

موقع تواصل اجتماعي

مشروع الفصلي

إعداد:

حيدر ه سلوم

اية صالح

حيدر ه غانم

إشراف:

الدكتور رائد الجابري

مقدمة:

يعتبر الفيسبوك أكثر شبكات التواصل الاجتماعي انتشاراً على الانترنت، وذلك من خلال مايزيد عن ملياري مستخدم فعال بشكل شهري على هذا البرنامج، ويحتوي الفيسبوك على العديد من الميزات المتنوعة كتطبيق المسنجر الذي يعتبر ثاني أشهر تطبيقات المراسلة حول العالم ويتيح الفيسبوك لمستخدميه إمكانية مشاركة الصور والأحداث أو حتى الفيديوهات مع المستخدمين الآخرين والتواصل مع الأصدقاء وغيرها من الأمور الأخرى، ومن هنا أتت فكرة إنشاء موقع إلكتروني شبيهه بالفيسبوك يحوي على العديد من الميزات التي يحويها الفيسبوك وتوفير بيئة تواصل مناسبة بين المستخدمين.

الفهرس

الفصل الاول

| | |
|--------------|---------------------------------------|
| ٧..... | الفصل الاول |
| ٧..... | ١-١ مدخل الى المشروع..... |
| ٧..... | ٢-١ المتطلبات الوظيفية للمستخدم |
| ٧..... | ٣-١ المتطلبات غير الوظيفية |
| ٧..... | ١-٣-١ متطلبات المخدم |
| ٨..... | ٢-٣-١ متطلبات المستخدم |
| ٨..... | ٤-١ الوظائف الرئيسية |
| ٨..... | ١-٤-١ تسجيل الدخول |
| ٨..... | ٢-٤-١ الصفحة الرئيسية..... |
| ٨..... | ٣-٤-١ صفحة المسنجر..... |
| ٩..... | ٥-١ مكونات النظام |
| ٩..... | ٦-١ إعداد البيئة..... |
| ٩..... | ٧-١ الاختبارات |
| ٩..... | ٨-١ التحديات |
| ١٠..... | ٩-١ USE CASE |
| ١١..... | ١٠-١ Class-diagram |
| ١٢..... | ١١-١ Activity-diagram |
| ١٣..... | ١٢-١ State-diagram |
| ١٤..... | ١٣-١ sequence-diagram |
| ١٥ - ١٦..... | ١٤-١ نتائج المشروع |
| ١٧..... | ١٥-١ الآفاق المستقبلية |

الفصل الثاني

| | |
|----|--------------------------------------|
| ١٨ | الفصل الثاني |
| ١٨ | ١-٢ إطار العمل لإنشاء واجهة المستخدم |
| ١٨ | ١-١-٢ ما هو React js |
| ١٨ | ٢-١-٢ مزايا React js |
| ١٨ | ٣-١-٢ عيوب React js |
| ١٩ | ٢-٢ Node js |
| ١٩ | 1-2-2 ما هو Node js |
| ١٩ | ٢-٢-٢ مميزات Node js |
| ٢٠ | ٣-٢-٢ سلبيات Node js |
| ٢٢ | ٣-٢ Express js |
| ٢٢ | ١-٣-٢ ما هو Express js |

٢٤ الفصل الثالث

| | |
|----|--|
| ٢٤ | ١-٣ ربط قاعدة البيانات (Database) مع الـ Backend |
| ٢٥ | ٢-٣ الربط بين الفرونت والباك |
| ٢٦ | ٣-٣ database الخاص بنا |
| ٢٧ | ٤-٣ packages |
| ٢٩ | ١-٤-٣ إنشاء واجهة المستخدم التفاعلية |
| ٢٩ | ٢-٤-٣ login page |
| ٣٠ | ٥-٣ register page |
| ٣١ | ٦-٣ Home page |
| ٣٢ | ٦-٣ User's personal page |
| ٣٣ | ٢-٦-٣ messenger page |

٣٤.....Backend ٣-٦-٣

٣٩.....الخاتمة

٤٠.....المراجع

فهرس الأشكال:

| | |
|---------|----------------------|
| ١٠..... | use case diagram |
| ١١..... | class diagram |
| ١٢..... | Activity-diagram |
| ١٣..... | State-diagram |
| ١٤..... | sequence-diagram |
| ٢٦..... | database |
| ٢٩..... | login page |
| ٣٠..... | register page |
| ٢٧..... | packages |
| ٣١..... | Home page |
| ٣٢..... | User's personal page |
| ٣٣..... | messenger page |

الفصل الأول

١-١ مدخل إلى المشروع:

- هذا المشروع عبارة عن موقع الكتروني يشبه الى حد ما منصة الفيسبوك. لقد قمنا بتصميم هذا الموقع بحيث يوفر مميزات عديدة منها :
- إنشاء حساب على الموقع وتسجيل الدخول عن طريقه.
 - إمكانية نشر ستوريات و منشورات والتعليق والتفاعل معها.
 - إمكانية إضافة الأصدقاء وإلغاء الصداقة والحظر.
 - إمكانية التواصل مع الاصدقاء عن طريق تطبيق مراسلة.

٢-١ المتطلبات الوظيفية للمستخدم:

- ١ - إنشاء حساب مستخدم
- ٢-تسجيل الدخول
- ٣- إنشاء ملف المستخدم الشخصي
- ٤ - إضافة أصدقاء ومتابعين
- ٥ - إنشاء مشورات
- ٦ - التفاعل مع المنشوات
- ٧- التعديل على الملف الشخصي
- ٨- اختيار الوضع الليلي او النهاري للموق

٣-١ المتطلبات غير الوظيفية:

١-٣-١ متطلبات المخدم:

- نظام تشغيل يدعم رياكت.
- الأداء: يجب أن يكون الموقع سريع الاستجابة وذو واجهة استخدام سهلة.
- الأمان: يجب حماية بيانات المستخدمين وخصوصيتهم.

١-٣-٢ متطلبات المستخدم:

- متصفح إنترنت مثل (google chrome).

1-4 الوظائف الرئيسية:

١ - ٤ - ١ تسجيل الدخول:

إدخال اسم المستخدم وكلمة السر إذا كان الحساب موجودا بالفعل يتم التوجيه الى الصفحة الرئيسية للمستخدم وإذا لم يكن الحساب موجود يتم عرض رسالة خطأ توضح هذا الخطأ.

١ - ٤ - ٢ إنشاء حساب جديد:

لإنشاء حساب جديد يجب ملئ بعض الحقول وهي اسم المستخدم الذي سوف يقوم به بتسجيل الدخول وكلمة السر والإيميل واسم المستخدم الذي سوف يظهر للأصدقاء .

١ - ٤ - ٣ الصفحة الرئيسية :

- قسم الستوريات : يتم عرض ستوريات الأصدقاء في هذا القسم .
- قسم المنشورات : يتم عرض أحدث المنشورات في الموقع في هذا القسم.
- قسم الإشعارات : يتم عرض الإشعارات وطلبات الصداقة الأخيرة .

١ - ٤ - ٤ صفحة المسنجر:

- تعرض المحادثات الخاصة بالمستخدم وتمكن المستخدم من أرشفة المحادثات وتغيير لون الخلفية وإعدادات الألوان والتغيير بين الوضع الليلي والمظلم.

١-٥ مكونات النظام:

- الواجهة الأمامية (react): استخدام react لبناء واجهة المستخدم المتعددة المنصات.
- قاعدة البيانات (node js): استخدام node js لبناء قاعدة البيانات .

١-٦ إعداد البيئة:

- react: إعداد مشروع react باستخدام Visual Studio Code.
- node js: إعداد مشروع node js وربط قاعدة البيانات بالمشروع.

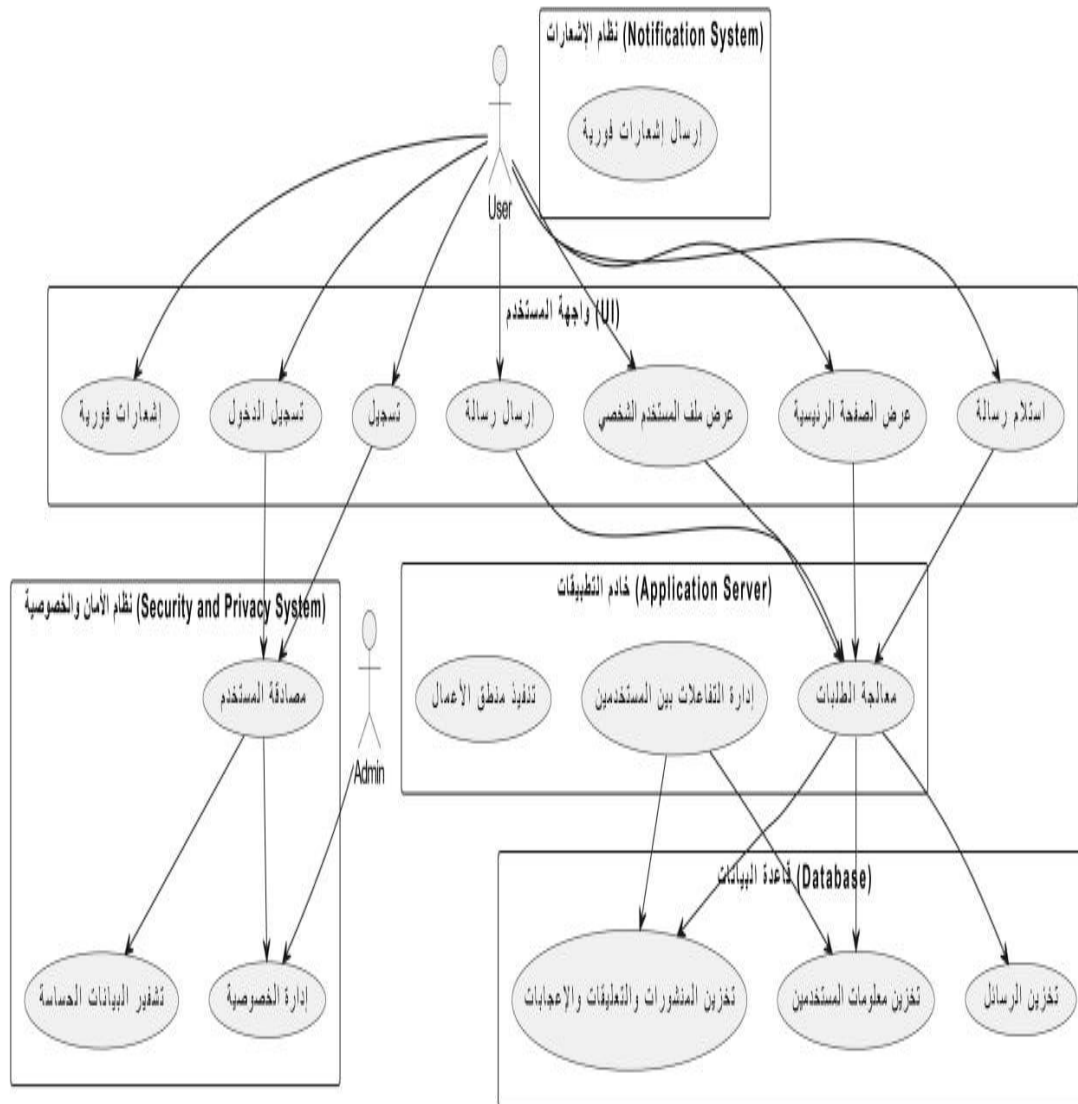
١-٧ الاختبارات:

- اختبارات الوحدات (Unit Tests): اختبار كل مكون و واجهة في react بشكل مستقل.
- اختبارات التكامل (Integration Tests): اختبار تكامل react مع التطبيق.
- اختبارات الأمان: ضمان أمان البيانات ومصادقة المستخدمين بشكل صحيح.

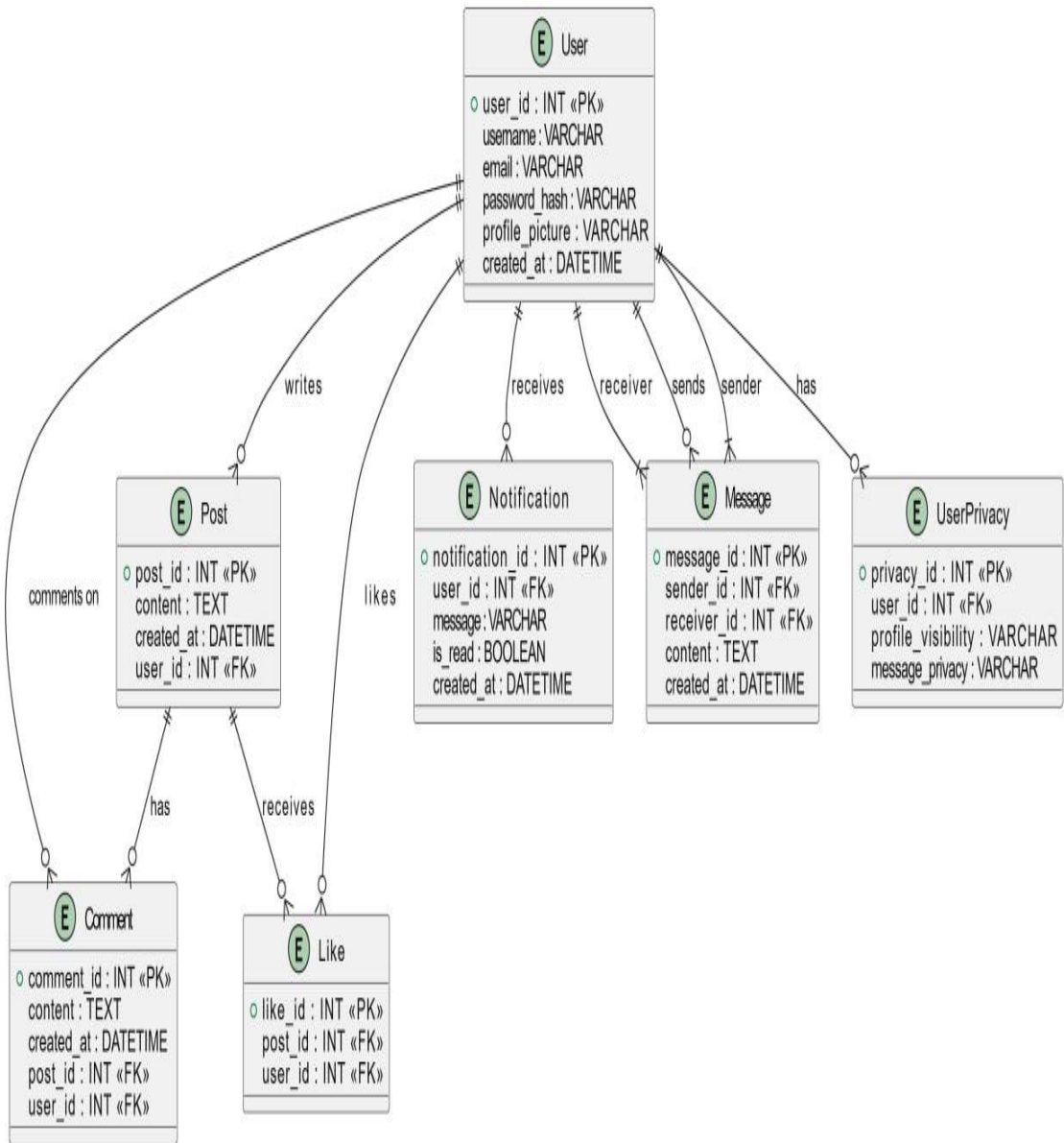
١-٨ التحديات:

- التعامل مع تزامن البيانات: ضمان تحديث البيانات بشكل لحظي.
- الأمان: حماية بيانات المستخدمين من الوصول غير المصرح به.

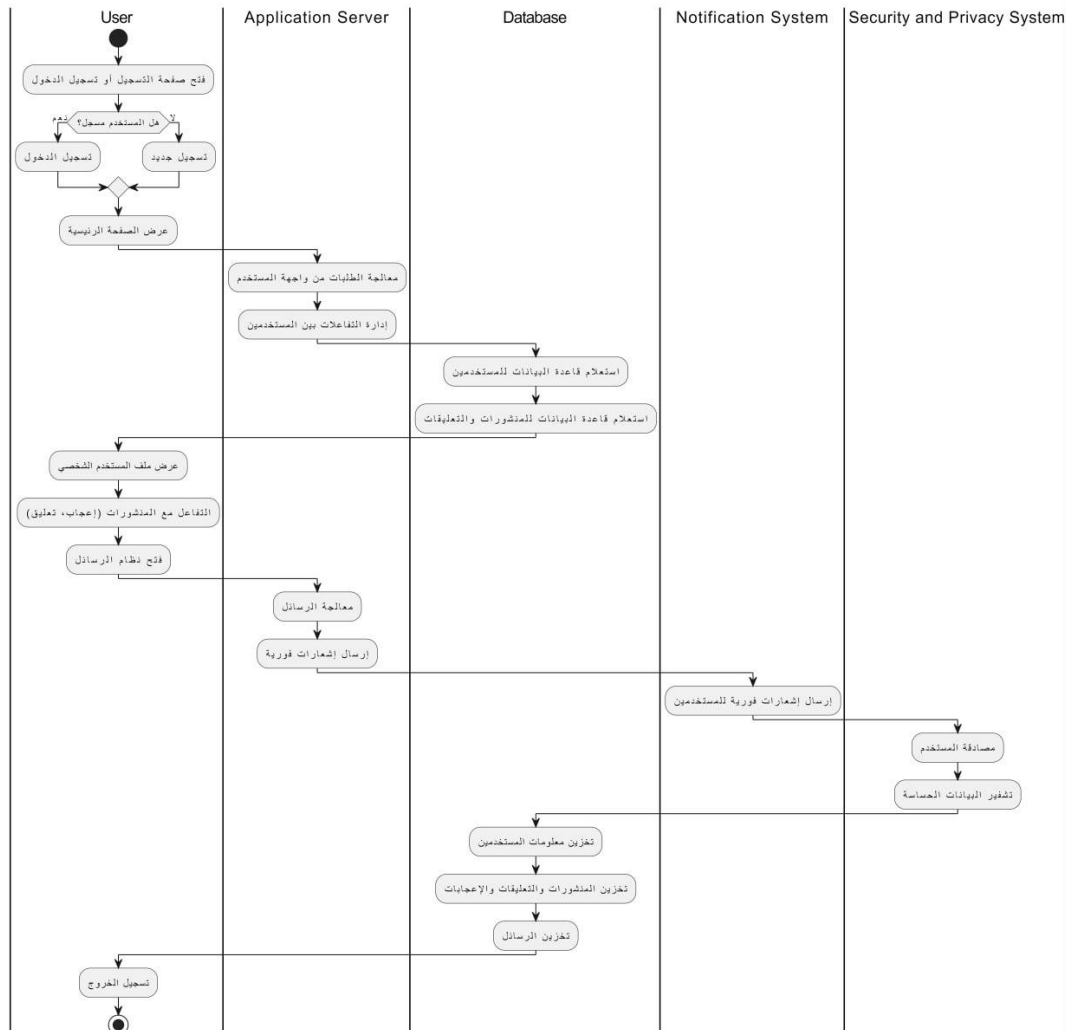
٩-١ :USE CASE DIAGRAM



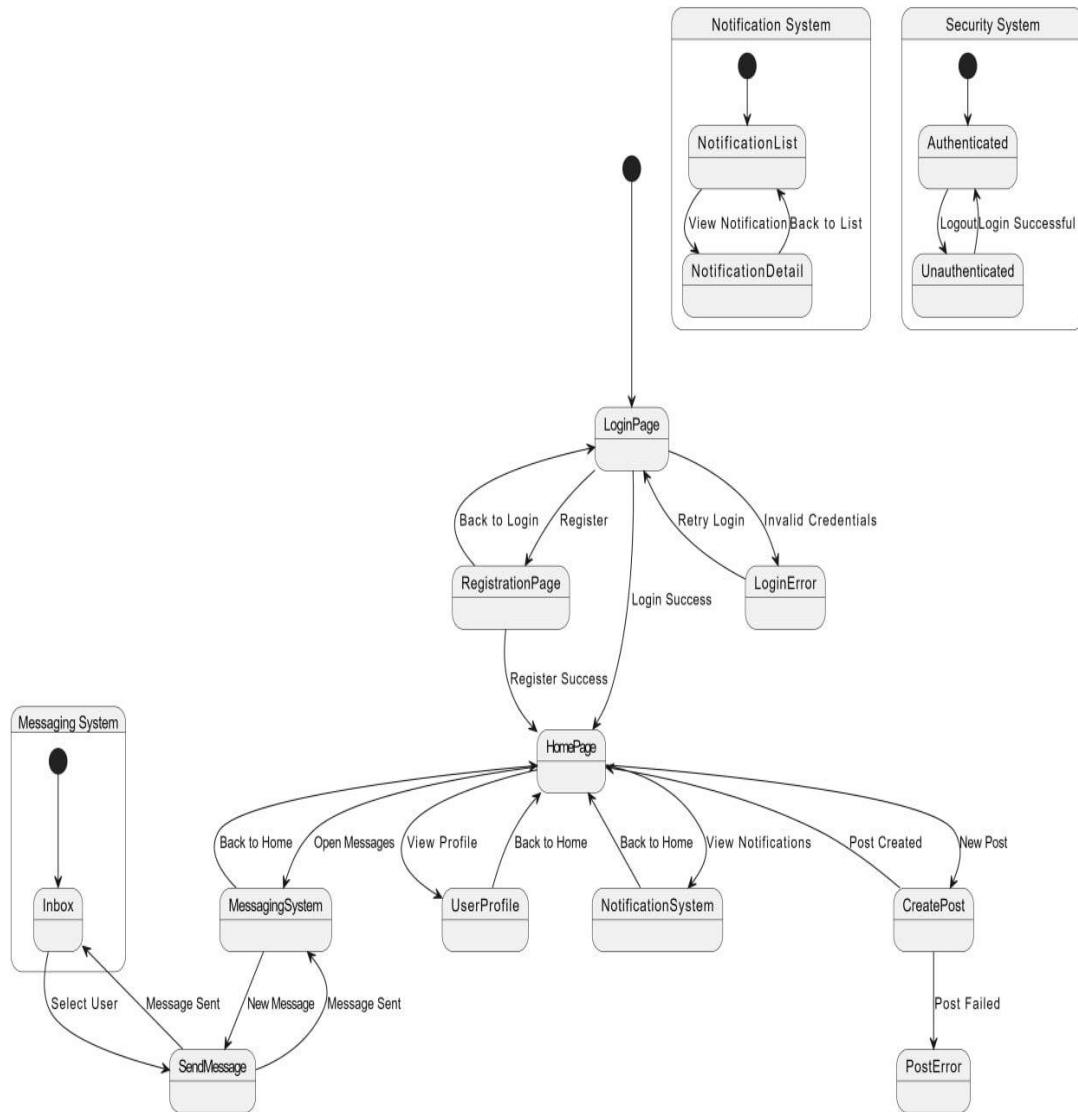
:CLASS DIAGRAM 1.1



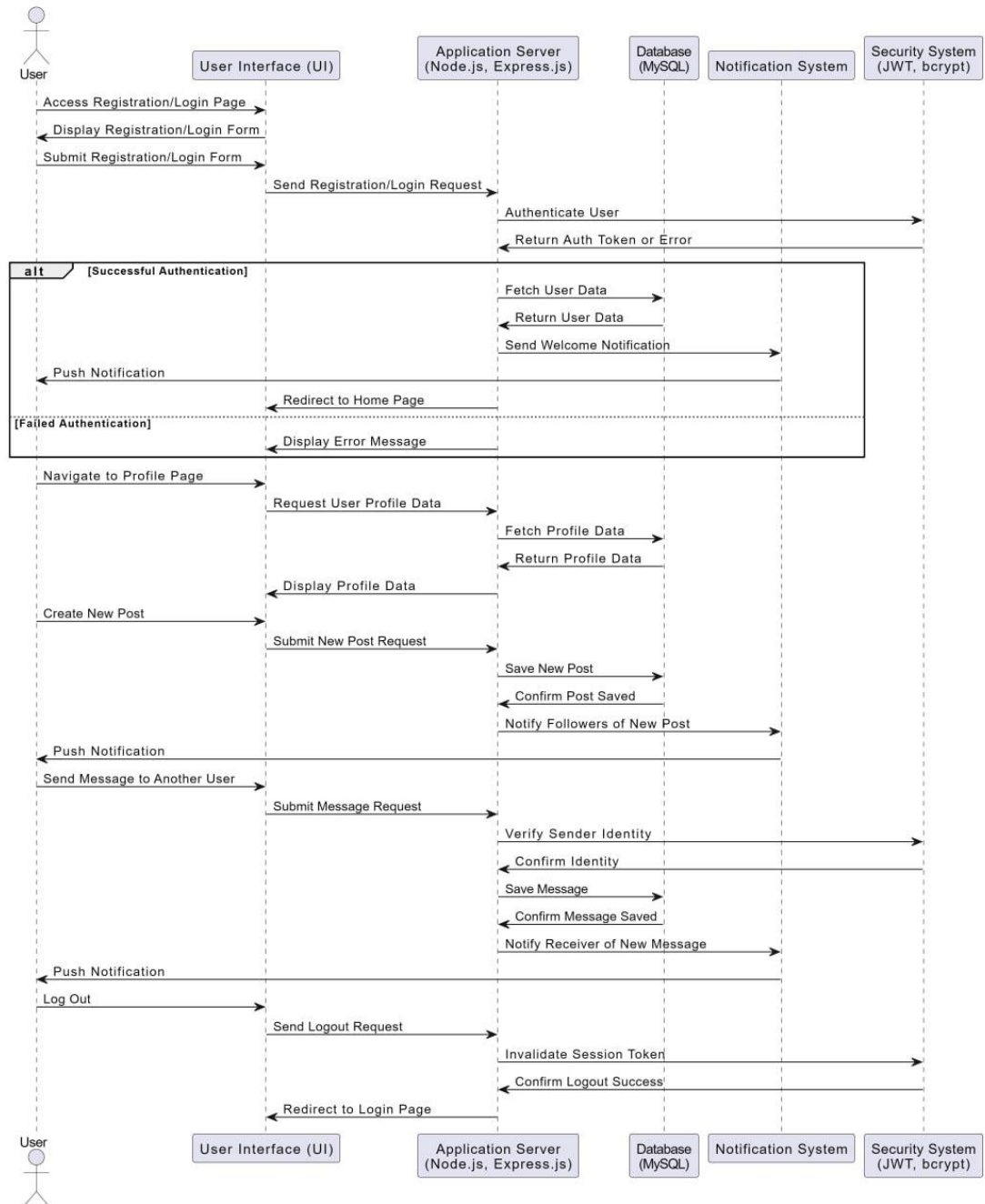
١١-١ :ACTIVITY DIAGRAM



:STATE DIAGRAM ١٢-١



:SEQUENCE DIAGRAM ١٣-١



١-٤ نتائج المشروع:

١. زيادة عدد المستخدمين

- نمو قاعدة المستخدمين: شهد الموقع زيادة ملحوظة في عدد المستخدمين الجدد، مما يدل على جاذبيته وشعبيته.
- نسبة الاحتفاظ بالمستخدمين: ارتفاع في نسبة المستخدمين الذين يعودون للموقع بانتظام، مما يشير إلى رضاهم عن التجربة.

٢. تفاعل المستخدمين

- زيادة التفاعل: ارتفاع في عدد التعليقات والإعجابات والمشاركات، مما يعكس تفاعل المستخدمين مع المحتوى.
- معدل المشاركة: زيادة في نسبة المستخدمين الذين يشاركون المحتوى مع أصدقائهم أو متابعيه E

٣. تحسين تجربة المستخدم

- تقييمات إيجابية: تلقى الموقع تقييمات إيجابية من المستخدمين فيما يتعلق بالواجهة وسهولة الاستخدام.
- تقليل الشكاوى: انخفاض في عدد الشكاوى المتعلقة بالأمان والخصوصية، مما يعكس فعالية التدابير المتخذة.

٤. تنوع المحتوى

- زيادة أنواع المحتوى: تنوع المحتوى المنشور، بما في ذلك النصوص، الصور، والفيديوهات، مما يجذب فئات مختلفة من المستخدمين.
- محتوى مميز: ظهور محتوى متميز من قبل المستخدمين، مما يعزز من جودة المنصة.

٥. الأمان والخصوصية

- تحسين الأمان: تنفيذ ميزات أمان جديدة ساهمت في حماية بيانات المستخدمين، مما زاد من ثقتهم في المنصة.
- خيارات الخصوصية: تقديم خيارات تخصيص متقدمة للمستخدمين للتحكم في معلوماتهم الشخصية.

٦. الإيرادات

- نمو الإيرادات: تحقيق إيرادات من خلال الإعلانات والشراكات الاستراتيجية.
- نموذج أعمال مستدام: تطوير نموذج أعمال يضمن الاستدامة المالية للمنصة.

٧. التوسع

- دخول أسواق جديدة: بدء التوسع إلى أسواق جديدة مع استراتيجيات تسويقية محلية.
- شراكات استراتيجية: بناء شراكات مع شركات أخرى لتعزيز النمو والتوسع.

الخلاصة

نجح مشروع موقع التواصل الاجتماعي في تحقيق أهدافه الرئيسية من خلال زيادة عدد المستخدمين، وتعزيز التفاعل، وتحسين تجربة المستخدم، بالإضافة إلى تحقيق نتائج مالية إيجابية. هذه النتائج تشير إلى أن المشروع يسير في الاتجاه الصحيح نحو تحقيق رؤيته المستقبلية.

١-١٥ آفاق المستقبلية للمشروع:

الرؤية المستقبلية لمشروع موقع تواصل اجتماعي

الهدف: إنشاء منصة تواصل اجتماعي تعزز التواصل الفعال والمشاركة الإيجابية بين الأفراد

المميزات الأساسية

- تجربة مستخدم متميزة: واجهة بسيطة وسهلة الاستخدام مع خيارات تخصيص
- الأمان والخصوصية: حماية بيانات المستخدمين ومنحهم السيطرة على معلوماتهم .
- محتوى متنوع: دعم أنواع مختلفة من المحتوى مثل النصوص، الصور، والفيديوهات .
- التفاعل والمشاركة: ميزات لتعزيز التفاعل مثل التعليقات والإعجاب .
- ابتكار مستمر: استخدام تقنيات مثل الذكاء الاصطناعي لتحسين التجربة

النمو والتوسع: استهداف أسواق جديدة وبناء شراكات استراتيجية

الاستدامة: نموذج إيرادات مستدام مع التركيز على المسؤولية الاجتماعية

قياس النجاح: تحديد مؤشرات أداء واضحة لقياس التقدم ورضا المستخدمين

الهدف النهائي

أن يصبح الموقع منصة رائدة في تعزيز التواصل الإيجابي والمشاركة الفعالة، مما يساهم في بناء مجتمع متفاعل ومتربط

الفصل الثاني

٢-١ إطار العمل لإنشاء واجهة المستخدم *React Js*

٢-١-١ ما هو *React js* :

هو مكتبة تُستخدم لبناء واجهات المستخدم، وتحديداً لتطبيقات الويب. ويستخدمه العديد من الشركات الكبرى بسبب كفاءته وسرعته.

Facebook تم تطويره بواسطة *React.js*

٢-١-٢ مزايا *React js* :

يستخدم *ReactJS* نطاقاً ظاهرياً ، والذي يستخدم ذاكرة تخزين مؤقت لهيكل البيانات في الذاكرة ، ويتم تحديث أحدث التعديلات فقط في نطاق المتصفح . هذا يسرع التطبيق .

باستخدام ميزة مكون التفاعل ، يمكنك تصميم مكونات من اختيارك . المكونات قابلة لإعادة الاستخدام ومفيدة لصيانة التعليمات البرمجية .

نظراً لأن *Reactjs* هي مكتبة جافا سكريبت مفتوحة المصدر ، فمن السهل تعلمها .

اكتسب *ReactJS* شعبية سريعاً وهو مدعوم من Facebook و Instagram . تستخدمه العديد من الشركات المعروفة ، بما في ذلك Apple و Netflix .

نظراً لأن Facebook يحتف بمكتبة *ReactJS* ، فهي مُدارة جيداً ومُحدّثة .

يمكن استخدام *ReactJS* لإنشاء واجهات مستخدم متطورة لكل من تطبيقات سطح المكتب والأجهزة المحمولة

٢-١-٣ عيوب *React js* :

تمت كتابة غالبية التعليمات البرمجية بلغة *JSX* ، مما يعني أن *HTML* و *CSS* جزء من كود جافا سكريبت. قد يكون هذا محيراً لأن معظم الأطر الأخرى تحب إبقاء *HTML* منفصلاً عن كود جافا سكريبت .

يحتوي *ReactJS* على حجم ملف ضخم.

٢-٢ Node Js:

٢-٢-١ ما هو Node Js:

Node.js ليس في الواقع إطار عمل أو مكتبة، ولكنه بيئة تشغيل، تعتمد على محرك V8 JavaScript من Chrome. تم تبني هذه التقنية لاحقاً من قبل عدد من رواد التكنولوجيا على سبيل المثال لا الحصر، eBay و Walmart و Netflix. عبارة عن نظام أساسي مبني على وقت تشغيل JavaScript في Chrome لبناء تطبيقات شبكة سريعة وقابلة للتطوير بسهولة. يستخدم Node.js نموذج إدخال / إخراج يحركه الحدث ولا يحظره مما يجعله سريع وفعال، ومثالي لتطبيقات الوقت الفعلي كثيفة البيانات التي تعمل عبر الأجهزة الموزعة. Node.js هي بيئة تشغيل مفتوحة المصدر تعمل عبر الأنظمة الأساسية لتطوير تطبيقات الخادم والشبكات. تتم كتابة تطبيقات Node.js بلغة JavaScript، ويمكن تشغيلها خلال وقت تشغيل Node.js على أنظمة التشغيل OS X و Microsoft Windows و Linux. يوفر Node.js أيضاً مكتبة غنية من وحدات JavaScript المتنوعة التي تبسط تطوير تطبيقات الويب باستخدام Node.js إلى حد كبير. Node.js = بيئة وقت التشغيل + مكتبة جافا سكريبت.

٢-٢-٢ ميزات Node Js:

غير مترامن ومدفوع بالأحداث:

جميع واجهات برمجة التطبيقات لمكتبة Node.js غير مترامنة، أي غير محظورة. هذا يعني بشكل أساسي أن الخادم المستند إلى Node.js لا ينتظر أبداً واجهة برمجة التطبيقات لإرجاع البيانات.

ينتقل الخادم إلى واجهة برمجة التطبيقات التالية بعد استدعائه وتساعد آلية الإخطار بأحداث Node.js الخادم في الحصول على استجابة من استدعاء واجهة برمجة التطبيقات السابق.

سريع جداً

نظراً لكونها مبنية على محرك جافا سكريبت V8 من Google Chrome، فإن مكتبة Node.js سريعة جداً في تنفيذ التعليمات البرمجية.

قابل للتطوير بدرجة كبيرة

يستخدم Node.js نموذجاً واحداً مترابطاً مع تكرار الأحداث. تساعد آلية الأحداث الخادم على الاستجابة بطريقة غير محظورة وتجعل الخادم قابلاً للتوسع بدرجة كبيرة على عكس الخوادم التقليدية التي تنشئ سلاسل عمليات محدودة للتعامل مع الطلبات.

يستخدم Node.js برنامجاً واحداً مترابطاً ويمكن للبرنامج نفسه تقديم خدمة لعدد أكبر من الطلبات مقارنة بالخوادم التقليدية مثل خادم Apache HTTP.

لا يوجد تخزين مؤقت

لا تقوم تطبيقات Node.js أبداً بتخزين أي بيانات مؤقتاً. تقوم هذه التطبيقات ببساطة بإخراج البيانات في أجزاء.

٢-٣-٢ سلبيات Node Js:

واجهة برمجة التطبيقات (API) غير مستقرة

واحدة من القضايا الحرجة التي غالبية تجربة المطور هي أن واجهة برمجة التطبيقات تستمر في التغيير في نطاقات مختلفة ولا تظل مستقرة. في مناسبات أخرى، قد يبدو أن واجهة برمجة تطبيقات أخرى بها تغييرات غير متسقة. وبالتالي، يضطر المهندسون إلى إجراء تغييرات في قواعد التعليمات البرمجية المتاحة لتنسيق التشابه مع أحدث إصدار لواجهة برمجة تطبيقات Node.js API.

Node.JS ليس لديها نظام دعم مكتبة قوي

لا تحتوي JavaScript على إطار مكتبة قوي ومعد جيدا على عكس لهجات البرمجة الأخرى. والنتيجة هي أن العملاء مجبرون على استخدام مكتبة قياسية لتنفيذ تعهدات مختلفة ، على سبيل المثال ، تعيين الكائنات العلائقية (ORM) ، وإعداد الصور ، والاهتمام بمهام قاعدة البيانات ، وتحليل XML. حاول تنفيذ مهام البرمجة القياسية باستخدام Node.js.

لا يوجد نموذج برمجة متزامن

المتطلبات الأساسية هي استقبال نموذج البرمجة الشاذ. ومع ذلك ، قد يرى العديد من المصممين أن نموذج البرمجة هذا مزعج بشكل متزايد من برمجة الإدخال / الإخراج للحظر المباشر.

٢-٣ Express Js:

٢-٣-١ ما هو Express Js:

Express.js هو أحد هذه الحلول التي جاءت لمساعدة عشاق Node.js على تطوير تطبيقات ويب على أساس متين ومنظم، إنه إطار عمل صغير (Micro framework) جلب معه عدد من المزايا التي يحتاجها قطعاً أي مطور في مشاريعه.

يجب الحذر عند مقارنة Express مع إطارات العمل المعروفة للغات برمجية أخرى، مثلاً Symfony أو Laravel بالنسبة لـ PHP و Django للغة بايثون، فهذه إطارات عمل ضخمة ومتكاملة، بينما Express.js هو إطار عمل صغير لا يوفر كل مقومات تلك الأطر.

سنتكلم عن ثلاثة أمور ومزايا مهمة جاء بها إطار العمل Express.js لجعل حياة المطور أسهل وأكثر تنظيماً.

١. Routing:

من المزايا المهمة والأساسية التي يجب على كل إطار عمل توفيرها للمطورين هي إتاحة نظام لقراءة العناوين (routes) وهذه من أجمل الميزات التي نجدها في Express، نظام Routing الخاص به رائع وسهل الفهم والإستيعاب كما سنرى في الأمثلة أسفله.

أول ما يجب القيام به هو تثبيت إطار العمل Express عن طريق الأمر التالي :

```
npm install express
```

٢. نظام القوالب Templates :

يفصل كود الجافاسكريبت عن HTML حيث يضع الأخيرة في ملفات مستقلة جاعلاً بذلك سير العمل أكثر تنظيماً واحترافية.

يمكن استخدام عدة أنظمة للقوالب مع إطار العمل Express، مثلاً Jade و Haml

٣. Middlewares :

ال Middleware (أو الوسيط) هو عبارة عن برنامج صغير له وظيفة معينة ومحددة. وإطار العمل Express مبني على مفهوم ال Middlewares و يوفّر على عدة أنواع منها، نذكر منها ما يلي :

cookie-parser** : **للتعامل مع الكوكيز.

cookie-session** : **للتعامل مع الجلسات (sessions).

CSRF : للحماية من ثغرات CSRF.

في إطار العمل لارافيل هناك مفهوم ال Middlewares كذلك، و Express يقوم باستخدامه بشكل رائع.

الفصل الثالث

العمل الذي قمنا به

٣-١ ربط قاعدة البيانات (Database) مع الـ Backend :

١. تثبيت المكتبات الضرورية:

- في مشروع الـ Backend، تثبيت المكتبات اللازمة للتواصل مع قاعدة البيانات باستخدام Node.js، قمنا بتضمين مكتبة mysql

٢. إنشاء اتصال بقاعدة البيانات:

- إنشاء اتصال بقاعدة البيانات من خلال الكود التالي.

#..

```
import mysql from "mysql2"

export const db = mysql.createConnection({
  host: "localhost",
  port: "3306",
  user: "root",
  password: "0000",
  database: "social"
})

db.connect(err => {
  if (!err) {
    console.log("DB Connection Succeeded");
  } else {
    console.log("DB Connection Failed");
  }
});
```


٢-٣ الربط بين الفرونت والباك :

ربط الـ Frontend مع الـ Backend هو عملية تفاعل بين واجهة المستخدم (Front-end) والخادم (Back-end) لتبادل البيانات:

- تم الربط بينهما من خلال مكتبة axios في frontend

. أنواع الطلبات:

- GET: لجلب البيانات من الخادم.

- POST: لإرسال بيانات جديدة إلى الخادم.

- PUT/PATCH: لتحديث بيانات موجودة.

- DELETE: لحذف بيانات.

ملف axios.js في قسم client ونقوم باستدعاء makeRequest عند الطلب

```
import axios from "axios";

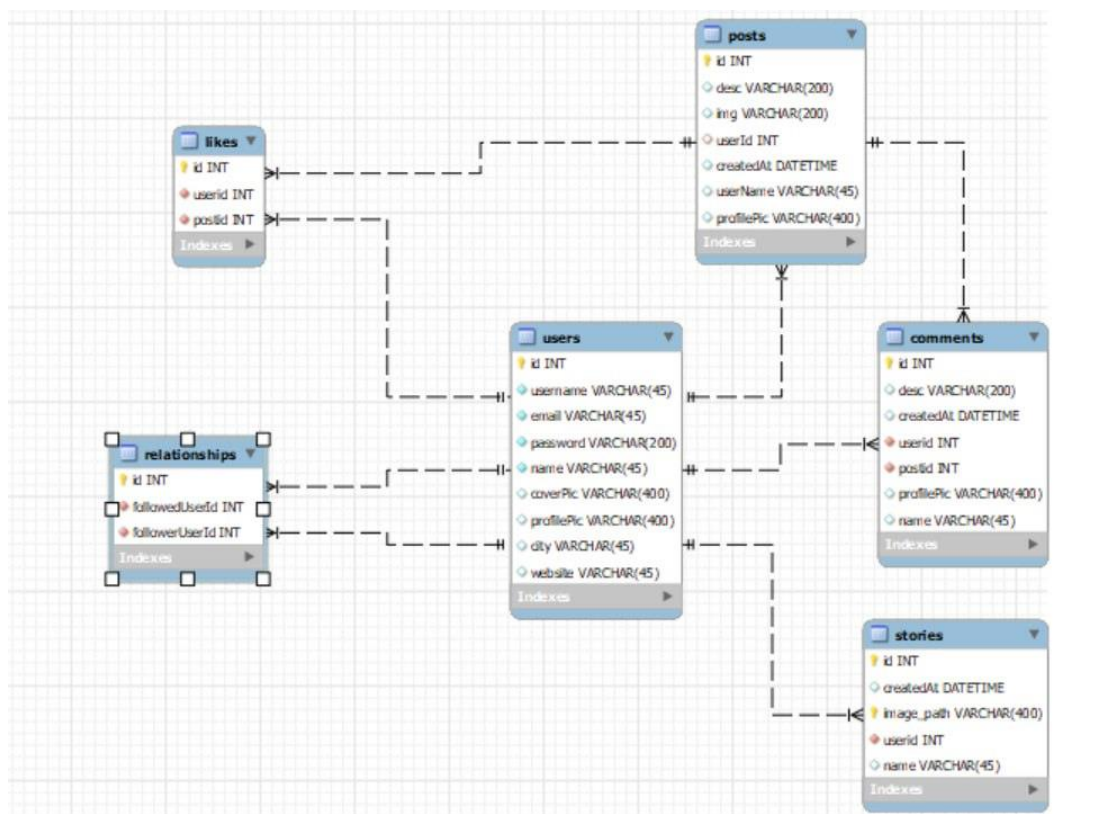
export const makeRequest = axios.create({
  baseURL: "http://localhost:8800/api",
  withCredentials: true,
});
```

مثال عنه عند عرض المنشورات في الصفحة الرئيسة يقوم بإرسال طلب :

```
import { makeRequest } from "../../axios";

Codiumate: Options | Test this function
const Posts = ({ userId }) => {
  // console.log(userId)
  const { isLoading, error, data } = useQuery(["posts", userId], () =>
    makeRequest.get(`/posts`).then((res) => {
      return res.data;
    })
  );
  // console.log(data);
```

٣-٣ ال database لمشروعنا :



٣-٤ ال Packages :

تطلب منا بناء المشروع الحزم (packages) التالية :

١. الواجهة الأمامية (Frontend):

- React: مكتبة لبناء واجهات المستخدم.

- Axios: مكتبة لإجراء الطلبات HTTP.

٢. الواجهة الخلفية (Backend):

- Node.js: بيئة تشغيل لـ JavaScript على الخادم.

- Express.js: إطار عمل لبناء تطبيقات الويب باستخدام Node.js.

٣. قواعد البيانات:

- MySQL: قاعدة بيانات SQL

٤. الأدوات الإضافية:

- JWT (JSON Web Tokens): لتوثيق المستخدمين وتأمين API.

- Multer: لمعالجة تحميل الملفات في تطبيقات Node.js.

٥. الأمان:

- bcrypt.js: لتشفير كلمات المرور.

```
"dependencies": {  
  "bcryptjs": "^2.4.3",  
  "cookie-parser": "^1.4.6",  
  "cors": "^2.8.5",  
  "express": "^4.19.2",  
  "jsonwebtoken": "^8.5.1",  
  "moment": "^2.29.4",  
  "multer": "^1.4.5-lts.1",  
  "mysql": "^2.18.1",  
  "mysql2": "^3.10.1",  
  "nodemon": "^2.0.20"  
}
```

```
"axios": "^1.6.8",  
"moment": "^2.29.4",  
"react": "^18.0.0",
```

٥-٣ إنشاء واجهة المستخدم التفاعلية:

باستخدام React قمنا بإنشاء مجموعة من الصفحات المرتبطة مع بعضها والتي تشكل بمجموعها واجهة المستخدم UI للموقع الإلكتروني خاصتنا:

١-٥-٣: login page

في هذه الصفحة لدينا حقول لتسجيل الدخول اذا كان المستخدم يملك حساب بالفعل وإذا لم يكن لديه حساب يذهب لصفحة إنشاء حساب جديد.

Social Media

Login

register

[Arabic](#) [English \(US\)](#) [Francais \(France\)](#) [Deutsch](#) [Espanol](#)

Social Media

Create a new account

create your account in seconds

haidara@gmail.com

Email

.....

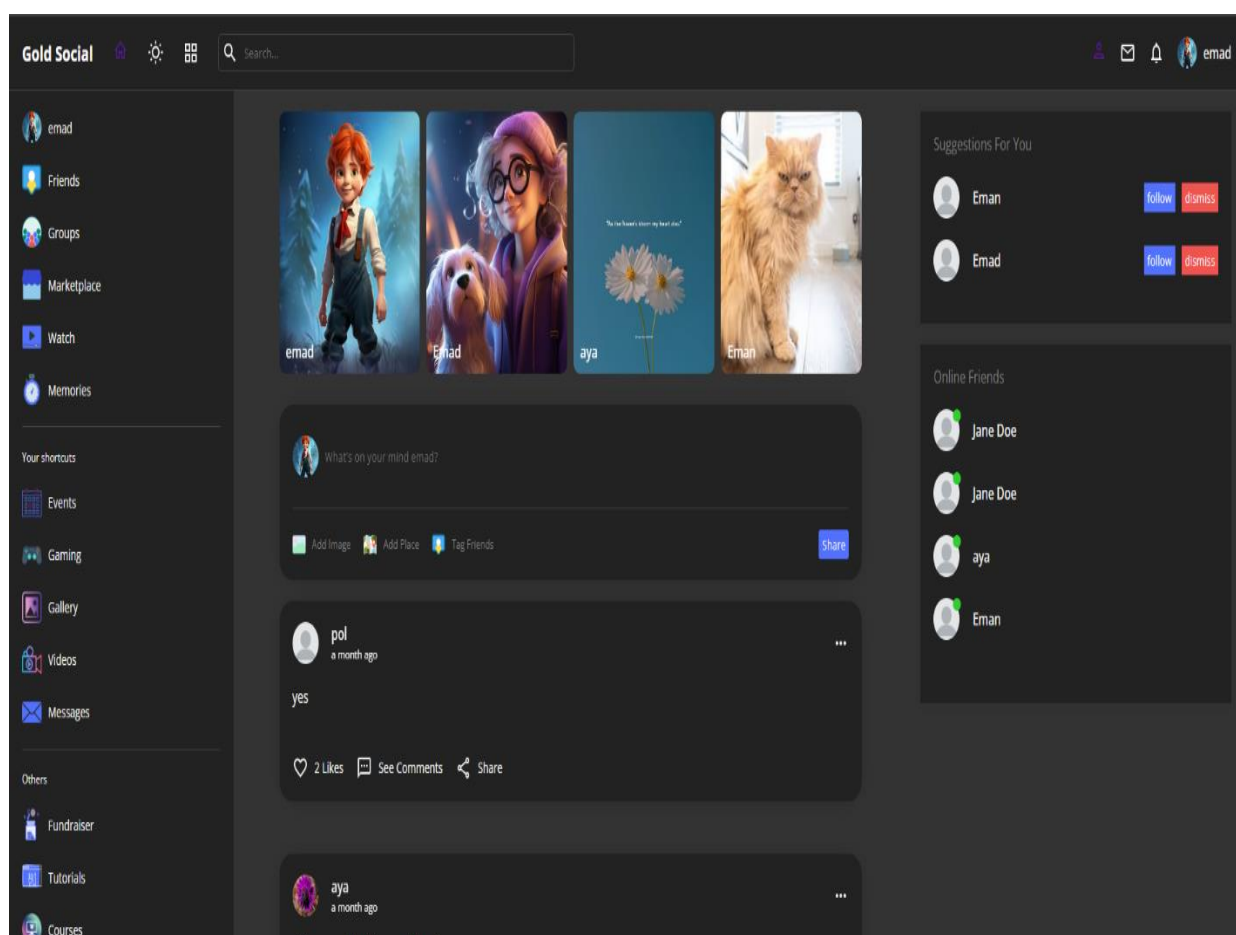
Name

register

i have an account

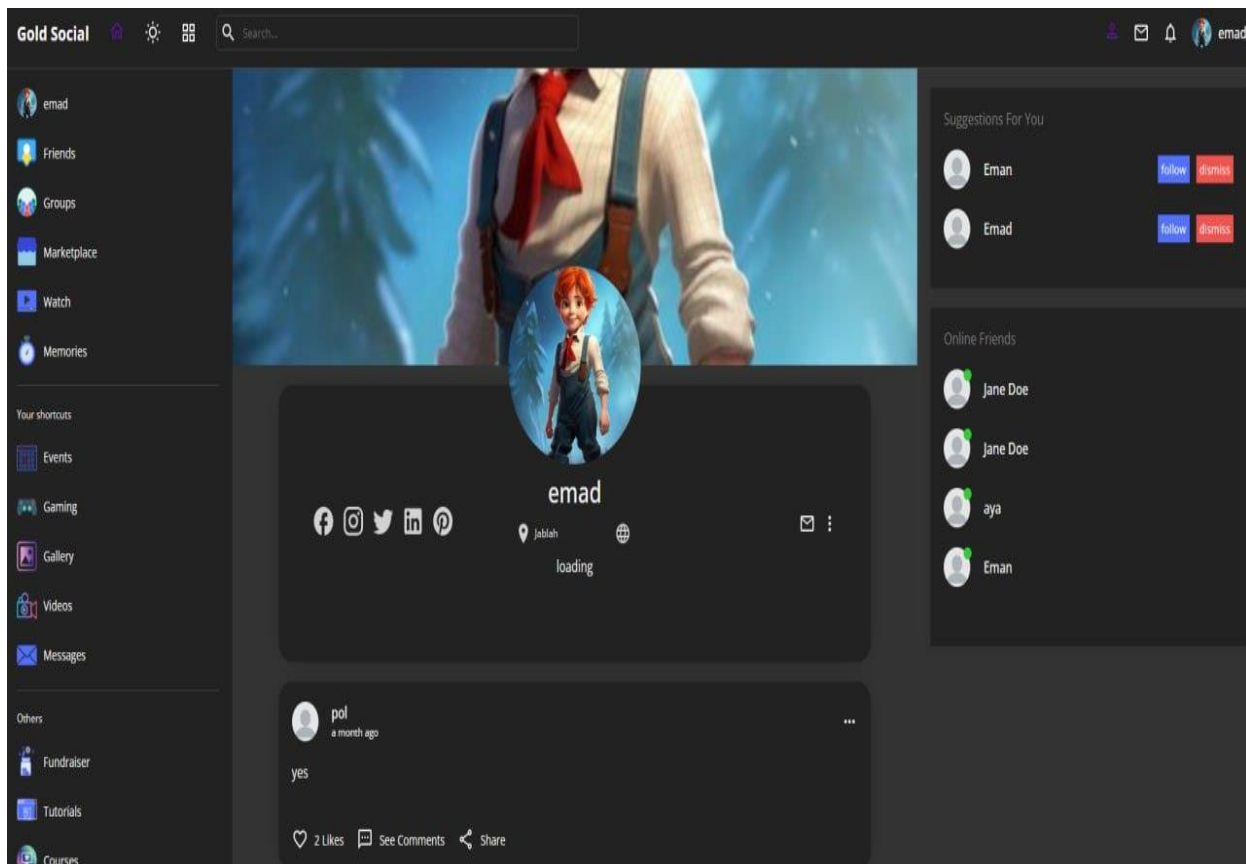
٣-٥-٣: Home page

وهي الصفحة الرئيسية والتي تضم أحدث المنشورات والأقسام المختلفة في الموقع.



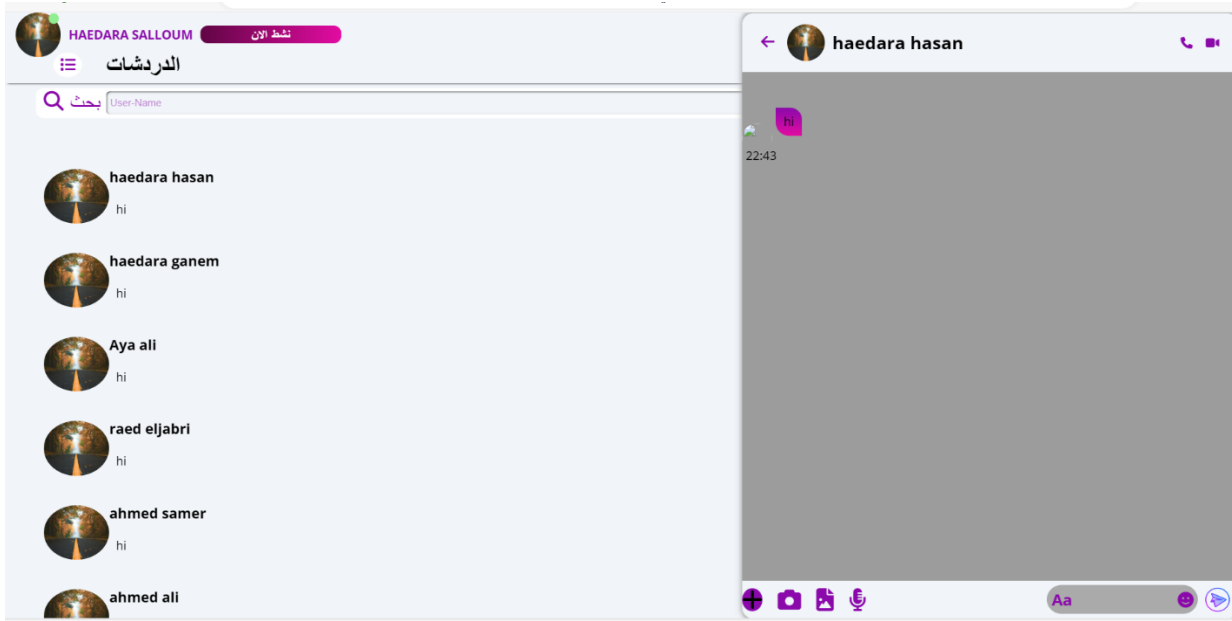
٣-٥-٤: User's personal page

وهي الصفحة الشخصية للمستخدم والتي تعرض المعلومات الخاصة به .



٣-٥-٥: messenger page

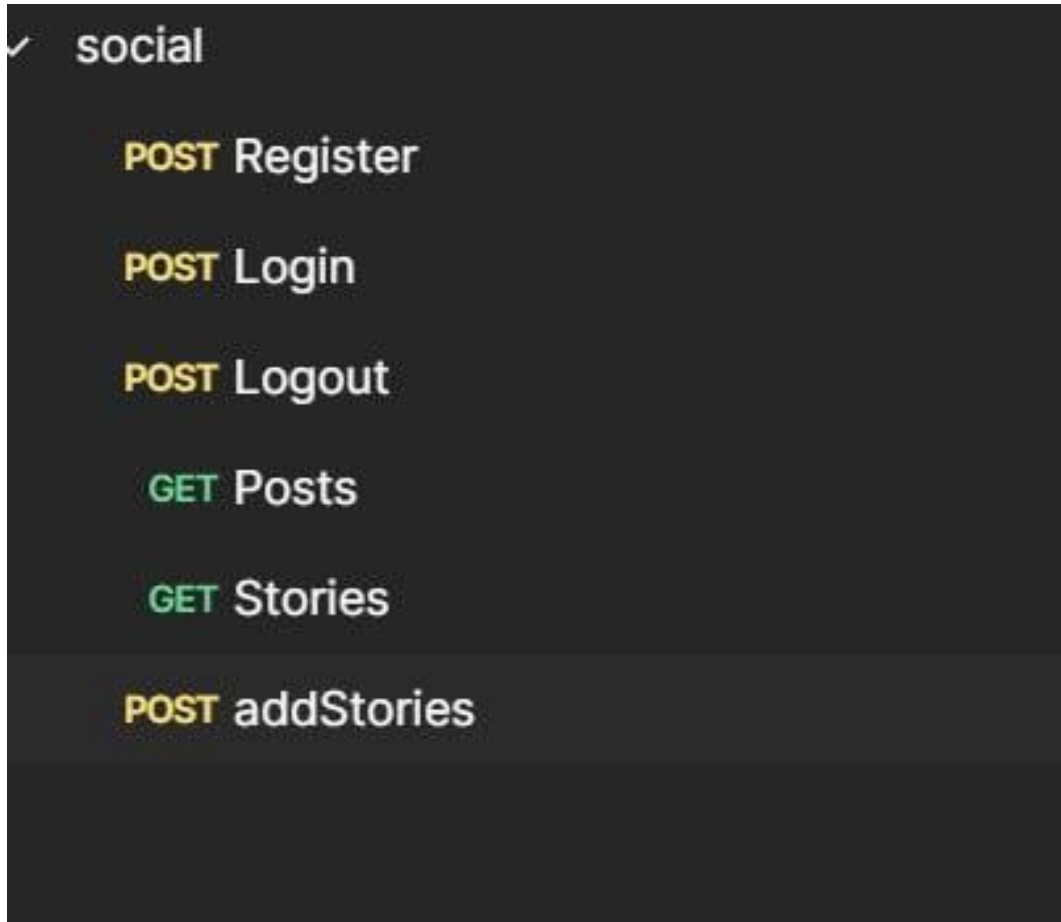
صفحة المسنجر والتي تعرض الأصدقاء النشطين والمحادثات وتمكن المستخدم من التواصل مع الأصدقاء وإرسال الصور والفيديوهات



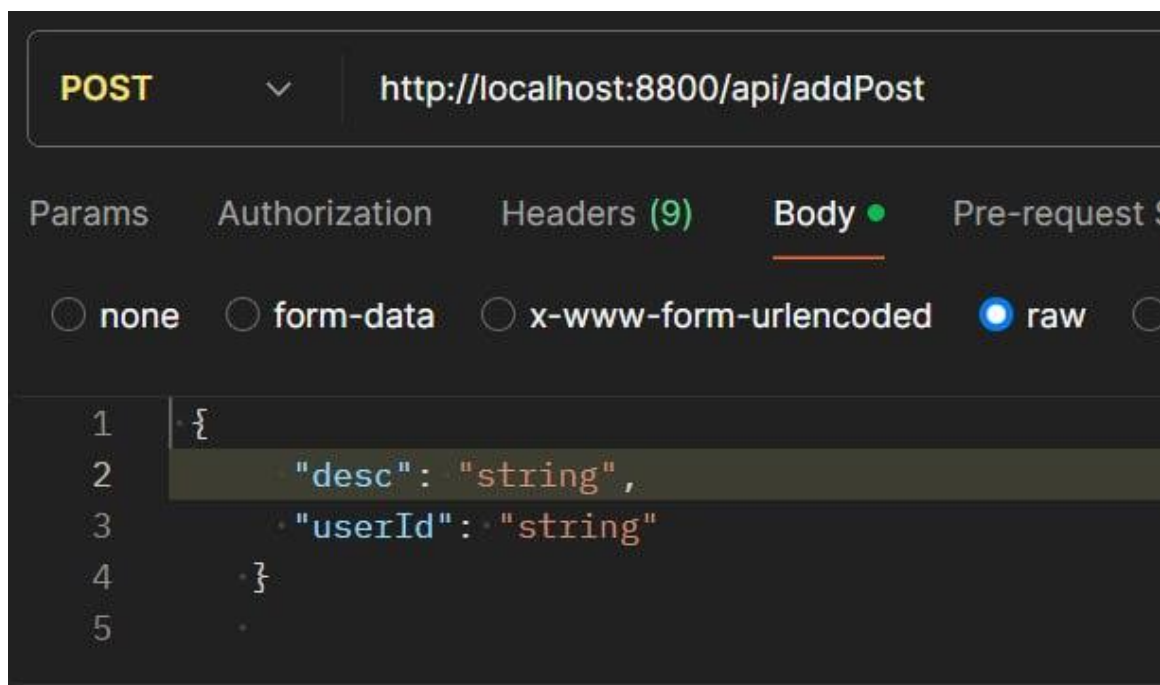
: Backend

في ال backend يتم توفر إمكانية الوصول إلى البيانات والوظائف الأساسية للموقع، مثل إدارة المستخدمين، المشاركات، التعليقات، والمراسلات.

باستخدام postman تم اختبار ال api :



- إدارة المشاركات :
- إنشاء مشاركة جديدة
- الطريقة: POST
- النقطة: api/posts/
- المدخلات:



سنشرح احد الطلبات وهو طلب login :

```
Codiumate: Options | Test this function
34 export const login = (req, res) => {
35   const q = "SELECT * FROM users WHERE username = ?"
36
37   db.query(q, [req.body.username], (err, data) => {
38     if (err) return res.status(500).json(err);
39     if (data.length === 0) return res.status(404).json({});
40
41     const checkPassword = bcrypt.compareSync(
42       req.body.password,
43       data[0].password
44     );
45
46     if (!checkPassword) return res.status(400).json({});
47
48     const token = jwt.sign({ id: data[0].id }, "secret");
49
50     const { password, ...others } = data[0];
51
52     res
53       .cookie("accessToken", token, {
54         httpOnly: true,
55       })
56       .status(200)
57       .json(others);
58   });
59 };
60
```

```
const q = "SELECT * FROM users WHERE username
```

- نقوم بتعريف استعلام SQL لاختيار جميع الأعمدة من جدول users حيث يكون اسم المستخدم مطابقاً للقيمة المقدمة.

```
} <= (db.query(q, [req.body.username], (err, data
```

- نقوم بتنفيذ الاستعلام باستخدام دالة db.query.

- نمرر الاستعلام واسم المستخدم الذي تم الحصول عليه من جسم الطلب (req.body.username).

التحقق من وجود المستخدم ، إذا كانت نتيجة الاستعلام فارغة (data.length === 0)، فهذا يعني أن المستخدم غير موجود. نقوم بإرجاع استجابة مع حالة HTTP 404 (غير موجود) ونرسل رسالة "User not found".

التحقق من كلمة المرور ، نستخدم مكتبة bcrypt للتحقق مما إذا كانت كلمة المرور المدخلة من قبل المستخدم (req.body.password) تطابق كلمة المرور المخزنة في قاعدة البيانات (data[0].password).

- compareSync هي دالة تعمل بشكل متزامن للتحقق من تطابق كلمات المرور.

معالجة كلمة المرور غير الصحيحة ، إذا كانت نتيجة التحقق من كلمة المرور غير صحيحة (checkPassword!)، نقوم بإرجاع استجابة مع حالة HTTP 400 (طلب خاطئ) ونرسل رسالة "Wrong password or username".

إنشاء التوكن

```
const token = jwt.sign({ id: data[0].id }, "secretkey")
```

- نقوم بإنشاء توكن JWT باستخدام مكتبة jsonwebtoken (jwt.sign).

- نقوم بتمرير كائن يحتوي على معرف المستخدم (id) وسر خفي ("secretkey") لتوقيع التوكن.

- إخفاء كلمة المرور

```
const { password, ...others } = data[0];
```

- نقوم بتفكيك الكائن data[0] لاستبعاد كلمة المرور من النتيجة.

- يتم تخزين بقية البيانات في متغير others.

١٠. إرسال الكوكي والتجاوب مع العميل

```
res.cookie("accessToken", token, { httpOnly: true })
```

```
.status(200)
```

```
.json(others);
```

- نقوم بإضافة الكوكي باسم "accessToken" مع قيمة التوكن الذي تم إنشاؤه، مع تعيين خاصية httpOnly لجعل الكوكي غير قابل للوصول عبر JavaScript.

- ثم نقوم بتعيين حالة HTTP إلى 200 (نجاح) ونرسل بقية بيانات المستخدم (بدون كلمة المرور) كاستجابة JSON.

إن التوكن يستخدم لتوثيق المستخدمين بعد تسجيل الدخول حيث يتم حفظ ملف صغير من البيانات يخزن على جهاز المستخدم بواسطة المتصفح (حفظ الجلسة) أي لا يحتاج المستخدم لإعادة تسجيل الدخول في كل مرة

الخاتمة:

في الختام بعد ان تحدثنا عن مزايا الموقع والخدمات التي يقدمها للمستخدمين سوف نتحدث عن أهمية وسائل التواصل الاجتماعي بشكل عام والتي تعتبر ذات صلة بالعديد من المواضيع في حياتنا اليوم فوسائل التواصل لها علاقة بتطوير الأعمال فهي توفر العديد من المزايا للأشخاص اصحاب الأعمال التجارية وتطوير التعليم عن طريق البرامج التعليمية والمواضيع المختلفة وتساعد أيضا على تكوين علاقات جديدة من خلال التعرف على الآخرين بالإضافة الى التسلية والترفيه فالיום اصبحت وسائل التواصل الاجتماعي جزء لا يتجزأ من حياتنا ومن خلال موقعنا حاولنا قدر الإمكان على توفير الخدمات السابقة مع واجهة سلسة وسهلة الاستخدام مع لغات متعددة للموقع ليلبي حاجات المستخدمين جميعهم.

المراجع:

: Youtube channels

.freeCodeCamp: (<https://www.youtube.com/@freecodecamp>) -

Lama Dev (<https://www.youtube.com/@LamaDev>) -

. Dev.to: مجتمع للمطورين (<https://dev.to>) .

: Books

.JavaScript "You Don't Know JS": سلسلة كتب تعمق في

"Learning Node.js Development": كتاب يركز على تطوير التطبيقات باستخدام
.Node.js

Stack Overflow: (<https://stackoverflow.com> /)

- توثيق [Node.js] (<https://nodejs.org/en/docs> /)

- توثيق [React] (<https://reacts.org/docs/getting-started.html>)

- توثيق [Express] (<https://expressjs.com/en/starter/installing.html>)