74            </div>
75        </div>
76    </div>
77    <!-- /.card-body -->
78 </div>
79 <!-- /.card -->
80 @endsection
81 @endsection
```

● دالة **create** تقوم بفتح صفحة إنشاء المستخدمين

```
1  @extends('admin.index')
2
3  @section('content')
4      <div class="col-md-12">
5          <div class="card card-primary">
6              <div class="card-header">
7                  <h3 class="card-title">Create New User</h3>
8              </div>
9
10             <form method="POST" action="{{route('admin.user.store')}}" enctype="multipart/form-data">
11                 @csrf
12                 <div class="card-body">
13                     <div class="form-group">
14                         <label for="name">Name</label>
15                         <input type="text" name="name" class="form-control" id="name">
16
17                         <label for="email">Email</label>
18                         <input type="text" name="email" class="form-control" id="email">
19
20                         <label for="password">Password</label>
21                         <input type="password" name="password" class="form-control" id="password">
22
23                         <label for="phone">Phone</label>
24                         <input type="text" name="phone" class="form-control" id="phone">
25
26                         <label for="image">Image</label>
27                         <input type="file" name="image" class="form-control" id="image">
28
29                         <label>Type</label>
30                         <select class="form-control" name="type">
31                             <option>Choose</option>
32                             <option value="editor">editor</option>
33                             <option value="viewer">viewer</option>
34                             <option value="admin">admin</option>
35                         </select>
36                     </div>
37                     @error('name')
38                         <p class="text-danger"> {{ $message }}</p>
39                     @enderror()
40                 </div>
41                 <div class="card-footer">
42                     <button type="submit" class="btn btn-primary">Submit</button>
43                 </div>
44             </form>
45
46         </div>
47         <!-- /.card -->
48
49     </div>
50
51     @endsection
52
```

- دالة الـ **edit** تقوم بفتح صفحة تعديل مستخدم

```

● ● ●

1 @extends('admin.index')
2
3 @section('content')
4     <div class="col-md-12">
5         <div class="card card-primary">
6             <div class="card-header">
7                 <h3 class="card-title">Edit User</h3>
8             </div>
9
10            <form method="POST" action="{{route('admin.user.update',$user)}}" enctype="multipart/form-data">
11                @csrf
12                @method('PUT')
13                <div class="card-body">
14                    <div class="form-group">
15                        <label for="exampleInputEmail1">Name</label>
16                        <input type="text" name="name" class="form-control" value="{{$user->name}}" id="exampleInputEmail1">
17
18
19                        <label for="exampleInputEmail1">Email</label>
20                        <input type="text" name="email" class="form-control" value="{{$user->email}}" id="exampleInputEmail1">
21
22                        <label for="exampleInputEmail1">Phone</label>
23                        <input type="text" name="phone" class="form-control" value="{{$user->phone}}" id="exampleInputEmail1">
24
25                        <label for="image">Image</label>
26                        <input type="file" name="image" class="form-control" id="image">
27
28                        <label>Type</label>
29                        <select class="form-control" name="type">
30                            <option>Choose</option>
31                            <option value="editor">editor</option>
32                            <option value="viewer">viewer</option>
33                            <option value="admin">admin</option>
34                        </select>
35
36                    </div>
37                    @error('name')
38                        <p class="text-danger"> {{$message}}</p>
39                    @enderror()
40                </div>
41                <div class="card-footer">
42                    <button type="submit" class="btn btn-primary">Submit</button>
43                </div>
44            </form>
45
46        </div>
47        <!-- /.card -->
48
49    </div>
50
51 @endsection
52

```

- دالة الـ **store** تقوم بالتخزين في ال database في جدول المستخدمين
- دالة الـ **update** تقوم بالتعديل على المستخدم الموجود في ال database
- دالة الـ **destroy** تقوم بحذف المستخدم من ال database

Front Auth Controller

```
1  <?php
2
3  namespace App\Http\Controllers\Front;
4
5  use App\Http\Controllers\Controller;
6  use App\Http\Requests\LoginRequest;
7  use App\Http\Requests\RegisterRequest;
8  use App\Models\User;
9  use Illuminate\Support\Facades\Auth;
10 use Illuminate\Support\Facades\Hash;
11 use RealRashid\SweetAlert\Facades\Alert;
12
13 class AuthController extends Controller
14 {
15     public function registerPage()
16     {
17         return view('front.pages.auth.register');
18     }
19
20     public function register(RegisterRequest $request)
21     {
22         $user = User::create([
23             'name' => $request->name,
24             'email' => $request->email,
25             'password' => Hash::make($request->password),
26         ]);
27
28         if($request->type == 'viewer')
29         {
30             $user->attachRole('viewer');
31         }
32         else if($request->type == 'editor')
33         {
34             $user->attachRole('editor');
35         }
36         Alert::toast('Thank you for registering!!!','success');
37         return redirect(route('home.loginPage'));
38     }
}
```

```
1
2
3     public function loginPage()
4     {
5         return view('front.pages.auth.login');
6     }
7
8     public function login(LoginRequest $request)
9     {
10        $data = $request->only(['email','password']);
11        if(Auth::attempt($data))
12        {
13            Alert::toast('Thank you for Login!!!','success');
14            return redirect(route('home.index'));
15        }
16        Alert::toast('Email or Password is Wrong!!!','error');
17        return redirect()->back();
18    }
19
20    public function logout()
21    {
22        Auth::logout();
23        Alert::toast('Thank you for Watching movies with us!!!','success');
24        return redirect(route('home.loginPage'));
25    }
26}
27
```

● دالة الـ registerPage تقوم بفتح صفحة انشاء حساب جديد

```
● ● ●
1 @extends('front.master')
2
3 @section('css')
4     <link rel="stylesheet" href="{{ asset('css/forms.css') }}>
5 @endsection
6
7 @section('content')
8     <div class="form-body">
9         <div class="row" style="width: 60%; margin:0 auto">
10            <div class="form-holder">
11                <div class="form-content">
12                    <div class="form-items">
13                        <h3>Register Today</h3>
14                        <p>Fill in the data below.</p>
15                        <form class="requires-validation" novalidate method="POST" action="{{route('home.register')}}">
16                            @csrf
17                            <div class="col-md-12">
18                                <input class="form-control" type="text" name="name" placeholder="Full Name"
19                                    required>
20                                @error('name')
21                                    <div class="invalid-feedback">{{ $message }}</div>
22                                @enderror
23                            </div>
24
25                            <div class="col-md-12">
26                                <input class="form-control" type="email" name="email"
27                                    placeholder="E-mail Address" required>
28                                @error('email')
29                                    <div class="invalid-feedback">{{ $message }}</div>
30                                @enderror
31                            </div>
32
33                            <div class="col-md-12">
34                                <input class="form-control" type="password" name="password"
35                                    placeholder="Password" required>
36                                @error('password')
37                                    <div class="invalid-feedback">{{ $message }}</div>
38                                @enderror
39                            </div>
40
41                            <div class="col-md-12">
42                                <input class="form-control" type="password" name="password_confirmation"
43                                    placeholder="Password" required>
44                                @error('confirm_password')
45                                    <div class="invalid-feedback">{{ $message }}</div>
46                                @enderror
47                            </div>
48
49                            <div class="col-md-12 mb-5">
50                                <select name="type" class="form-select mt-3" required>
51                                    <option selected disabled value="0">Type</option>
52                                    <option value="viewer">Viewer</option>
53                                    <option value="editor">Editor</option>
54                                </select>
55                                @error('type')
56                                    <div class="invalid-feedback">{{ $message }}</div>
57                                @enderror
58                                <div class="invalid-feedback" style="visibility: hidden"> asdfsaf </div>
59                            </div>
60
61
62                            <div class="col-md-12">
63                                <button type="submit" class="btn btn-primary">Register</button>
64                            </div>
65                        </form>
66                    </div>
67                </div>
68            </div>
69        </div>
70    @endsection
71
72
73
74 @section('script')
75     <script></script>
76 @endsection
77
```

● دالة الـ **register** تستقبل بيانات المستخدم وتنشئ له حساب جديد

● دالة الـ **loginPage** تفتح صفحة تسجيل دخول المستخدمين

```
1 @extends('front.master')
2
3 @section('css')
4     <link rel="stylesheet" href="{{ asset('css/forms.css') }}>
5 @endsection
6
7 @section('content')
8     <div class="form-body">
9         <div class="row" style="width: 60%; margin:0 auto">
10            <div class="form-holder">
11                <div class="form-content">
12                    <div class="form-items">
13                        <h3>Login</h3>
14                        <p>Fill in the data below.</p>
15                        <form class="requires-validation" novalidate method="POST" action="{{ route('home.login') }}>
16                            @csrf
17                            <div class="col-md-12">
18                                <input class="form-control" type="email" name="email"
19                                       placeholder="E-mail Address" required>
20                                @error('email')
21                                    <div class="invalid-feedback">{{ $message }}</div>
22                                @enderror
23                            </div>
24
25                            <div class="col-md-12">
26                                <input class="form-control" type="password" name="password"
27                                       placeholder="Password" required>
28                                @error('password')
29                                    <div class="invalid-feedback">{{ $message }}</div>
30                                @enderror
31                                <div class="invalid-feedback" style="visibility: hidden"> asdfsaf </div>
32                            </div>
33
34                            <div class="col-md-12">
35                                <button type="submit" class="btn btn-primary">Login</button>
36                            </div>
37                        </form>
38                    </div>
39                </div>
40            </div>
41        </div>
42    </div>
43 @endsection
44
45
46 @section('script')
47     <script></script>
48 @endsection
49
```

● دالة الـ **login** تستقبل بيانات المستخدم وإذا كان له حساب في الموقع تسمح له بالدخول. وإلا تظهر

رسالة خطأ.

● دالة الـ **logout** تقوم بتسجيل الخروج

Front Editor Controller

```
1  <?php
2
3  namespace App\Http\Controllers\Front;
4
5  use App\Http\Controllers\Controller;
6  use App\Http\Requests\StoreProductRequest;
7  use App\Http\Requests\UpdateProductRequest;
8  use App\Http\Requests\UpdateUserRequest;
9  use App\Models\Category;
10 use App\Models\Product;
11 use App\Models\User;
12 use Illuminate\Support\Facades\Auth;
13 use Illuminate\Support\Facades\File;
14 use Illuminate\Support\Facades\Hash;
15 use RealRashid\SweetAlert\Facades\Alert;
16
17 use function PHPUnit\Framework\fileExists;
18
19 class EditorController extends Controller
20 {
21     public function profile()
22     {
23         $movies = Product::where('user_id',Auth::user()->id)->get();
24         return view('front.pages.editor.profile',compact('movies'));
25     }
26
27     public function edit(User $user)
28     {
29         return view('front.pages.editor.edit',compact('user'));
30     }
31
32     public function update(UpdateUserRequest $request,User $user)
33     {
34         $data=$request->validated();
35         if($request->file('image')){
36             if(fileExists(public_path('images/' . $user->image)))
37             {
38                 File::delete(public_path('images/' . $user->image));
39             }
40             $file= $request->file('image');
41             $filename= date('YmdHi').$file->getClientOriginalName();
42             $file->move(public_path('images'), $filename);
43             $data['image'] = $filename;
44         }
45         $user->update($data);
46         Alert::success('Success Title', 'User Was Updated');
47         return redirect(route('home.editor.profile'));
48     }
}
```

```
● ● ●
```

```
1  public function movieCreate()
2  {
3      $categories = Category::get();
4      return view('front.pages.editor.movieCreate',compact('categories'));
5  }
6
7  public function movieStore(StoreProductRequest $request)
8  {
9      if($request->file('image')){
10         $file= $request->file('image');
11         $filename= date('YmdHi').$file->getClientOriginalName();
12         $file-> move(public_path('images'), $filename);
13         $data['image'] = $filename;
14     }
15
16     Product::create([
17         'name' => $request->name,
18         'description' => $request->description,
19         'category_id' => $request->category_id,
20         'user_id' => Auth::user()->id,
21         'image' => $filename ?? ''
22     ]);
23
24     Alert::success('Success Title', 'Movie Was Created Successfully !!!');
25     return redirect()->back();
26 }
27
28 public function movieEdit(Product $product)
29 {
30     $categories = Category::get();
31     return view('front.pages.editor.editMovie',compact('product','categories'));
32 }
```

```
● ● ●
1     public function movieUpdate(UpdateProductRequest $request, Product $product)
2     {
3         $data=$request->all();
4         if($request->file('image')){
5             if(fileExists(public_path('images/' . $product->image)))
6             {
7                 File::delete(public_path('images/' . $product->image));
8             }
9             $file= $request->file('image');
10            $filename= date('YmdHi').$file->getClientOriginalName();
11            $file-> move(public_path('images'), $filename);
12            $data['image'] = $filename;
13        }
14        $data['is_approved'] = isset($data['is_approved']) ? $data['is_approved'] : 0 ;
15        $product->update($data);
16        Alert::success('Success Title', 'Product Was Updated');
17        return redirect()->back();
18    }
19
20    public function movieDelete(Product $product)
21    {
22        $product->delete();
23        Alert::success('Success Title', 'Product Was Deleted');
24        return redirect()->back();
25    }
26}
27
```

● دالة الـ **profile** تفتح الصفحة الشخصية

```
1 @extends('front.master')
2
3 @section('css')
4     <link rel="stylesheet" href="{{ asset('css/profile.css') }}">
5 @endsection
6
7 @section('content')
8     <div class="row gutters-sm">
9         <div class="col-md-4 mb-3">
10            <div class="card">
11                <div class="card-body">
12                    <div class="d-flex flex-column align-items-center text-center" style="text-align: center">
13                        
15                        <div class="mt-3">
16                            <h4>{{ Auth::user()->name }}</h4>
17                            <p class="text-secondary mb-1">Editor on Movies</p>
18                            <p class="text-muted font-size-sm">{{ Auth::user()->phone }}</p>
19                            <p class="text-muted font-size-sm">{{ Auth::user()->email }}</p>
20
21                    </div>
22                </div>
23            </div>
24        </div>
25        <div class="col-md-8">
26            <div class="card mb-3">
27                <div class="card-body">
28                    <div class="row">
29                        <div class="col-sm-3">
30                            <h6 class="mb-0">Full Name</h6>
31                        </div>
32                        <div class="col-sm-9 text-secondary">
33                            {{ Auth::user()->name }}
34                        </div>
35                    </div>
36                    <hr>
37                    <div class="row">
38                        <div class="col-sm-3">
39                            <h6 class="mb-0">Email</h6>
40                        </div>
41                        <div class="col-sm-9 text-secondary">
42                            {{ Auth::user()->email }}
43                        </div>
44                    </div>
45                    <hr>
46                    <div class="row">
47                        <div class="col-sm-3">
48                            <h6 class="mb-0">Phone</h6>
49                        </div>
50                        <div class="col-sm-9 text-secondary">
51                            {{ Auth::user()->phone }}
52                        </div>
53                    </div>
54                    <hr>
55                    <div class="row">
56                        <div class="col">
57                            <a class="btn btn-primary" target="_blank"
58                                href="{{ route('home.editor.edit', Auth::user()->id) }}>Edit</a>
59                        </div>
60                        <div class="col">
61                            <a class="btn btn-primary" target="_blank"
62                                href="{{ route('home.editor.movieCreate') }}>Create Movie</a>
63                        </div>
64                    </div>
65                </div>
66            </div>
67        </div>
68    </div>
```

```

1      <div class="col">
2          <table class="table table-dark table-striped">
3              <thead>
4                  <tr>
5                      <th scope="col">ID</th>
6                      <th scope="col">Name</th>
7                      <th scope="col">Description</th>
8                      <th scope="col">Image</th>
9                      <th scope="col">Category</th>
10                     <th scope="col">Is Approved</th>
11                     <th scope="col">Edit</th>
12                     <th scope="col">Delete</th>
13             </tr>
14         </thead>
15         <tbody>
16             @foreach ($movies as $movie)
17             <tr>
18                 <th scope="row">{{$movie->id}}</th>
19                 <td>{{$movie->name}}</td>
20                 <td>{!! $movie->description !!}</td>
21                 <td>
22                     
23                 </td>
24                 <td>{{$movie->category->name ?? ""}}</td>
25
26                 @if($movie->is_approved == false)
27                     <td>
28                         <button class="btn btn-danger">No</button>
29                     </td>
30                 @else
31                     <td>
32                         <button class="btn btn-success">Yes</button>
33                     </td>
34                 @endif
35
36                 <td>
37                     <form method="get" action="{{route('home.editor.movieEdit',$movie)}}>
38                         @csrf
39                         <button type="submit" class="btn btn-success">
40                             <i class="fa fa-edit"></i>
41                         </button>
42                     </form>
43                 </td>
44                 <td>
45                     <form method="POST" action="{{route('home.editor.movieDelete',$movie)}}>
46                         @csrf
47                         @method('DELETE')
48                         <button type="submit" class="btn btn-danger">
49                             <i class="fa fa-trash"></i>
50                         </button>
51                     </form>
52                 </td>
53             </tr>
54         @endforeach
55     </tbody>
56 </table>
57 </div>
58
59
60
61
62     </div>
63 @endsection
64
65
66 @section('script')
67     <script></script>
68 @endsection
69

```

• دالة ال edit تفتح صفحة تعديل المعلومات الشخصية لل editor

```
1 @extends('front.master')
2
3 @section('css')
4     <link rel="stylesheet" href="{{ asset('css/forms.css') }}>
5 @endsection
6
7 @section('content')
8     <div class="form-body">
9         <div class="row" style="width: 60%; margin:0 auto">
10            <div class="form-holder">
11                <div class="form-content">
12                    <div class="form-items">
13                        <h3>Edit Editor Personal Data</h3>
14                        <p>Fill in the data below.</p>
15                        <form class="requires-validation" novalidate method="POST" enctype="multipart/form-data" action="{{ route('home.editor.update', $user->id) }}">
16                            @csrf
17                            <div class="col-md-12">
18                                <input class="form-control" type="email" name="email"
19                                       placeholder="E-mail Address" value="{{ $user->email }}>
20                                @error('email')
21                                    <div class="invalid-feedback">{{ $message }}</div>
22                                @enderror
23                            </div>
24
25                            <div class="col-md-12">
26                                <input class="form-control" type="text" name="name"
27                                       placeholder="Name" value="{{ $user->name }}>
28                                @error('name')
29                                    <div class="invalid-feedback">{{ $message }}</div>
30                                @enderror
31                            </div>
32
33                            <div class="col-md-12">
34                                <input class="form-control" type="text" name="phone"
35                                       placeholder="phone" value="{{ $user->phone }}>
36                                @error('phone')
37                                    <div class="invalid-feedback">{{ $message }}</div>
38                                @enderror
39                            </div>
40
41                            <div class="col-md-12">
42                                <input class="form-control" type="file" name="image"
43                                       placeholder="image">
44                                @error('image')
45                                    <div class="invalid-feedback">{{ $message }}</div>
46                                @enderror
47                            </div>
48
49                            <div class="col-md-12">
50                                <button type="submit" class="btn btn-primary">Update</button>
51                            </div>
52                        </form>
53                    </div>
54                </div>
55            </div>
56        </div>
57    @endsection
58
59
60 @section('script')
61     <script></script>
62 @endsection
63
64
```

- دالة الـ **update** تستقبل بيانات التعديل على المستخدم وتقوم بالتعديل على البيانات في الـ **database**
- دالة الـ **moveCreate** تفتح صفحة للـ **editor** لإنشاء فيلم جديد

```

1  @extends('front.master')
2
3  @section('css')
4      <link rel="stylesheet" href="{{ asset('css/forms.css') }}>
5  @endsection
6
7  @section('content')
8      <div class="form-body">
9          <div class="row" style="width: 60%; margin: 0 auto">
10             <div class="form-holder">
11                 <div class="form-content">
12                     <div class="form-items">
13                         <h3>Create a Movie</h3>
14                         <p>Fill in the data below.</p>
15                         <form class="requires-validation" novalidate method="POST" enctype="multipart/form-data" action="{{ route('home.editor.movieStore') }}">
16                             @csrf
17
18                             <div class="col-md-12">
19                                 <input class="form-control" type="text" name="name"
20                                     placeholder="Name">
21                                 @error('name')
22                                     <div class="invalid-feedback">{{ $message }}</div>
23                                 @enderror
24                             </div>
25
26                             <div class="col-md-12 mt-5">
27                                 <textarea class="form-control" type="text" name="description"
28                                     placeholder="Description"> </textarea>
29                                 @error('description')
30                                     <div class="invalid-feedback">{{ $message }}</div>
31                                 @enderror
32                             </div>
33
34
35                             <div class="col-md-12">
36                                 <input class="form-control" type="file" name="image"
37                                     placeholder="Image">
38                                 @error('image')
39                                     <div class="invalid-feedback">{{ $message }}</div>
40                                 @enderror
41                             </div>
42
43                             <div class="col-md-12 mb-5">
44                                 <select name="category_id" class="form-select mt-3" required>
45                                     <option selected disabled value="0">Choose Category</option>
46                                     @foreach ($categories as $category)
47                                         <option value="{{ $category->id }}>{{ $category->name }}</option>
48                                     @endforeach
49                                 </select>
50                                 @error('category_id')
51                                     <div class="invalid-feedback">{{ $message }}</div>
52                                 @enderror
53                                 <div class="invalid-feedback" style="visibility: hidden"> asdfsaf </div>
54                             </div>
55
56                             <div class="col-md-12">
57                                 <button type="submit" class="btn btn-primary">Create</button>
58                             </div>
59                         </form>
60                     </div>
61                 </div>
62             </div>
63         </div>
64     @endsection
65
66
67     @section('script')
68         <script></script>
69     @endsection
70
71

```

- دالة ال **movieStore** تستقبل بيانات الفيلم الجديد وتقوم ب تخزينه في ال database
- دالة ال **movieEdit** تفتح صفحة لل التعديل على الفلم editor

```

1  @extends('front.master')
2
3  @section('css')
4      <link rel="stylesheet" href="{{ asset('css/forms.css') }}>
5  @endsection
6
7  @section('content')
8      <div class="row" style="width: 60%; margin:0 auto">
9          <div class="form-holder">
10             <div class="form-content">
11                 <div class="form-items">
12                     <h3>Edit a Movie</h3>
13                     <p>Fill in the data below.</p>
14                     <form class="requires-validation" novalidate method="POST" enctype="multipart/form-data" action="{{route('home.editor.movieUpdate', $product)}}>
15                         @csrf
16                         @method('PUT')
17
18                         <input type="hidden" name="id" value="{{Auth::user()->id}}>
19                         <div class="col-md-12">
20                             <input class="form-control" type="text" name="name"
21                                 placeholder="Name" value="{{$product->name}}>
22                             @error('name')
23                                 <div class="invalid-feedback">{{ $message }}</div>
24                             @enderror
25                         </div>
26
27
28                         <div class="col-md-12 mt-5">
29                             <textarea class="form-control" type="text" name="description"
30                                 placeholder="description" value="{{$product->description}}> </textarea>
31                             @error('description')
32                                 <div class="invalid-feedback">{{ $message }}</div>
33                             @enderror
34                         </div>
35
36
37                         <div class="col-md-12">
38                             <input class="form-control" type="file" name="image"
39                                 placeholder="image">
40                             @error('image')
41                                 <div class="invalid-feedback">{{ $message }}</div>
42                             @enderror
43                         </div>
44
45                         <div class="col-md-12 mb-5">
46                             <select name="category_id" class="form-select mt-3 required">
47                                 <option selected disabled value="0">Choose Category</option>
48                                 @foreach ($categories as $category)
49                                     <option value="{{ $category->id }}" {{($category->id == $product->category_id) ? "selected" : ""}}>{{ $category->name }}</option>
50                                 @endforeach
51                             </select>
52                             @error('category_id')
53                                 <div class="invalid-feedback">{{ $message }}</div>
54                             @enderror
55                             <div class="invalid-feedback" style="visibility: hidden"> asdfsaf </div>
56                         </div>
57
58                         <div class="col-md-12">
59                             <button type="submit" class="btn btn-primary">Update</button>
60                         </div>
61
62                     </form>
63
64             </div>
65         </div>
66     </div>
67     @endsection
68
69
70     @section('script')
71         <script></script>
72     @endsection
73
74

```

- دالة الـ **movieUpdate** تستقبل بيانات التعديل من ال editor وتقوم بتعديل الفيلم في الـ database
- دالة الـ **movieDelete** تزحف الفيلم من الـ database

Front Home Controller

```
● ○ ●  
1  <?php  
2  
3  namespace App\Http\Controllers\Front;  
4  
5  use App\Http\Controllers\Controller;  
6  use App\Models\Product;  
7  use App\Models\Review;  
8  use Illuminate\Http\Request;  
9  
10 class HomeController extends Controller  
11 {  
12     public function index()  
13     {  
14         $first = Product::IsApproved()->limit(3)->get();  
15         $second = Product::IsApproved()->skip(3)->take(2)->get();  
16         $third = Product::IsApproved()->skip(5)->take(4)->get();  
17         $reviews = Review::limit(3)->get();  
18         return view('front.index',compact('first','second','third','reviews'));  
19     }  
20  
21     public function about()  
22     {  
23         return view('front.pages.about.index');  
24     }  
25 }  
26
```

• دالة ال index تفتح صفحة ال index في واجهة الموقع

```
1 @extends('front.master')
2
3 @section('content')
4     <h1 style="margin-top: 20px; margin-bottom: 50px; text-align: center; color: #DE0A0A; font-weight: bolder; letter-spacing: 0.2em;">
5         Latest Movies
6     </h1>
7     <div class="row">
8         <div class="col-md-9">
9             <div class="slider">
10                <ul class="slides">
11                    @foreach ($first as $index => $product)
12                        <li>
13                            <a href="{{route('home.movies.show',$product)}}>
14                                
16                            </a>
17                        </li>
18                    @endforeach
19                </ul>
20            </div>
21        </div>
22        <div class="col-md-3">
23            <div class="row">
24                @foreach ($second as $index => $product)
25                    <div class="col-sm-6 col-md-12">
26                        <div class="latest-movie">
27                            <a href="{{route('home.movies.show',$product)}}></a>
29                        </div>
30                    </div>
31                @endforeach
32            </div>
33        </div>
34    </div>
35 </div> <!-- .row -->
36 <div class="row">
37     @foreach ($third as $product)
38         <div class="col-sm-6 col-md-3">
39             <div class="latest-movie">
40                 <a href="{{route('home.movies.show',$product)}}></a>
42             </div>
43         </div>
44     @endforeach
45 </div>
46 </div> <!-- .row -->
47
48 <br />
49 <br />
50 <h1 style="margin-top: 30px; margin-bottom: 30px; text-align: center; color: #DE0A0A; font-weight: bolder; letter-spacing: 0.2em;">
51     Making Reviews
52 </h1>
53 <p class="txt">Writing a movie review is a great way to express your opinion of a movie.
54     The
55     purpose of most movie reviews is to help readers determine if they want to
56     watch, rent, or buy the movie. The review should give enough details about the
57     movie so that the reader can make an informed decision, without giving away
58     any essentials such as the plot or any surprises.
59     <br />
60     <br />
61     views analyze the effectiveness of the plot, theme, acting, direction, special
62     effects, musical effects, cinematography, and all other elements that created the
63     movie. There are qualities and guidelines that a critique of a movie should
64     possess. Avoid the use of generalized opinions such as "it was a great movie" or
65     "the acting was horrible," but rather give specific reasons and the whys.
66     <br />
67     <br />
68 </p>
```

```

1  <br />
2  </br>
3  <h1 style="margin-top: 20px;text-align: center; color: #DE0A0A; font-weight: bolder; letter-spacing: 0.2em;">
4      Latest Reviews
5  </h1>
6  <div class="reviews">
7
8      <div class='container mx-auto mt-5 col-md-10 col-11 col-sm-12 text-center' 
9          style="text-align: center; align-items: center; margin-left: 0%; position: relative; transform: translate(10%, 5%);>
10
11         <div class="row" style="justify-content: center">
12             @foreach ($reviews as $review)
13                 <div class="card col-md-4 col-11">
14                     <div class="card-content text-center">
15                         <div class="card-body p-0">
16                             <div class="profile mb-4 mt-3"> 
17                         </div>
18                         <br />
19                         <h4 style="color: #DE0A0A;" id="rv1"> Movie Name : {{$review->product->name ?? ""}}</h4>
20                         <div class="card-subtitle">
21                             <p> <small class="text-muted"> <i class="fa fa-quote-left"></i>
22                                 {{$review->review}} <i class="fa fa-quote-right"></i>
23                             </small>
24                         </p>
25                     </div>
26
27                     <div class="ratings">
28                         <i class="fa fa-star"></i>
29                         <i class="fa fa-star"></i>
30                     </div>
31                 </div>
32             </div>
33         @endforeach
34     </div>
35
36     </div>
37
38     </div>
39 </endsection>
```

• دالة ال about تفتح صفحة ال about

```
1 @extends('front.master')
2
3 @section('css')
4     <link rel="stylesheet" href="{{ asset('css/movies.css') }}">
5 @endsection
6 @section('content')
7     <div class="row">
8         <div class="col-md-4">
9             <figure></figure>
10        </div>
11        <div class="col-md-8">
12            <p class="leading">There is one simple way through which movie buffs can actually save their
13                money. From our movie review site,
14                movie buffs come to know people's opinions about a particular movie. On the basis of the
15                review a movie buff
16                can take his decision (whether to watch it in the theater or categorize it as a flop
17                movie).</p>
18            <p style="line-height: 1.5">Television commercials on new releases are enough to entice
19                millions of movie lovers who run off to the movie theatres to watch those movies. These
20                movie commercials are created to hoodwink the audience. A commercial tends to show that
21                a movie is simply amazing, whereas when a movie buff actually sees it, he finds it to be
22                a complete flop picture.
23                Just by watching these commercials, movie buffs run to theaters and waste their money to
24                watch flop films
25                that do not even stay in the theaters for a week. .</p>
26        </div>
27    </div>
28
29    <div class="row">
30        <div class="col-md-9">
31            <h2 class="section-title" style="color: #DE0A0A">Vision & Misssion</h2>
32            <p style="line-height: 1.5">we offer the best reviews of the newly released films. The
33                registration process is
34                simple and once you are a registered member, you can read free film reviews and write
35                and submit your own reviews as well. A bad movie can spoil your mood and interest.
36                Moreover,
37                it also burns a hole in your pocket as you spend dollars to watch a flop movie. So it is
38                always
39                better to visit our movie review site, read the review of a newly released movie and
40                then,
41                decide whether it is worth watching in the theater.</p>
42
43            <p style="line-height: 1.5">our movie review website are free site where people can visit
44                and read reviews any time they wish. The user can also
45                register to some of the movie sites. This allows him to write reviews about the films he
46                has watched. After all,
47                others might also be interested to know about the films that you have
48                watched.Registration is very simple procedure.
49                All you need to do is to follow the steps and get your name registered in the particular
50                site.</p>
51    </div>
52    <div class="col-md-3">
53        <h2 class="section-title" style="color: #DE0A0A; font-size:larger;">Best Websites for
54            watching movies</h2>
55        <ul class="arrow" style="color: #DE0A0A">
56            <li><a href="#">Crackle</a></li>
57            <li><a href="#">Popcornflix</a></li>
58            <li><a href="#">Yidio</a></li>
59            <li><a href="#">YouTube</a></li>
60            <li><a href="#">NetFlix</a></li>
61            <li><a href="#">EgyBest</a></li>
62            <li><a href="#">Vudu</a></li>
63        </ul>
64    </div>
65 </div> <!-- .row -->
```

```
1 <h2 class="section-title" style="color: #DE0A0A">Our Team</h2>
2     <div class="row">
3
4         <div class="col-md-4">
5             <div class="team">
6                 <figure class="team-image"></figure>
7                 <h2 class="team-name">Ahmed Oweid</h2>
8                 <small class="team-title">Team Leader</small>
9             </div>
10        </div>
11        <div class="col-md-4">
12            <div class="team">
13                <figure class="team-image"></figure>
14                <h2 class="team-name">Mahmoud </h2>
15                <small class="team-title">Front-End</small>
16            </div>
17        </div>
18        <div class="col-md-4">
19            <div class="team">
20                <figure class="team-image"></figure>
21                <h2 class="team-name">Koloud Abd-ElFattah</h2>
22                <small class="team-title">Front-End</small>
23            </div>
24        </div>
25        <div class="col-md-4">
26            <div class="team">
27                <figure class="team-image"></figure>
28                <h2 class="team-name">Rania Moahmed</h2>
29                <small class="team-title">Front-End</small>
30            </div>
31        </div>
32        <div class="col-md-4">
33            <div class="team">
34                <figure class="team-image"></figure>
35                <h2 class="team-name">Omar Mohammed</h2>
36                <small class="team-title">Back-End</small>
37            </div>
38        </div>
39        <div class="col-md-4">
40            <div class="team">
41                <figure class="team-image"></figure>
42                <h2 class="team-name">Kholoud Bahgat</h2>
43                <small class="team-title">BAck-End</small>
44            </div>
45        </div>
46
47
48    </div>
49 @endsection
```

Front Movie Controller

```
1  <?php
2
3  namespace App\Http\Controllers\Front;
4
5  use App\Http\Controllers\Controller;
6  use App\Models\Product;
7  use Illuminate\Http\Request;
8
9  class MovieController extends Controller
10 {
11     public function index()
12     {
13         $movies = Product::IsApproved()->get();
14         return view('front.pages.movies.index', compact('movies'));
15     }
16
17     public function show(Product $product)
18     {
19         $reviews = $product->reviews;
20         $sum = 0;
21         foreach ($reviews as $review) {
22             $sum += $review->rate;
23         }
24         $avg = $sum / ($reviews->count() == 0 ? 1 : $reviews->count());
25         $avg = $avg * 5;
26         $avg = round($avg, 2);
27         return view('front.pages.movies.show', compact('product', 'avg'));
28     }
29 }
30 }
```

● دالة ال **index** تفتح صفحة عرض جميع الأفلام

```
1 @extends('front.master')
2
3 @section('css')
4     <link rel="stylesheet" href="{{ asset('css/movies.css') }}">
5 @endsection
6 @section('content')
7     <div class="latest-posts text-center">
8
9         <div class="row">
10            <h1 style="margin-top: 30px; margin-bottom: 50px; text-align: center; color: #DE0A0A; font-weight: bolder; letter-spacing: 0.2em;">
11                Recent Movies
12            </h1>
13
14            @foreach ($movies as $movie)
15                <div class="col-md-3 col-sm-6 mb-5">
16                    <div class="card">
17                        
18                        <h4 class="card-title">{{$movie->name}}</h4>
19                        <h4 class="card-title">By :{{$movie->user->name ?? ""}}</h4>
20                        <a href="{{route('home.movies.show', $movie)}}" class="card-link">
21                            <button class="but">Show Movie</button></a>
22
23                    </div>
24                </div>
25            @endforeach
26        </div>
27    </div>
28 @endsection
29
```

- دالة ال **show** تفتح صفحة عرض الفيلم مع التعليقات الخاصة به

```

● ● ●
1 @extends('front.master')
2
3 @section('css')
4     <link rel="stylesheet" href="{{ asset('css/makeReview.css') }}">
5 @endsection
6 @section('content')
7     <div class="page">
8
9         <div class="content">
10            <div class="row">
11                <div class="col-md-6">
12                    <figure class="movie-poster"></figure>
13                </div>
14                <div class="col-md-6">
15                    <h2 class="movie-title">{{ $product->name }}</h2>
16                    <div class="movie-summary">
17                        {!! $product->description !!}
18                    </div>
19                    <ul class="movie-meta">
20                        <li><strong>Category:</strong> {{ $product->category->name ?? '' }}</li>
21                        <li><strong>User:</strong> {{ $product->user->name ?? '' }}</li>
22                    </ul>
23                    <h2 class="movie-title">
24                        Rate :
25                        @if($avg >= 2.5)
26                            <span class="text-success">{{$avg}}</span>
27                        @else
28                            <span class="text-danger">{{$avg}}</span>
29                        @endif
30                        @if($avg >= 0 && $avg <= 1)
31                            @elseif($avg > 1 && $avg <= 2)
32                            @elseif($avg > 2 && $avg <= 3)
33                            @elseif($avg > 3 && $avg <= 4)
34                            @endif
35                        </h2>
36                    </div>
37                <!-- .row -->
38                <div class="entry-content">
39                    <div class="row d-flex justify-content-left">
40                        <div class="col">
41                            <div class="card bg-dark text-light">
42                                <div class="p-2 text-center" style="font-size: 1.5em">
43                                    <h4 class="mb-0">Recent Reviews</h4>
44                                    <p class="fw-light mb-4 pb-2">Latest Reviews section by users</p>
45                                </div>
46                                @foreach ($product->reviews as $review)
47                                    <div class="card-body p-4">
48                                        <div class="row d-flex flex-start align-items-center">
49                                            <div class="col-10">
50                                                
53                                            <div>
54                                                <h6 class="fw-bold mb-1">{{ $review->user->name ?? "" }}</h6>
55                                                <div class="d-flex align-items-center mb-3">
56                                                    <p class="mb-0">
57                                                        {{ $review->created_at }}
58                                                    </p>
59                                                </div>
60                                                <p class="mb-0">
61                                                    {{ $review->review }}
62                                                </p>
63                                            </div>
64                                        </div>
65                                    </div>
66                                </div>

```

Front Review Controller

```
1  <?php
2
3  namespace App\Http\Controllers\Front;
4
5  use App\Http\Controllers\Controller;
6  use App\Http\Requests\StoreReviewRequest;
7  use App\Models\Review;
8  use Illuminate\Http\Request;
9  use Illuminate\Support\Facades\Http;
10 use RealRashid\SweetAlert\Facades\Alert;
11
12 class ReviewController extends Controller
13 {
14     public function create(StoreReviewRequest $request)
15     {
16         $res = Http::get('http://127.0.0.1:5005/get_rate', [
17             'word' => $request->review
18         ]);
19
20         $result = $res->body();
21         $result = str_replace("\n", "", $result);
22         $result = str_replace('"', "", $result);
23         if ($result == "positive") {
24             Review::create([
25                 'user_id' => $request->user_id,
26                 'product_id' => $request->product_id,
27                 'review' => $request->review,
28                 'rate' => true,
29             ]);
30         } else if ($result == "negative") {
31             Review::create([
32                 'user_id' => $request->user_id,
33                 'product_id' => $request->product_id,
34                 'review' => $request->review,
35                 'rate' => false,
36             ]);
37         } else {
38             Alert::error('Error Title', 'There was an error');
39             return redirect()->back();
40         }
41
42         Alert::success('Success Title', 'Review Was Created');
43         return redirect()->back();
44     }
45 }
46 }
```

- دالة **create** تقوم بإنشاء تعليق على فيلم وهنا نقوم بإرسال التعليق إلى ال endpoint الخاصة بال machine learning ونستقبل النتيجة ونعرضها للمستخدم

Front Viewer Controller

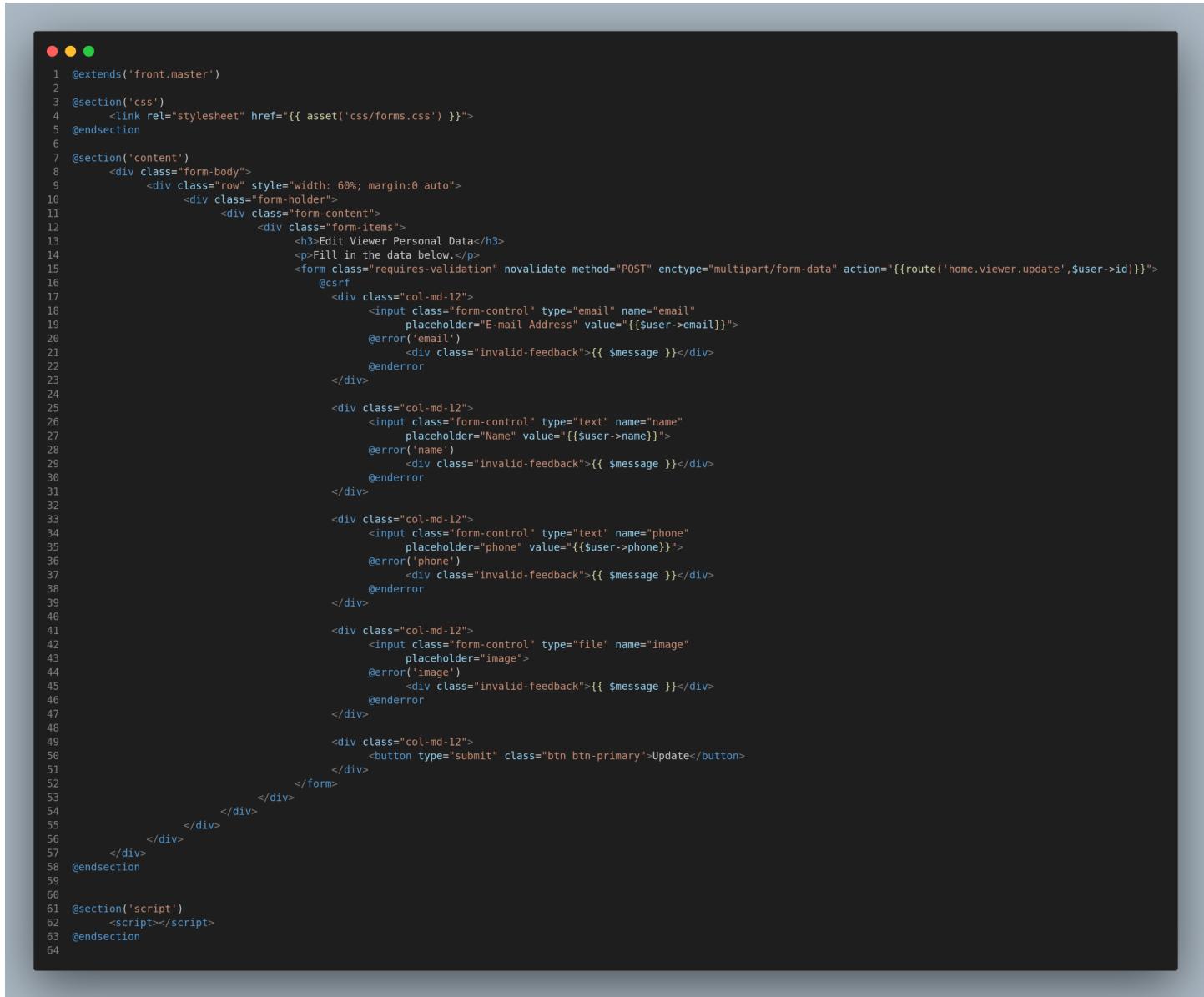
```
● ● ●
1 <?php
2
3 namespace App\Http\Controllers\Front;
4
5 use App\Http\Controllers\Controller;
6 use App\Http\Requests\UpdateUserRequest;
7 use App\Models\User;
8 use Illuminate\Http\Request;
9 use Illuminate\Support\Facades\File;
10 use RealRashid\SweetAlert\Facades\Alert;
11
12 use function PHPUnit\Framework\fileExists;
13
14 class ViewerController extends Controller
15 {
16     public function profile()
17     {
18         return view('front.pages.viewer.profile');
19     }
20
21     public function edit(User $user)
22     {
23         return view('front.pages.viewer.edit', compact('user'));
24     }
25
26     public function update(UpdateUserRequest $request, User $user)
27     {
28         $data=$request->validated();
29         if($request->file('image')){
30             if(fileExists(public_path('images/' . $user->image)))
31             {
32                 File::delete(public_path('images/' . $user->image));
33             }
34             $file= $request->file('image');
35             $filename= date('YmdHi').$file->getClientOriginalName();
36             $file->move(public_path('images'), $filename);
37             $data['image'] = $filename;
38         }
39         $user->update($data);
40         Alert::success('Success Title', 'User Was Updated');
41         return redirect(route('home.viewer.profile'));
42     }
43 }
44
```

● دالة الـ **profile** تفتح الصفحة الشخصية للـ **viewer**

```
● دالة الـ profile تفتح الصفحة الشخصية للـ viewer

1  @extends('front.master')
2
3  @section('css')
4      <link rel="stylesheet" href="{{ asset('css/profile.css') }}">
5  @endsection
6
7  @section('content')
8      <div class="row gutters-sm">
9          <div class="col-md-4 mb-3">
10             <div class="card">
11                 <div class="card-body">
12                     <div class="d-flex flex-column align-items-center text-center" style="text-align: center">
13                         
15                         <div class="mt-3">
16                             <h4>{{ Auth::user()->name }}</h4>
17                             <p class="text-secondary mb-1">Editor on Movies</p>
18                             <p class="text-muted font-size-sm">{{ Auth::user()->phone }}</p>
19                             <p class="text-muted font-size-sm">{{ Auth::user()->email }}</p>
20                         </div>
21                     </div>
22                 </div>
23             </div>
24         <div class="col-md-8">
25             <div class="card mb-3">
26                 <div class="card-body">
27                     <div class="row">
28                         <div class="col-sm-3">
29                             <h6 class="mb-0">Full Name</h6>
30                         </div>
31                         <div class="col-sm-9 text-secondary">
32                             {{ Auth::user()->name }}
33                         </div>
34                     </div>
35                     <hr>
36                     <div class="row">
37                         <div class="col-sm-3">
38                             <h6 class="mb-0">Email</h6>
39                         </div>
40                         <div class="col-sm-9 text-secondary">
41                             {{ Auth::user()->email }}
42                         </div>
43                     </div>
44                     <hr>
45                     <div class="row">
46                         <div class="col-sm-3">
47                             <h6 class="mb-0">Phone</h6>
48                         </div>
49                         <div class="col-sm-9 text-secondary">
50                             {{ Auth::user()->phone }}
51                         </div>
52                     </div>
53                     <hr>
54                     <div class="row">
55                         <div class="col">
56                             <a class="btn btn-primary" target="_blank"
57                                 href="{{ route('home.viewer.edit', Auth::user()->id) }}>Edit</a>
58                         </div>
59                     </div>
60                 </div>
61             </div>
62         </div>
63     </div>
64
65     </div>
66  @endsection
67
68
```

● دالة الـ **edit** تفتح صفحة تعديل الملف الشخصي



```
1 @extends('front.master')
2
3 @section('css')
4 <link rel="stylesheet" href="{{ asset('css/forms.css') }}">
5 @endsection
6
7 @section('content')
8     <div class="form-body">
9         <div class="row" style="width: 60%; margin:0 auto">
10            <div class="form-holder">
11                <div class="form-content">
12                    <div class="form-items">
13                        <h3>Edit Viewer Personal Data</h3>
14                        <p>Fill in the data below.</p>
15                        <form class="requires-validation" novalidate method="POST" enctype="multipart/form-data" action="{{route('home.viewer.update',$user->id)}}>
16                            @csrf
17                            <div class="col-md-12">
18                                <input class="form-control" type="email" name="email"
19                                      placeholder="E-mail Address" value="{{$user->email}}>
20                                @error('email')
21                                    <div class="invalid-feedback">{{ $message }}</div>
22                                @enderror
23                            </div>
24
25                            <div class="col-md-12">
26                                <input class="form-control" type="text" name="name"
27                                      placeholder="Name" value="{{$user->name}}>
28                                @error('name')
29                                    <div class="invalid-feedback">{{ $message }}</div>
30                                @enderror
31                            </div>
32
33                            <div class="col-md-12">
34                                <input class="form-control" type="text" name="phone"
35                                      placeholder="phone" value="{{$user->phone}}>
36                                @error('phone')
37                                    <div class="invalid-feedback">{{ $message }}</div>
38                                @enderror
39                            </div>
40
41                            <div class="col-md-12">
42                                <input class="form-control" type="file" name="image"
43                                      placeholder="image">
44                                @error('image')
45                                    <div class="invalid-feedback">{{ $message }}</div>
46                                @enderror
47                            </div>
48
49                            <div class="col-md-12">
50                                <button type="submit" class="btn btn-primary">Update</button>
51                            </div>
52                        </form>
53                    </div>
54                </div>
55            </div>
56        </div>
57    @endsection
58
59
60    @section('script')
61        <script></script>
62    @endsection
63
64
```

● دالة الـ **update** تستقبل الـ **data** من المستخدم وتقوم بالتعديل عليها في الـ **database**

Sentiment Analysis Code

```
● ● ●

1 from flask import request
2 from tqdm.notebook import tqdm
3 from nltk.sentiment import SentimentIntensityAnalyzer
4 import seaborn as sns
5 import matplotlib.pyplot as plt
6 import numpy as np
7 import pandas as pd
8 import nltk
9 from flask import Flask
10 app = Flask(__name__)
11 nltk.download('maxent_ne_chunker')
12 nltk.download('words')
13 nltk.download('vader_lexicon')
14 plt.style.use('ggplot')
15
16 df = pd.read_csv('./machine/archive/Twitter_Data.csv')
17
18 ax = df['category'].value_counts().sort_index().plot(
19     kind='bar', title='Count of Reviews by Category', figsize=(10, 15))
20
21 ax.set_xlabel('Rev')
22
23 example = df['clean_text'][50]
24 print(example)
25 tokens = nltk.word_tokenize(example)
26
27 tagged = nltk.pos_tag(tokens)
28
29 entities = nltk.chunk.ne_chunk(tagged)
30
31 sia = SentimentIntensityAnalyzer()
32
33 x = sia.polarity_scores(example)
34
35
36 @app.get('/get_rate')
37 def list_programming_languages():
38     word = request.args['word']
39     tokens = nltk.word_tokenize(word)
40     tagged = nltk.pos_tag(tokens)
41     entities = nltk.chunk.ne_chunk(tagged)
42     sia = SentimentIntensityAnalyzer()
43     x = sia.polarity_scores(word)
44     result = ''
45     if (x['compound'] >= 0):
46         result = 'positive'
47     else:
48         result = "negative"
49     return result
50
```

ثم نقوم بتشغيل المشروع عن طريق الـ command

`php artisan serve`

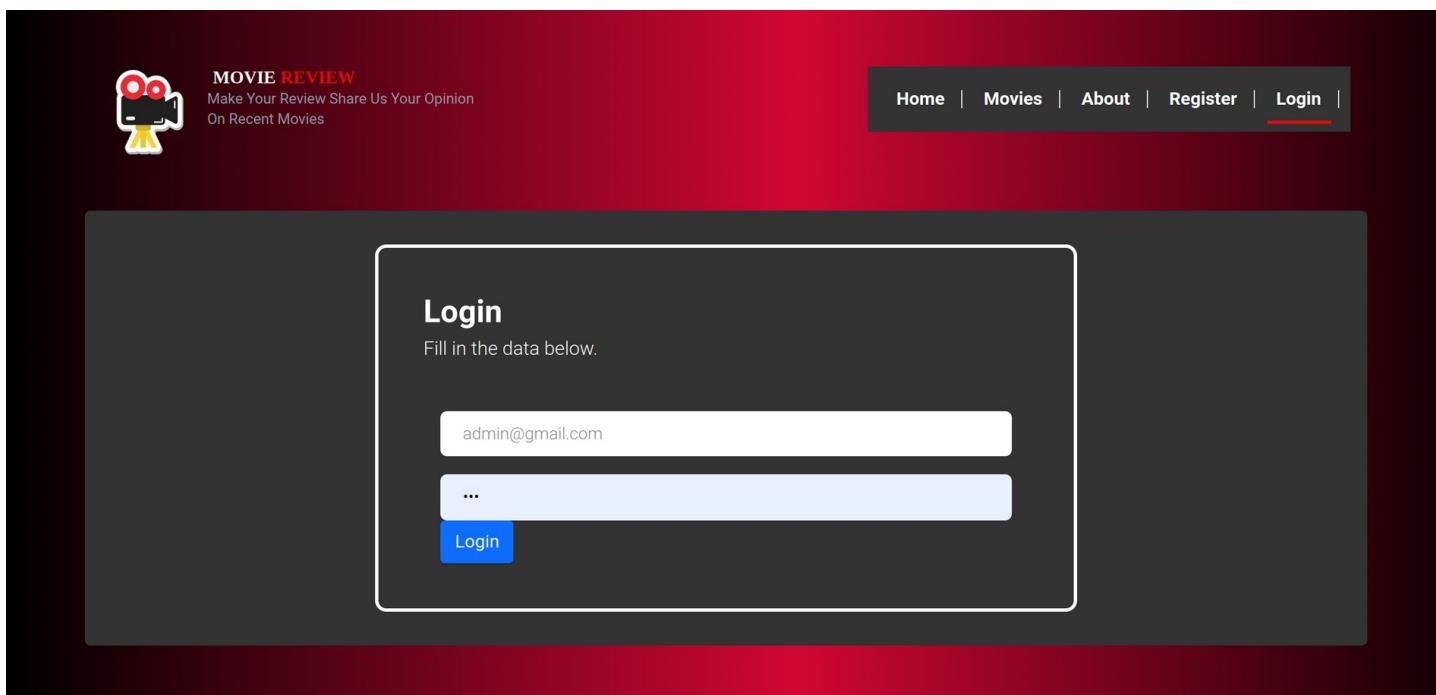
```
mm@mm-OptiPlex-9020:~/Documents/term 2/movieReview$ php artisan serve
Starting Laravel development server: http://127.0.0.1:8000
[Thu May 11 00:02:01 2023] PHP 8.1.2-1ubuntu2.11 Development Server (http://127.0.0.1:8000) started
```

هذا يفتح server على الـ url المظلل

صفحة الدخول

البريد الإلكتروني: admin@gmail.com

رقم المرور : 123



The screenshot shows a movie review website's login page. At the top left is a camera icon with the text "MOVIE REVIEW" and "Make Your Review Share Us Your Opinion On Recent Movies". At the top right is a navigation bar with links: Home, Movies, About, Register, and Login (which is underlined). The main content area has a dark background with a white login form. The form title is "Login" and it says "Fill in the data below." It contains two input fields, one with "admin@gmail.com" and another with three dots (...), followed by a blue "Login" button.

About Us

Writing a movie review is a great way to express your opinion of a movie. The purpose of most movie reviews is to help readers determine if they want to watch, rent, or buy the movie.

Recent Review

Godzilla WeakStory
Maleficent High Performance
Luca Funny Movie

Help Center

Contact Us Send Us Message about Your problem
Send Us E-mail : Movie Rvu

Join Us

To Join Our Team Send us Your Information To Our E-mail Movie Rvu Email Address

Social Media

Facebook
Twitter
Google+

Newsletter

Email Address →

Copyright 2022 © All rights reserved , Designed by Movie Rvu Team.

صفحة إنشاء مستخدم جديد



MOVIE REVIEW
Make Your Review Share Us Your Opinion
On Recent Movies

Home | Movies | About | Register | Login |

Register Today

Fill in the data below.

Full Name

E-mail Address

Password

Type

Register

About Us

Writing a movie review is a great way to express your opinion of a movie. The purpose of most movie reviews is to help readers determine if they want to watch, rent, or buy the movie.

Recent Review

Godzilla WeakStory
Maleficent High Performance
Luca Funny Movie

Help Center

Contact Us Send Us Message about Your problem
Send Us E-mail : Movie Rvu

Join Us

To Join Our Team Send us Your Information To Our E-mail Movie Rvu Email Address

Social Media

Facebook
Twitter
Google+

Newsletter

Email Address →

الصفحة الرئيسية للوحة التحكم

تظهر الصفحة الرسمية للوحة التحكم بعد عمل login وفيها:

- عدد categories
- عدد الأفلام
- عدد المستخدمين
- عدد reviews
- كل صفح لوحدة التحكم الأخرى navigation links

The screenshot shows the AdminLTE movie review dashboard. On the left is a dark sidebar with user information and navigation links for Category, User, Movie, and Review. The main area has a header with a user icon and the text "Movie Review". Below the header are four colored boxes showing statistics: 10 Categories (teal), 7 Movies Count (green), 6 User Registrations (yellow), and 0 Review Count (red). Each box has a "More info" button. At the bottom, there is a copyright notice and a version number.

Movie Review

admin

Category User Movie Review

Home Web Site Logout

10	7	6	0
Categories	Movies Count	User Registrations	Review Count
More info ↗			

Copyright © 2014-2019 AdminLTE.io. All rights reserved.

Version 3.0.4

صفحة ال Categories

فيها جدول بكل ال categories وهي تصنيفات الأفلام لابد من وجود تصنيفات لاماكنية صناعة الأفلام يمكن إنشاء category او التعديل عليها او حذفها.

The screenshot shows the 'Movie Review' application's interface. On the left is a dark sidebar with user information ('admin') and navigation links: 'Category' (selected), 'Categories', 'User', 'Movie', and 'Review'. The main content area has a header with 'Home', 'Web Site', and 'Logout' links. Below is a title 'Categories' and a green 'Create New Category' button. A table lists categories with columns for '#', 'Name', 'Edit' (with a pencil icon), and 'Delete' (with a trash icon). The table contains five rows: Action (id 16), Drama (id 17), Horor (id 18), Science Fiction (id 19), and Documentary (id 20). At the bottom, it says 'Showing 1 to 10 of 57 entries'. The footer includes copyright information ('Copyright © 2014-2019 AdminLTE.io. All rights reserved.') and a version number ('Version 3.0.4').

#	Name	Edit	Delete
16	Action		
17	Drama		
18	Horor		
19	Science Fiction		
20	Documentary		

صفحة انشاء

يمكن من هنا انشاء category بادخال الاسم

The screenshot shows a dark-themed administrative interface for 'Movie Review'. On the left is a sidebar with a user icon labeled 'admin' and four main menu items: 'Category', 'User', 'Movie', and 'Review', each with a corresponding icon. The main content area has a blue header bar with the text 'Create New Category'. Below the header is a form with a single input field labeled 'Name' and a blue 'Submit' button. At the bottom of the page, there is a copyright notice 'Copyright © 2014-2019 AdminLTE.io. All rights reserved.' and a 'Version 3.0.4' link.

Movie Review

≡ Home Web Site Logout

Create New Category

Name

Submit

Copyright © 2014-2019 AdminLTE.io. All rights reserved.

Version 3.0.4

جدول المستخدمين

فيها جدول بكل المستخدمين يمكن إنشاء المستخدم او التعديل عليه او حذفه

Movie Review

admin

- Category
- User
- User Table
- Movie
- Review

Users Table

Create New User

#	Name	Email	Phone	Type	Image	Edit	Delete
5	editor	editor@editor.com		editor			
8	ahmed	ahmed@ahmed.com		editor			
9	admin	admin@admin.com		admin			
10	ali	ali@ali.com		viewer			

Showing 1 to 10 of 57 entries

صفحة إنشاء المستخدم

Movie Review

≡ Home Web Site Logout

admin

- Category
- User
- Movie
- Review

Create New User

Name

Email

Password

Phone

Image

Choose File No file chosen

Type

Submit

Copyright © 2014-2019 AdminLTE.io. All rights reserved.

Version 3.0.4

صفحة الأفلام

فيها جدول بكل الأفلام يمكن من هنا إنشاء أو تعديل أو حذف أي فيلم لا يظهر في الموقع الرئيسي إلا إذا كان مقبولاً من قبل الـ admin فإذا قام مثلاً الـ editor بإنشاء فيلم فإنه لن يتم عرضه إلى بعد أن يقوم admin بعرضه.

Movie Review

admin

Category

User

Movie

Movie Table

Review

Products Table

Create New Product

#	Name	Is_Approve	Category	User	Image	Edit	Delete
13	Iron Man	Approved	Action	admin			
14	Avatar	Approved	Action	admin			
15	Venom	Approved	Action	admin			
16	Inception	Approved	Science Fiction	admin			
17	Harry poter	Approved	Drama	admin			

Showing 1 to 10 of 57 entries

< 1 2 >

صفحة إنشاء الأفلام

Movie Review

admin

Category User Movie Review

Home Web Site Logout

Create New Product

Name

Image

Choose File No file chosen

X F B C S | T x | : = | ; = | , , | Styles Format ?

Is_Approved

Category

Action

User

editor

Submit

This screenshot shows a web-based application for managing movie reviews. On the left, there's a dark sidebar with icons for 'Category', 'User', 'Movie', and 'Review', and a user profile for 'admin'. The main area has a blue header bar with the text 'Create New Product'. Below it, there are fields for 'Name' (with a text input), 'Image' (with a file upload button showing 'No file chosen' and a rich text editor toolbar), and checkboxes for 'Is_Approved' and 'Category' (with a dropdown menu for 'Action'). There's also a 'User' dropdown set to 'editor'. At the bottom right is a blue 'Submit' button.

صفحة الأفلام في الموقع الرئيسي

يتم عرض كل الأفلام التي قبلها ال admin ويمكن من هنا الضغط على الفلم لفتح صفحة الفيلم.

MOVIE REVIEW
Make Your Review Share Us Your Opinion
On Recent Movies

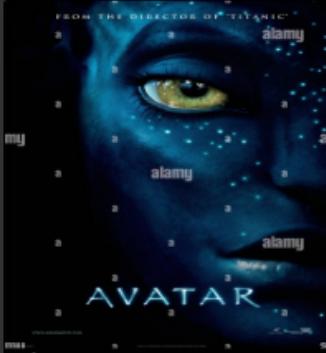
Home | **Movies** | About | Admin Dashboard | Logout |

Recent Movies



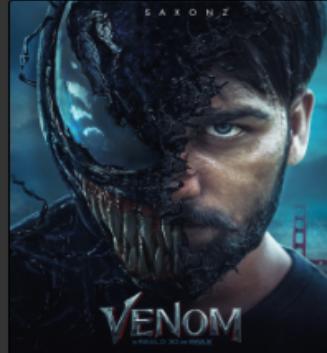
Iron Man
By :admin

[Show Movie](#)



Avatar
By :admin

[Show Movie](#)



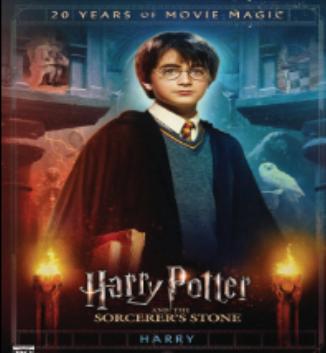
Venom
By :admin

[Show Movie](#)



Inception
By :admin

[Show Movie](#)



Harry poter
By :admin

[Show Movie](#)



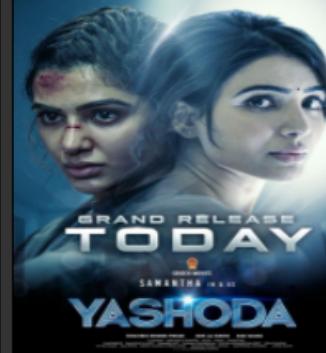
Robots
By :editor

[Show Movie](#)



Smile
By :admin

[Show Movie](#)



Yashoda
By :admin

[Show Movie](#)

صفحة الفيلم الواحد

يمكن فيها رؤية معلومات اكثر عن الفلم

MOVIE REVIEW
Make Your Review Share Us Your Opinion
On Recent Movies

Home | Movies | About | Admin Dashboard | Logout |



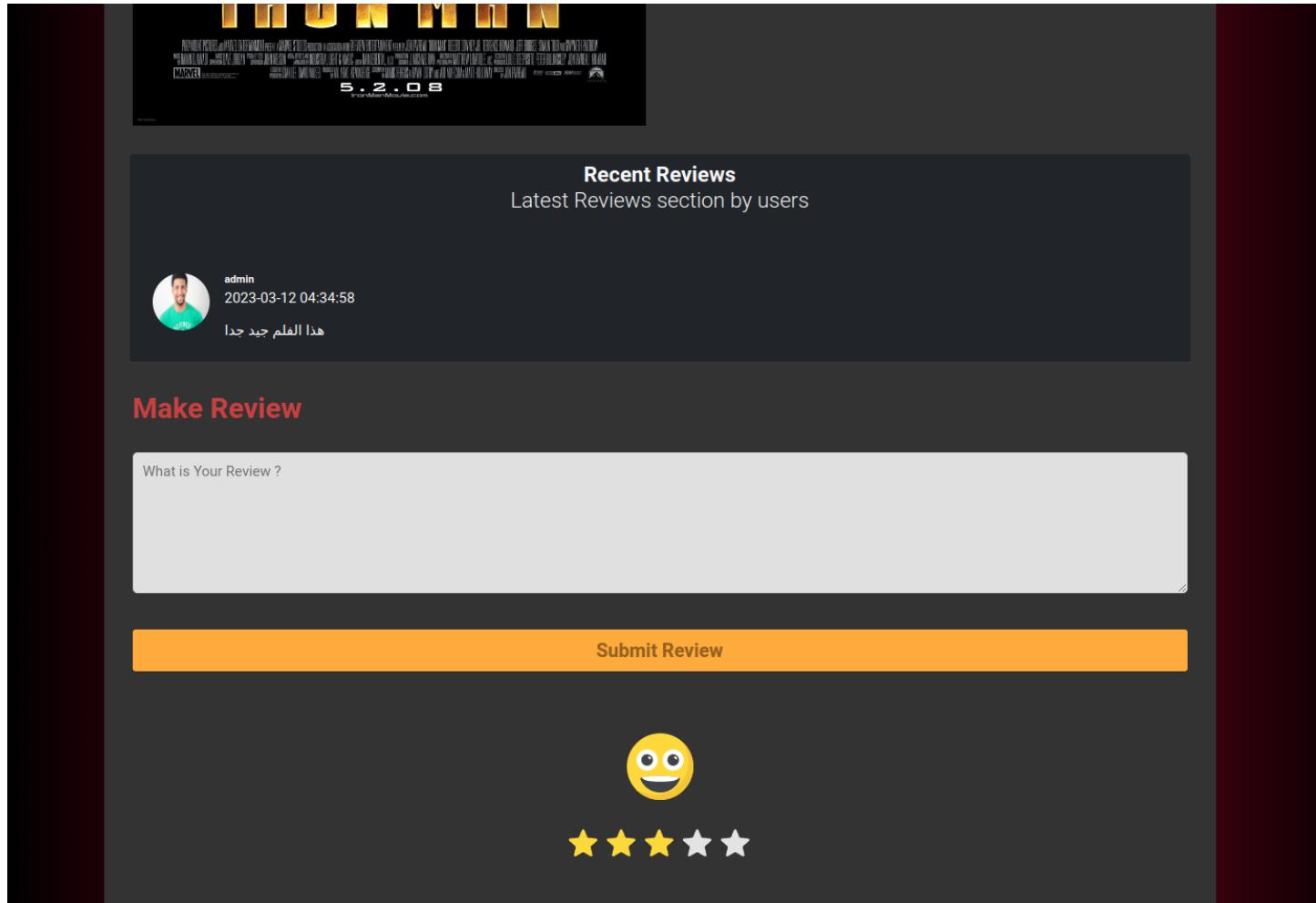
The movie poster for Iron Man features Tony Stark (Robert Downey Jr.) in his Iron Man suit, looking intensely at the viewer. In the background, other characters are visible: Obadiah Stane (Terrence Howard), Pepper Potts (Gwyneth Paltrow), and Rhodey (Jeff Bridges). The poster is set against a dark, dramatic background with glowing blue energy fields.

Iron Man

هو فيلم من إنتاج عام 2008، يستند (بالإنجليزية) Iron Man إلى أحد قصص مارفل كومكس الكرتونية المصورة التي تحمل نفس الاسم [24]. الفيلم هو الأول ضمن عالم مارفل السينمائي من إخراج جون فافرو مع ستاربو من كتابة مارك فراغن وفوك اوسنبي وارنر ماركون ومانتو هولاوي. الفيلم من بطولة روبرت داوني جونيور بدور توني ستارك / آيرون مان وجينيفر بالترو ودينيس هاوارد وبيف بيردجر.

Category: Action
User: admin

وأيضاً إضافة review على الفلم



A screenshot of a movie review interface. At the top, there is a movie poster for "The King's Speech" with the rating "5.2.08". Below the poster, a section titled "Recent Reviews" displays a single review from a user named "admin" posted on "2023-03-12 04:34:58". The review text is "هذا الفلم جيد جدا". Below this section, there is a "Make Review" button and a text input field asking "What is Your Review ?". A large orange "Submit Review" button is positioned below the input field. At the bottom center, there is a yellow smiley face emoji and a row of five stars, with the first four stars filled and the fifth star outlined.

Recent Reviews
Latest Reviews section by users

admin
2023-03-12 04:34:58
هذا الفلم جيد جدا

Make Review

What is Your Review ?

Submit Review

★★★★★

About

فيها معلومات عامة عن الموقع

The screenshot shows a movie review website with a red header. On the left is a camera icon and the text "MOVIE REVIEW: Make Your Review Share Us Your Opinion On Recent Movies". On the right is a navigation bar with "Home", "Movies", "About" (underlined), "Admin Dashboard", and "Logout". Below the header is a banner with a red background featuring a film strip, popcorn, and a clapperboard with "HOME CINEMA" and "MOVIE NIGHT". The main content area contains text about movie buffs saving money by reading reviews, followed by a paragraph about commercial waste.

There is one simple way through which movie buffs can actually save their money. From our movie review site, movie buffs come to know people's opinions about a particular movie. On the basis of the review a movie buff can take his decision (whether to watch it in the theater or categorize it as a flop movie).

Television commercials on new releases are enough to entice millions of movie lovers who run off to the movie theatres to watch those movies. These movie commercials are created to hoodwink the audience. A commercial tends to show that a movie is simply amazing, whereas when a movie buff actually sees it, he finds it to be a complete flop picture. Just by watching these commercials, movie buffs run to theaters and waste their money to watch flop films that do not even stay in the theaters for a week. .

Vision & Mission

we offer the best reviews of the newly released films. The registration process is simple and once you are a registered member, you can read free film reviews and write and submit your own reviews as well. A bad movie can spoil your mood and interest. Moreover, it also burns a hole in your pocket as you spend dollars to watch a flop movie. So it is always better to visit our movie review site, read the review of a newly released movie and then, decide whether it is worth watching in the theater.

our movie review website are free site where people can visit and read reviews any time they wish. The user can also register to some of the movie sites. This allows him to write reviews about the films he has watched. After all, others might also be interested to know about the films that you have watched.Registration is very simple procedure. All you need to do is to follow the steps and get your name registered in the particular site.

Our Team

Ahmed Oweid
Team Leader

Mahmoud
Front-End

Koloud Abd-Elfattah
Front-End

Best Websites for watching movies

- Crackle
- Popcornflix
- Yidio
- YouTube
- NetFlix
- EgyBest
- Vudu

الصفحة الرئيسية للموقع

فيها مجموعة من الأفلام

MOVIE REVIEW
Make Your Review Share Us Your Opinion
On Recent Movies

[Home](#) | [Movies](#) | [About](#) | [Admin Dashboard](#) | [Logout](#) |

Latest Movies

The interface features a red header bar with a camera icon and movie review text. Below is a navigation bar with links to Home, Movies, About, Admin Dashboard, and Logout. The main content area is titled 'Latest Movies' and displays movie posters for 'VENOM', 'INCEPTION', 'Harry Potter', 'The Muppets', 'SMILE', 'YASHODA', and 'KUNG FU PANDA'. Each poster includes its title and release information.

VENOM
IN THEATERS 10.5.18

INCEPTION
THE DREAM IS REAL
IN THEATERS JULY 16

Harry Potter
20 YEARS OF MOVIE MAGIC
HARRY

THE MUPPETS
Repair For Adventure!

SMILE
ONLY IN THEATERS SEPTEMBER 30

YASHODA
GRAND RELEASE TODAY

KUNG FU PANDA
PREPARE FOR AWESOMENESS

وايضا فيها مجموعة من التعليقات



Making Reviews

Writing A Movie Review Is A Great Way To Express Your Opinion Of A Movie. The Purpose Of Most Movie Reviews Is To Help Readers Determine If They Want To Watch, Rent, Or Buy The Movie. The Review Should Give Enough Details About The Movie So That The Reader Can Make An Informed Decision, Without Giving Anyway Any Essentials Such As The Plot Or Any Surprises.

Views Analyze The Effectiveness Of The Plot, Theme, Acting, Direction, Special Effects, Musical Effects, Cinematography, And All Other Elements That Created The Movie. There Are Qualities And Guidelines That A Critique Of A Movie Should Possess. Avoid The Use Of Generalized Opinions Such As "It Was A Great Movie" Or "The Acting Was Horrible," But Rather Give Specific Reasons And The Whys.

Latest Reviews

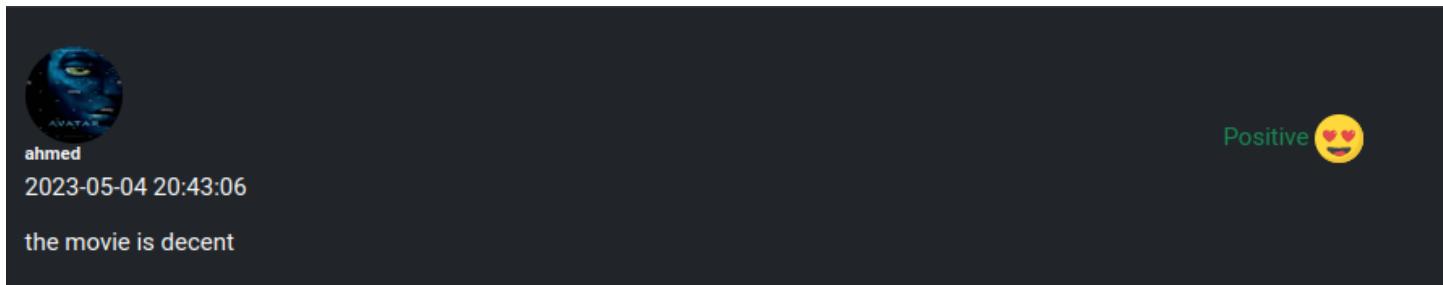
A circular profile picture of a man with short brown hair, wearing a green t-shirt. He is smiling slightly.

Movie Name : Iron Man

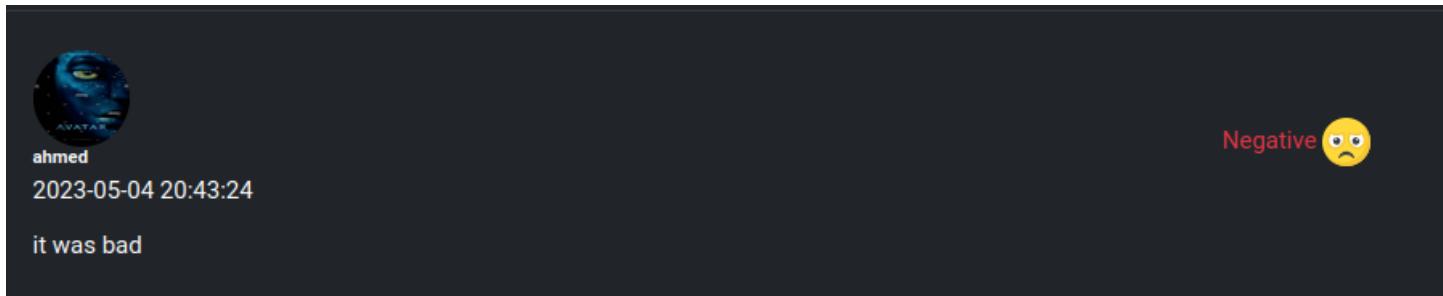
“ هذا الفلم جيد جدا ”

★★

A positive comment



A negative comment





الفصل الرابع

ملخص الدراسة

الفصل الرابع

ملخص الدراسة

1. ملخص الدراسة

2. المراجع

ملخص الدراسة

هذه الدراسة كانت بعنوان Movies Reviews Classification ، كان الهدف الأساسي منها هو التعرف على مجال الذكاء الاصطناعي وتعلم الآلة عموماً و مجال ال Sentiment Analysis خصوصاً واستغلال التقدم الواسع فيهم لتقديم خدمة سهلة للعملاء مما يزيد من جودة الواقع الإلكتروني وكانت مشكلة الدراسة أن العميل كان يستهلك وقتاً طويلاً لكتابة التعليق على المنتجات وأيضاً التقييم عليها مما يجعل معظم العملاء أن لا يقوموا بالتعليق أو التقييم أو كلامها ، وفي هذه الدراسة قمنا بإنشاء موقع لعرض الأفلام كمنتجات مع وجود أدمين لإدارة الموقع وجود Editor لإنشاء المحتوى ووجود viewer وهو العميل الذي يبدي رأيه في المحتوى لكي نحاكي الواقع الإلكتروني على شبكة الإنترنت قمنا ببناء نموذج تعلم آلة باستخدام لغة python باستخدام خوارزمية تعرف ب VADER للتعرف ما إذا كان التعليق سلبي أو إيجابي وبناءً عليه نعرض للمستخدم نتيجة تعليقه ، وهذا أدى إلى جودة أكبر في تجربة المستخدمين للموقع مما شجع عدد أكبر من المستخدمين من استخدام وقتهم في التعليق على المنتجات وبالتالي وجود feedback أكثر لأصحاب هذه المنتجات.

المصادر

[الذكاء الاصطناعي](#)

[اهداف الذكاء الاصطناعي](#)

[خصائص الذكاء الصناعي](#)

[التعلم الآلي](#)

[تاريخ التعلم الآلي](#)

[خصائص تعلم الآلة](#)

[sentiment analysis](#)

[Xampp](#)

[Composer](#)

[Laravel](#)

[laravel directory structure](#)

[Laratrust](#)

[Sentiment Analysis](#)

