



الجمهورية العربية السورية

المعهد العالي للعلوم التطبيقية والتكنولوجيا

قسم المعلومات

مشروع في هندسة البرمجيات والذكاء الصناعي  
بعنوان  
تصميم وتطوير نظام لفهم وتحليل المخططات البيانية

إعداد الطالبة زينب رستم  
إشراف د. رياض سنبل

١٠ أغسطس ٢٠٢٥

## ١ الملخص

في عصر البيانات الذي نعيشه، تواجه مختلف الفئات تحديات كبيرة في تحليل الكم الهائل من المخططات والرسوم البيانية، حيث تعتمد الأساليب التقليدية على الخبرة الميدانية والجهد اليدوي المكلف، مما يؤدي غالباً إلى ضياع الرؤى أو سوء الفهم. هذه التحديات تؤثر بشكل مباشر على المحترفين الذين يحتاجون لتحليلات سريعة، والباحثين الذين يتعاملون مع بيانات معقدة، والمعلمين والطلاب في رحلتهم التعليمية، بالإضافة إلى الصحفيين الذين يبنون قصصهم على البيانات، وصناع القرار الذين يعتمدون على دقة هذه التحليلات في تحديد مساراتهم الاستراتيجية.

إن ChartAnalyzer هي منصة متكاملة مدعومة بالذكاء الاصطناعي، تهدف لتطوير تفاعل المستخدمين مع البيانات المرئية وفهمها. من خلال الاستفادة من أحدث نماذج الذكاء الاصطناعي متعددة الوسائط، يحول النظام صور المخططات الثابتة إلى تجارب تحليل بيانات تفاعلية وقابلة للمحادثة، مما يجعل تفسير الرؤى المعقدة للبيانات في متناول المستخدمين من جميع المستويات.

يحدد ChartAnalyzer الإمكانيات التحويلية للذكاء الاصطناعي في تحليل البيانات، حيث يجمع بين قدرات الذكاء الاصطناعي متعدد الوسائط المتطورة وهندسة البرمجيات المدروسة لإنشاء منصة لا تحل التحديات التحليلية الفورية فحسب، بل تفتح أيضاً إمكانيات جديدة لكيفية تفاعلنا مع تصورات البيانات والتعلم منها. مع استمرار تقدم الذكاء الاصطناعي في الفهم المرئي ومعالجة اللغة الطبيعية، ستصبح منصات مثل ChartAnalyzer أدوات أساسية بشكل متزايد لاتخاذ القرارات القائمة على البيانات في جميع قطاعات المجتمع.

## Abstract

In the data-driven era we live in, various groups face significant challenges in analyzing the vast amount of charts and graphs. Traditional methods rely on field expertise and costly manual effort, often leading to missed insights or misunderstandings. These challenges directly impact professionals who need rapid analyses, researchers dealing with complex data, educators and students in their learning journeys, as well as journalists who build stories on data, and decision-makers who rely on the accuracy of these analyses to shape their strategic directions.

**ChartAnalyzer** is a comprehensive, AI-powered platform designed to enhance users' interaction with and understanding of visual data. By leveraging cutting-edge multimodal AI models, the system transforms static chart images into interactive, conversational data analysis experiences, making complex data insights accessible to users of all levels.

**ChartAnalyzer** embodies the transformative potential of AI in data analytics, combining advanced multimodal AI capabilities with thoughtful software engineering to create a platform that not only solves immediate analytical challenges but also unlocks new possibilities for how we interact with and learn from data visualizations. As AI continues to advance in visual understanding and natural language processing, platforms like ChartAnalyzer will increasingly become essential tools for data-driven decision-making across all sectors of society.

## المحتويات

١	الملخص	
٢	أهداف المشروع	
٣	الدراسة مرجعية	
٩	١-٣ مجموعات البيانات Datasets for Chart Understanding	
٩	١-١-٣ Chart-to-Text	
١٠	٢-١-٣ ChartQA	
١٠	٣-١-٣ OpenCQA	
١١	٢-٣ النماذج متعددة الوسائط	
١٢	١-٢-٣ ChartAdapter	
١٣	٢-٢-٣ ChartGemma	
١٥	٣-٢-٣ ChartAssistant	
١٥	٤-٢-٣ ChartLlama	
١٦	٥-٢-٣ TinyChart	
١٦	٦-٢-٣ مقارنة الأداء	
١٧	٣-٣ نماذج استخراج المعطيات	
١٨	١-٣-٣ OneChart	
٤	الدراسة التحليلية	
٢٠	١-٤ متطلبات النظام	
٢٠	١-١-٤ المتطلبات الوظيفية	
٢١	٢-١-٤ المتطلبات غير الوظيفية	
٢٢	٢-٤ مخطط حالات الاستخدام	
٢٤	٣-٤ مخططات تدفق النظام	
٥	تصميم النظام	
٢٦	١-٥ البنية العامة للنظام	
٢٧	٢-٥ البنية المعمارية للطبقة الخلفية	
٢٩	١-٢-٥ أجزاء الطبقة الخلفية	
٣٠	٣-٥ أجزاء الطبقة الأمامية Frontend	
٣١	٤-٥ الخدمات الخارجية ونماذج الذكاء	

## ٦ التنفيذ العملي

٣٣	١-٦ الأدوات والبيئات المستخدمة . . . . .
٣٣	١-١-٦ إطار العمل FastAPI . . . . .
٣٣	٢-١-٦ منصة Google Colab . . . . .
٣٤	٣-١-٦ الحزمة البرمجية Gradio . . . . .
٣٤	٤-١-٦ المكتبة البرمجية React مع TypeScript . . . . .
٣٥	٥-١-٦ نظام PostgreSQL . . . . .
٣٥	٦-١-٦ المكتبة SQLAlchemy . . . . .
٣٥	٧-١-٦ الأداة Alembic . . . . .
٣٦	٢-٦ تنجيز النظام . . . . .
٣٦	٣-٦ التخزين . . . . .
٣٨	٤-٦ الخلفية Backend . . . . .
٣٨	١-٤-٦ نمط المستودع Repository . . . . .
٣٩	٢-٤-٦ حقن الاعتمادية Dependency Injection . . . . .
٣٩	٥-٦ معالجة طلبات التحليل . . . . .

## ٧ تنفيذ واختبار النظام

## ٨ الخاتمة والخطوات المستقبلية

٤٧	١-٨ الخاتمة . . . . .
٤٧	٢-٨ الصعوبات . . . . .
٤٧	٣-٨ الخطوات المستقبلية . . . . .

## ٩ قائمة الأشكال

## قائمة الجداول

١	مقارنة أداء نماذج فهم المخططات البيانية .....	١٦
---	---	----

## ٢ أهداف المشروع

توفير منصة تمكن المستخدمين من:

١. رفع صور المخططات والحصول على تحليل مدعوم بالذكاء الصناعي وإجابات على الأسئلة المتعلقة بتلك المخططات.
  ٢. المحادثة التفاعلية مع نماذج الذكاء الصناعي حول المخططات المرفوعة، مع مراعاة السياق.
  ٣. تتبع وحفظ سجل التحليلات والمحدثات لكل مستخدم.
- تقديم المنصة كتطبيق ويب حديث وسهل الاستخدام لتحليل تصورات البيانات.

### ٣ الدراسة مرجعية

ترافق تطور تقنيات التعلم العميق مع توسع كبير في مجال التعلم متعدد الوسائط، مما أسهم في تحقيق تقدم ملحوظ في توليد الوصف النصي للصور وتوفير إجابات دقيقة للأسئلة المتعلقة بها، ولكن تحورت أغلب الدراسات الحديثة حول الصور الطبيعية أو العامة لذلك كان لابد من المزيد من البحث للاستفادة من هذا النتاج في مجالات معرفية أكثر تخصصاً، مثل المجال الطبي أو مجال الاستشعار عن بعد وغيرها، بينما يبقى فهم وتحليل المخططات البيانية أحد المجالات الأكثر حداثة والأقل من حيث عدد الأبحاث حولها. سنستعرض في هذه الدراسة المرجعية بعض النماذج المبنية لتنفيذ مهام متعلقة بفهم وتحليل المخططات البيانية. أهم المعايير لتصنيف تقنيات فهم صور المخططات البيانية:

#### • المهام المتعلقة بفهم صور المخططات البيانية

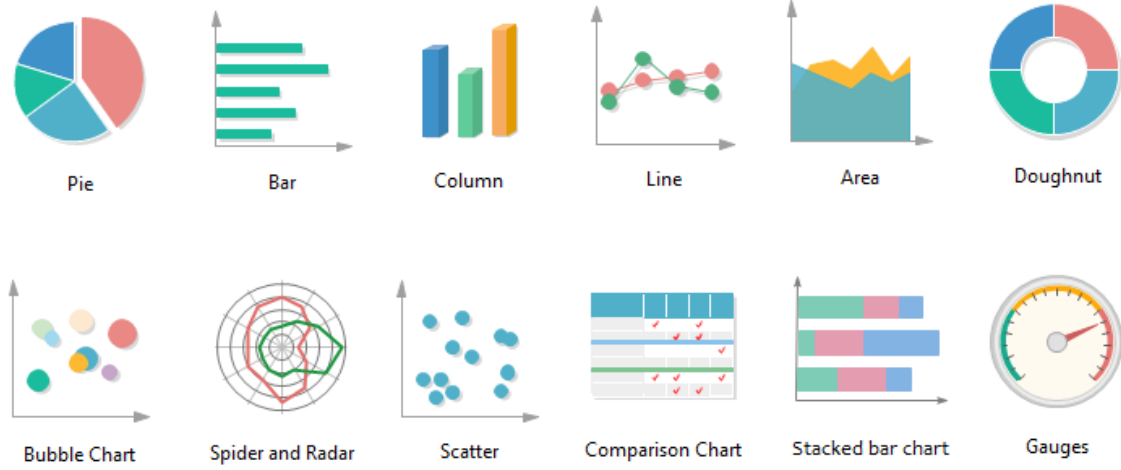
١. التعرف على المخطط
٢. تصنيف المخطط (تحديد نوعه)
٣. استخلاص البيانات من المخطط
٤. توليد توصيف نصي للمخطط
٥. توليد إجابات لأسئلة حول المخطط
٦. توليد تلخيص لمحتوى المخطط
٧. تحويل مخطط لجدول بيانات

#### • أنواع المخططات البيانية

١. مخطط الأعمدة البياني
٢. الخط البياني
٣. مخطط الانتشار
٤. المخطط الدائري
٥. chart candle
٦. chart contour



شكل ١: بعض أنواع المخططات البيانية



### ١-٣ مجموعات البيانات Datasets for Chart Understanding

يتسارع تطور فهم الرسوم البيانية (CU) بفضل مجموعات البيانات متعددة الوسائط التي توفر تمثيلاً شاملاً للمخططات وبياناتها. تعتمد جودة أبحاث CU بشكل أساسي على جودة البيانات، كما تعتمد على تحسين البنى التحتية للنماذج. تتنوع أشكال المخططات وأنواعها (أكثر من 60 نوعاً) وتنسيقات تخزينها (مثل الصور النقطية، SVG، أو تنسيقات الأدوات مثل Vega-Lite)، حيث تسهل التنسيقات المتجهة إعادة بناء الجداول بدقة. تحتوي بعض مجموعات البيانات على نصوص مصاحبة مثل أزواج الأسئلة والأجوبة أو ملخصات المخططات، بالإضافة إلى عناصر محددة بصناديق إحدائيات أو جداول بيانات كمرجع أساسي. سنذكر فيما يلي بعض مجموعات البيانات الرئيسية في هذا المجال، مع التركيز على خصائصها المشتركة وملاءمتها لمهام CU المختلفة. [1]

#### ١-١-٣ Chart-to-Text

في سياق تطوير نماذج الترجمة التلقائية للمخططات البيانية، قدمت دراسة Chart-to-Text [2] مجموعتي بيانات واقعتين: Statista وPew. تم جمع مجموعة Statista من موقع (statista.com) تضم 34,811 مخططاً مصنفاً إلى فئتين: بسيط

(عمودين) ومعقد (ثلاثة أعمدة أو أكثر)، مع وجود جداول بيانات ووصف نصي مرفق. غلب عليها مخططات الأعمدة (88%)، تليها المخططات الخطية (10%). أما مجموعة Pew فقد استُخلصت من موقع (pewresearch.org) وشملت 9,285 مخططاً من 3,999 مقالة، مع ندرة جداول البيانات المرافقة (143 فقط). واجهت عملية تعليق النصوص على المخططات صعوبة بسبب عدم ارتباط الفقرات النصية مباشرة بالمخططات، مما استدعى استخدام تصنيف مكونات المخططات وفرز النصوص عبر عمال AMT. تنوعت أنواع المخططات في Pew بين الأعمدة (67%) والخطية (26%)، مع وجود هياكل معقدة بنسبة 84% من العينات. توفر هاتان المجموعتان مصدراً غنياً لتدريب النماذج على تفسير الرسوم البيانية في سياقات واقعية متباينة التعقيد.

### ChartQA ٢-١-٣

اعتمدت مجموعة بيانات ChartQA [3] على مصادر واقعية متنوعة. لضمان تنوع المحتوى، تم جمع الأسئلة والأجوبة عبر مسارين: الأول يعتمد على كتابة بشرية يدوية، والثاني يستخدم توليداً آلياً عبر نموذج T5 لتحويل الملخصات النصية إلى أزواج أسئلة-أجوبة، مما أنتج مجموعتين فرعيتين: ChartQA-H (البشرية) وChartQA-M (الآلية). تحورت الأسئلة حول العمليات الاستدلالية المعقدة التي تتطلب تحليلات رياضية أو منطقية متقدمة. تحتوي المجموعة على 9,600 سؤال بشري عبر 4,800 مخطط، مما يجعلها مرجعاً غنياً لتدريب النماذج على فهم الرسوم البيانية والإجابة عن الاستفسارات المعقدة.

### OpenCQA ٣-١-٣

تم جمع مجموعة البيانات OpenCQA [2] من موقع Research Pew. تضم 7,7 ألف سؤال من إعداد البشر، مصنفة إلى أربعة أنواع رئيسية: تحديد (استرجاع بيانات مباشر من المخطط)، مقارنة (تحليل عنصرين أو أكثر)، تلخيص (أسئلة تتطلب تحليلاً إحصائياً)، واكتشاف (أسئلة استدلالية تحتاج إلى استخلاص رؤى). تنوع أنواع المخططات في المجموعة بين أعمدة وخطية ودائرية ومساحية ومخططات انتشار توفر هذه المجموعة مصدراً قيماً لتدريب النماذج على فهم الأسئلة المعقدة وتوليد إجابات شاملة تشبه المحادثات الذكية.

شكل ٢: بعض نماذج المخططات في مجموعتي Pew و Statista [1]



## ٣-٢ النماذج متعددة الوسائط

تُعَدُّ نماذج الذكاء متعددة الوسائط (Multimodal Models) أنظمة قادرة على معالجة أكثر من نوع من بيانات الإدخال بشكل متزامن، حيث تُدمج بين أنماط مختلفة كالصور والنصوص المكتوبة باللغة الطبيعية. وقد خضعت العديد من هذه النماذج لعمليات تدريب متنوعة، شملت مهاماً عامة متعددة المجالات، وأخرى متخصصة في نطاقات محددة. وفي السياق ذاته، ظهرت مؤخراً نماذج مُهيكلَة خصيصاً لتحليل الأسئلة المتعلقة بصور المخططات البيانية (Chart Question Answering). وفي هذا الإطار، سنستعرض - وفق ترتيب زمني تنازلي - أبرز هذه النماذج، ونقوم بمقارنة أدائها استناداً إلى المعايير الأساسية المعتمدة في هذا الحقل البحثي.

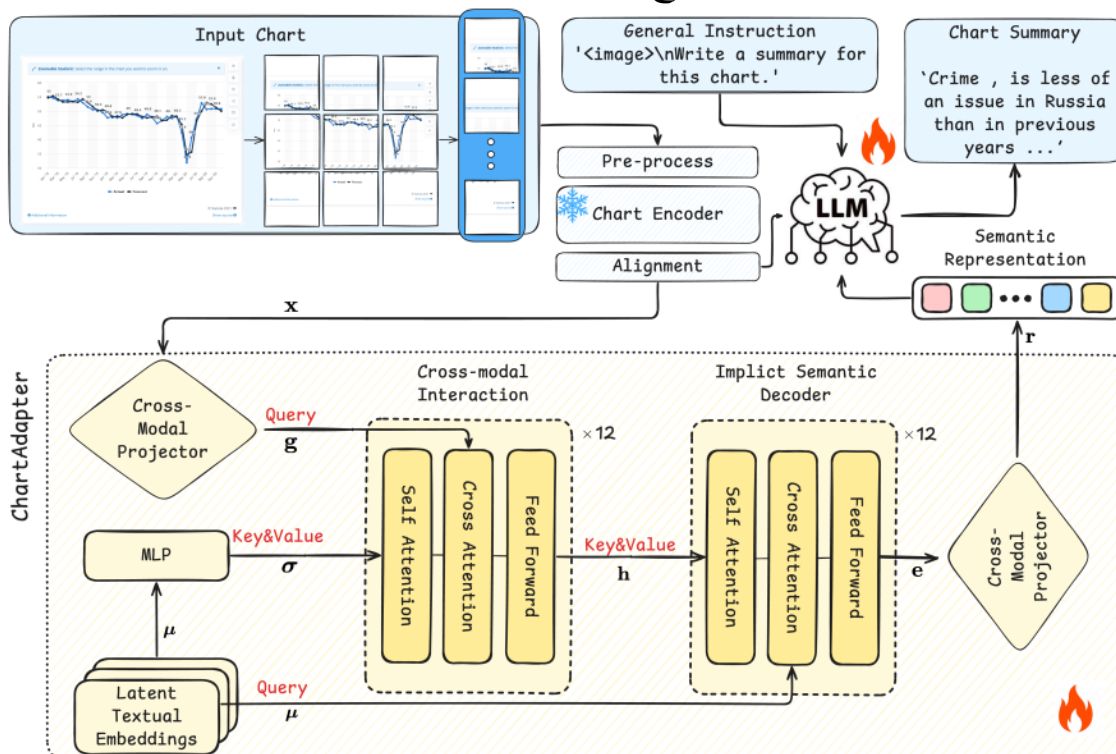
### ChartAdapter ١-٢-٣

اعتمدت بنية هذا النموذج على بنية encoder-decoder [4] المعتمدة في الكثير من أطر العمل السابقة بمجال فهم الصور والمخططات البيانية، كما يوضح الشكل ٣، تمت إضافة جسر بين المرمز البصري visual encoder و نموذج اللغة الكبير textual decoder ، يتكون الدخل من صورة و استعلام نصي، يتم ترميز الصورة بعد إجراء المعالجة المسبقة، وهنا تبدأ المرحلة الخاصة بالمحول ChartAdapter التي تنقسم إلى أربع مكونات جزئية:

١. Cross-modal projector: يحاذي التضمينات البصرية والنصية.
٢. Latent textual embeddings: تستخلص الدلالات الضمنية للمخطط البياني.
٣. Cross-modal interaction layer: تدمج السمات البصرية والنصية عن طريق Attention mechanism.
٤. Implicit semantic decoder: يحول الدلالات البصرية إلى صيغة تتوافق مع نموذج اللغة الكبير LLM.

ثم يتم تمرير كل السمات الناتجة إلى نموذج اللغة الكبير لتوليد ملخصات عالية الجودة. كان الهدف الأساسي من التصميم هو الدمج بين المرمز البصري ونموذج اللغة الكبير من خلال وحدة التحويل ChartAdapter، التي تعزز التعاون بين فهم المخططات وتوليد تلخيصات أو الإجابة على الأسئلة، انعكس هذا التكامل على نتائج النموذج التي تتجاوز بها الكثير من النماذج المهمة السابقة.

شكل ٣: بنية النموذج ChartAdapter [4]



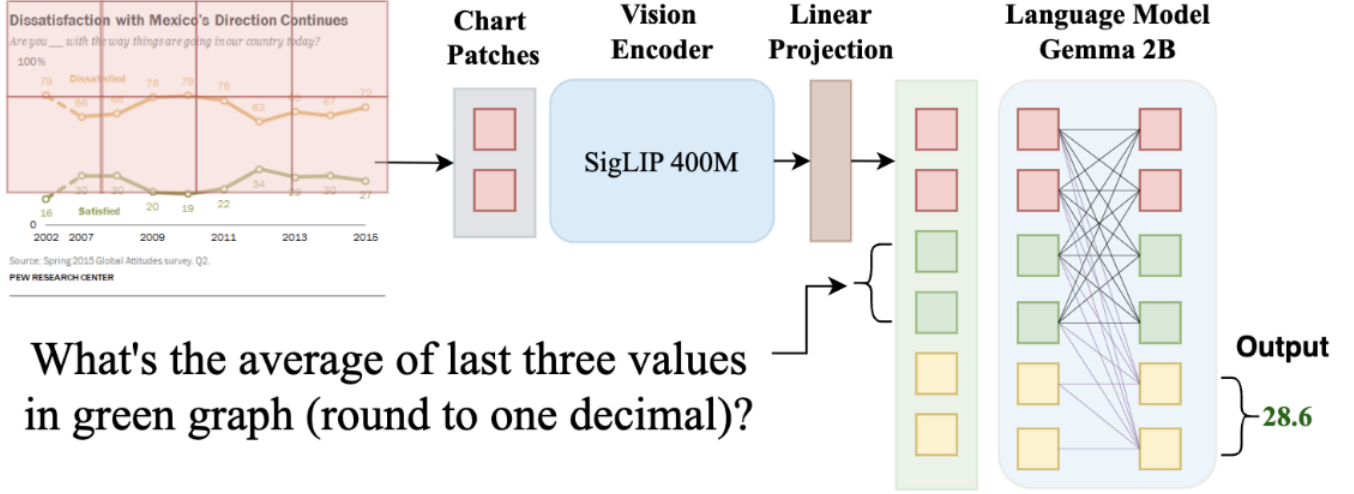
### ChartGemma ٢-٢-٣

هو نموذج خفيف الحجم مقارنة بأمثاله في مجال فهم المخططات البيانية (3 مليارات معامل) [5]، له بنية encoder-decoder بحيث يشكل النموذج السابق PaliGemma [6] بنيته الأساسية حيث تم تدريبه على مجموعة بيانات كبيرة (10 مليارات زوج من الصور والنصوص)، تنقسم بنية ChartGemma إلى مكونين أساسيين:

١. المرمز البصري : يستخدم النموذج vision transformer (ViT) encoder.

٢. مفكك الترميز اللغوي (النصي) : يُستخدم نموذج اللغة Gemma-2B المدرب مسبقاً على بيانات مكونة من وثائق نصية باللغة الانكليزية بالإضافة إلى نصوص برمجية ورياضياتية، مما يجعله مؤهلاً أكثر للتعامل مع المفاهيم المتعلقة بالمخططات البيانية.

شكل ٤: بنية النموذج ChartGemma [5]



في الشكل ٤؛ لدينا نظرة عالية المستوى للبنية الخاصة بالنموذج ChartGemma، تتم معالجة صورة المخطط البياني وتمثيلها على شكل دفعات إلى المرمز الذي يستخلص السمات البصرية من الصورة، بعد ذلك يتم نقل هذه السمات إلى فضاء التضمينات الخاص بنموذج اللغة الكبير، وتُلقى التضمينات النصية للاستعلام المُدخل بالتضمينات البصرية، ثم يقوم نموذج اللغة بتطبيق تقنية الانتباه الكلي على كل رموز tokens الدخول وبالنسبة يتم تطبيق mask على رموز الخرج.

#### مميزاته

أهم ما يميز هذا النموذج أن بيانات التدريب هي تعليمات مولدة بشكل مباشر من صور المخططات البيانية (visual instruction-tuning)، وبالتالي لا يعتمد على جداول البيانات أو أية تقنيات لتوليد هذه البيانات من المخطط، مما يمنح النموذج قابلية تعميم أفضل على أنواع الرسوم البيانية الواقعية، ويسمح له بالتقاط السمات المرئية العامة والتفاصيل الدقيقة بشكل أفضل.

#### أدائه

حقق النموذج ChartGemma نتائج متقدمة في 5 معايير تقييم شملت عدة مهام كتحليل الرسوم البيانية،

الإجابة على الأسئلة، وتقصي الحقائق، وقد تفوق على النماذج الأخرى في إنشاء ملخصات دقيقة وفهم المخططات المعقدة.

### ChartAssistant ٣-٢-٣

هذا نموذج تم تخصيصه لفهم المخططات البيانية [7]. يهدف النموذج للتعامل مع عدة مهام تتعلق بالمخططات البيانية، مثل التحويل من مخطط إلى جدول بيانات أو الإجابة على أشكال متعددة من الأسئلة (nu- merical QA – referring QA – open-ended QA) بالإضافة إلى توليد ملخص نصي عن المخططات، شاملاً بتلك المهام أنواع مختلفة من المخططات والرسوم البيانية. أهم ما يميز هذا النموذج هو التدريب المسبق على تحويل المخططات إلى جداول بيانات نصية، وقد تم تدريبه على مجموعة البيانات ChartSFT الكبيرة والتي تضم أشكال متنوعة من المخططات. تفوق هذا النموذج على نماذج مهمة وحديثة مثل UniChart و ChartLlama لاسيما في اختبارات numerical QA، وذلك على بيانات حقيقية real-world charts بدون fine-tuning فقط اعتماداً على تدريبه المسبق. - البنية: تم تطوير شكلين مختلفين من هذا النموذج:

١. ChartAst-D: نموذج رؤية-لغة صغير نسبياً مبني على أساس النموذج Donut (260M متوسط)، يستخدم المرمز البصري Swin-Base بينما مفكك الترميز النصي BART، يعالج الصورة بدقة 224x224.

٢. ChartAst-S: نموذج رؤية-لغة كبير تم بناؤه على النموذج SPHINX (13B متوسط)، يتم توظيف عدة رموزات بصرية لاستخراج السمات من صورة الدخل، وخلاف ChartAst-D، يتولى نموذج اللغة الكبير عملية تحويل التضمينات النصية بعد دمجها مع البصرية إلى نص لغة طبيعية، وبفضل التصميم المركب للمرمز البصري بالإضافة إلى قوة نموذج اللغة الكبير بالاستنتاج فقد أظهر هذا النموذج نتائج جيدة جداً.

### ChartLlama ٤-٢-٣

بني على النموذج LLaVA-1.5 (مع بعض التعديلات) الذي يشكل بحد ذاته أساساً قوياً لمهمة فهم المخططات البيانية، وهو موجه لعدة مهام مثل فهم المخططات البيانية (Chart QA, Chart Summa- rization/Chart2Text, Data extraction) وتوليدها (Text2Chart, Chart-editing)، وبعض المهام الجديدة التي تشمل (Detailed chart descriptions, multi-chart reasoning, Chart2Chart)، كما يمكنه التعامل مع عدد كبير من أنواع المخططات البيانية المتخصصة، وقد تفوق على عدة نماذج سابقة في المخططات البيانية المتخصصة (funnel, Gantt, heatmap). [8]

### TinyChart ٥-٢-٣

هو نموذج لغة كبير متعدد الوسائط يملك 3 مليارات معامل [9] ، تم تطويره لفهم المخططات البيانية، يتبع طريقة التعلم Program-of-Thoughts بحيث يتدرب على توليد نصوص برمجية بلغة Python لتنفيد العمليات الحسابية وبالتالي لتقليل الأخطاء في الحساب وتخفيف عبء الحسابات المعقدة، كما يطبق Visual Token Merging لسلاسل السمات الطويلة الناتجة عن الصور عالية الدقة مما يحسن من الفعالية الحسابية للنموذج. تتكون بنيته بشكل أساسي من Vision Transformer Encoder, Vision-Language Connector, Large Language Model (Phi-2) تفوق هذا النموذج على نماذج أخرى أكبر حجماً منه (up to 13B parameters) ونماذج أخرى غير مفتوحة المصدر مثل GPT-4V.

### ٦-٢-٣ مقارنة الأداء

فيما يلي جدول مقارنة لأداء النماذج في مهام فهم المخططات البيانية:

جدول ١: مقارنة أداء نماذج فهم المخططات البيانية

Model	Params	ChartQA			Chart-to-Text				OpenCQA	ChartFC
		Aug.	Hum.	Avg.	BLEU-4	ROUGE-1	ROUGE-2	ROUGE-L	BLEU-4	Accuracy
TinyChart@768 [9]	3B	93.86	<b>73.34</b>	<b>83.60</b>	17.18	–	–	–	<b>20.39</b>	–
TinyChart@512 [9]	3B	93.60	72.16	82.88	17.93	–	–	–	19.62	–
ChartAdapter [4]	3B	–	–	–	<b>35.55</b>	<b>41.49</b>	<b>15.75</b>	<b>25.79</b>	–	–
ChartLlama [8]	13B	90.36	48.96	69.66	27.47	–	–	–	–	–
ChartInstruct-Llama [10]	7B	82.40	40.64	61.52	26.1	–	–	–	14.78	64.99
ChartInstruct-Flan-T5-XL [10]	3B	93.84	50.16	72.00	25.01	–	–	–	14.81	<b>72.65</b>
ChartAst-D [7]	260M	91.3	45.3	68.3	27.1	–	–	–	14.9	–
ChartAst-S [7]	13B	<b>93.9</b>	65.9	79.9	28.1	–	–	–	–	–
ChartGemma [5]	3B	90.80	69.52	80.16	–	–	–	–	–	70.33

• الخط العريض يشير إلى أفضل أداء في كل عمود.

• " -- " تشير إلى بيانات غير متوفرة في الجداول الأصلية.

تظهر النتائج في الجدول تفاوتاً واضحاً في أداء النماذج عبر المقاييس المختلفة، مما يبرز ميزة ChartGemma كخيار مناسب لمشروعنا. فعلى الرغم من أن بعض النماذج مثل TinyChart@768 حققت أداءً أعلى في بعض المقاييس، إلا أن ChartGemma يقدم توازناً فريداً يجمع بين:



## ١. أداء متوازن عبر المهام

حقق 80.16 متوسط دقة في ChartQA (بين المُعزّز والبشري)، متفوقاً على نماذج أكبر حجماً مثل ChartLlama 13B فارق كبير نسبياً، مما يدل على كفاءته الحسابية.

## ٢. مميز في التعامل مع المدخلات البشرية

سجل 69.52 في قسم Human من ChartQA، وهو ثاني أعلى نتيجة بعد TinyChart.

## ٣. قوة في الاستدلال المنطقي

حقق ثاني أعلى نتيجة في معيار التأكد من الحقائق، متفوقاً على غيره بنصف عدد المعاملات.

## ٤. ملائمة للسياق التطبيقي

ينما تتفوق ChartAdapter في مهام Chart-to-Text، فإن ChartGemma يقدم أداءً متكاملًا يدعم التحليل متعدد المهام (الإجابة على الأسئلة، الاستدلال، التلخيص)، وكفاءة في استخدام الموارد (3B معامل فقط)، بالإضافة لدعم مُضمن للتعليمات المرئية (visual instruction-tuning)

تم تفضيل ChartGemma لأنه يوفر أفضل توازن بين الأداء والتعقيد الحسابي، مدعوماً بأبحاث حديثة [5] تثبت تفوقه في التحليل الاستدلالي، كذلك فإنّ أدائه المستقر عبر جميع المهام يقلل من مخاطر التحيز لمهمة محددة. وتجدر هنا الإشارة إلى أنه من بين النماذج المتاحة للعموم (ChartLlama - ChartInstruct) فإن ChartGemma هو الأنسب بالنسبة للنتائج والمعطيات المتاحة.

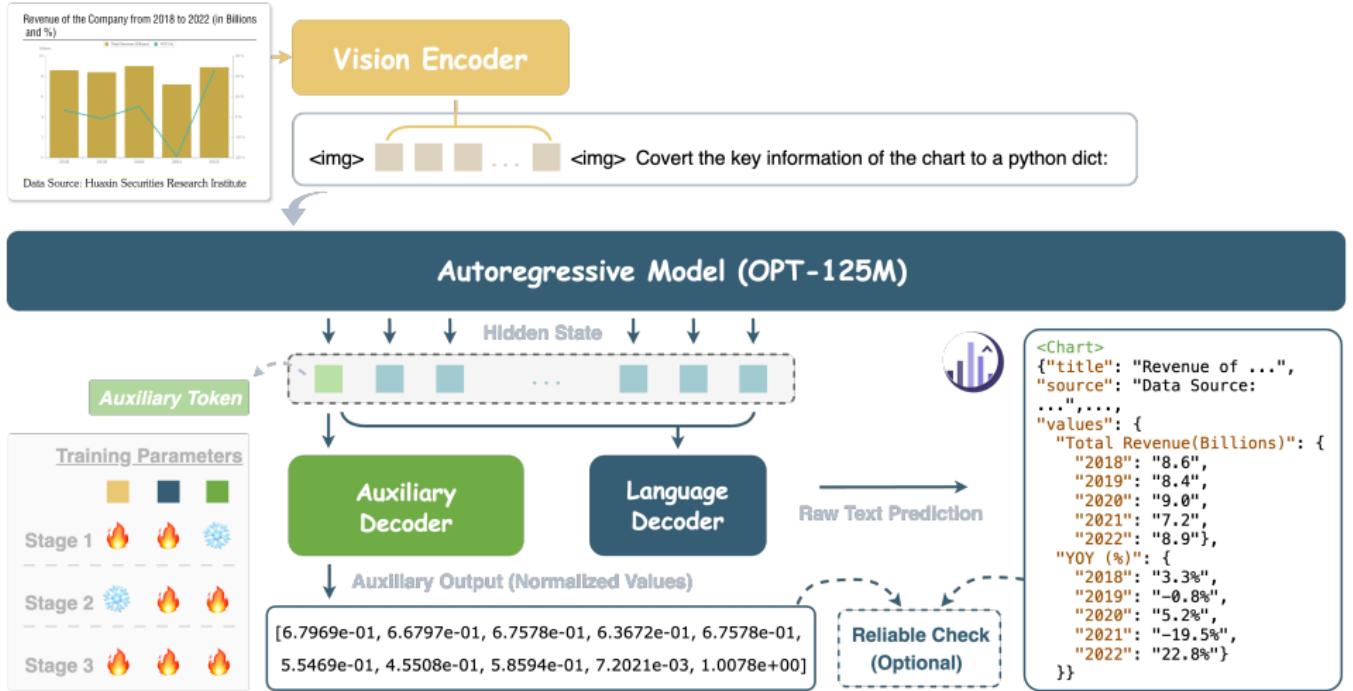
## ٣-٣ نماذج استخراج المعطيات

اتجهت العديد من الدراسات الحديثة إلى تطوير نماذج ذكاء اصطناعي قادرة على استخراج المعلومات بدقة عالية من الصور، مع التركيز على ضمان موثوقية البيانات المُستخلصة وخلوها من الأخطاء. وفي هذا الإطار، برّجت بعض النماذج لتتخصص في معالجة صور المخططات البيانية (Charts)، حيث يمثل الهدف الرئيسي في استخراج القيم العددية التي تمثل نقاط البيانات (Data Points) ضمن هذه المخططات. وفي سياق هذا البحث، سنسلط الضوء على النموذج (OneChart) [11] الذي تميز بأدائه المتفوق في تحليل المخططات البيانية، محققاً نتائج إحصائية لافتة مقارنةً بالنماذج الأخرى في نفس المجال.

### OneChart ١-٣-٣

يُعتبر نموذج OneChart [11] نقلة نوعية في مجال استخراج البيانات المنظمة من الرسوم البيانية، حيث يقدم حلاً مبتكراً للتحديات التقليدية في دقة الاستخلاص الرقمي والهيكلية. يعتمد النموذج على بنية فريدة تدمج بين مكونات رؤية الحاسوب ومعالجة اللغة الطبيعية عبر إدخال علامة مساعدة خاصة (<Chart>) في بداية التسلسل النصي، مصحوبة بجهاز فك ترميز عددي مخصص يعمل بخسارة L1 لتحسين الدقة العددية بشكل منفصل عن المكون اللغوي.

شكل ٥: بنية النموذج OneChart [11]



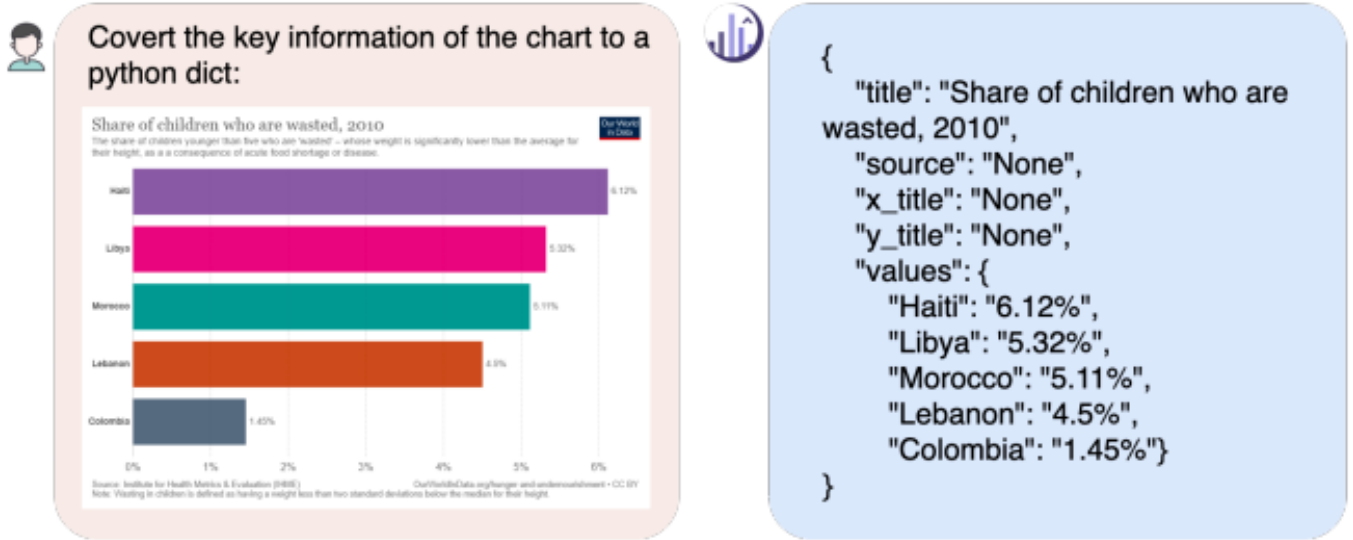
تميز النموذج بقدرته على تقييم موثوقية تنبؤاته ذاتياً عبر آلية قياس الاتساق الداخلي التي تقارن بين مخرجات العلامة المساعدة والقيم المستخلصة، مع إمكانية تصفية النتائج غير الموثوقة بناءً على عتبة إحصائية (أقل من 0.1).  
تم تدريب النموذج على مجموعة بيانات ChartY المتنوعة لغوياً (بالإنجليزية والصينية) وأنماطياً (أعمدة،

خطوط، دوائر)، مما أكسبه مرونة في التعامل مع الرسوم الواقعية المعقدة، حيث تفوق على نماذج أكبر حجماً مثل ChartVLM (B 7.3 معاملات) بنسبة 19.1-29.4 % في دقة الاستخلاص الهيكلي للرسوم غير الموسومة رقمياً.

تكن قيمة الابتكار في تصميمه الخفيف الوزن (B 0.2 معاملات فقط) الذي يحافظ على الأداء العالي مع تمكين النشر على أجهزة ذات موارد محدودة، بالإضافة إلى تكامله السلس مع نماذج اللغة الكبيرة لتعزيز مهام الاستدلال اللاحقة.

في مشروعنا، شكل OneChart حجر الأساس في تحويل الصور البيانية إلى تمثيلات قابلة للمعالجة آلياً، مما مهد الطريق لتحليلات لغوية دقيقة وخالية من الأخطاء الشائعة في النماذج التقليدية.

شكل ٦: مثال لاستخدام النموذج OneChart [11]



بعد التمعن في أهم الإنجازات بمجال فهم المخططات البيانية، واختيارنا للنموذج ChartGemma و النموذج OneChart، سوف نتقل في الفصل التالي إلى الجزء المتعلق بمنصة الويب والتطبيق الخاص بمشروعنا.

## ٤ الدراسة التحليلية

يقدم هذا الفصل دراسة تحليلية للنظام تشمل تحديد المتطلبات الوظيفية وغير الوظيفية، ثم عرض نماذج النظام عبر مخططات حالات الاستخدام ومخططات تنامي النظام، مدعومةً بوصف نصي مُنظم لأهم الحالات، بهدف توضيح آليات التفاعل بين المستخدم والنظام وسير العمليات التفصيلية.

### ١-٤ متطلبات النظام

#### ١-١-٤ المتطلبات الوظيفية

##### ١. المصادقة وإدارة المستخدمين

تسجيل المستخدمين، تسجيل الدخول، تسجيل الخروج، والتحقق من الجلسات (باستخدام JWT) فقط المستخدمون الموثوق بهم يمكنهم إجراء الدردشة.

##### ٢. معالجة صور المخططات

رفع صور المخططات. تخزين واسترجاع صور المخططات المرتبطة بالمستخدمين.

##### ٣. تحليل المخططات

تحليل صور المخططات المرفوعة باستخدام نماذج الذكاء الاصطناعي المتكاملة.

دعم التحليل العام والمتقدم (السياقي، أو باستخدام البرمجة الذهنية - thoughts of Program -).

تمكين المستخدمين من طرح أسئلة محددة حول مخططاتهم.

##### ٤. الدردشة التفاعلية

واجهة محادثة فورية للمستخدمين للتفاعل مع الذكاء الاصطناعي حول مخططاتهم.

الحفاظ على سياق المحادثة لضمان استجابات ذكية وذات صلة.

تتبع وعرض سجل المحادثات والرسائل.

##### ٥. سجل التحليلات والمحادثات

حفظ واسترجاع نتائج التحليل وسجل المحادثات لكل مستخدم.

توفير نقاط نهاية (endpoints) لعرض المحادثات وحذفها.

٠٦. إدارة النظام  
التحقق من حالة الخادم والاتصال بقاعدة البيانات.  
عرض نماذج الذكاء الاصطناعي المتاحة واختبار اتصالها.
٠٧. واجهة المستخدم الأمامية  
واجهة مستجابة وحديثة تدعم جميع الميزات الرئيسية.  
مؤشرات الكتابة، توارينج الرسائل، ودعم الأجهزة المحمولة.
٠٨. واجهة البرمجة (API) والنظام الخلفي  
نظام خلفي باستخدام FastAPI مع مسارات محددة بوضوح لجميع العمليات الرئيسية.  
معالجة الأخطاء وتسجيلها لجميع نقاط النهاية.

#### ٢-١-٤ المتطلبات غير الوظيفية

٠١. الأداء  
إدارة فعالة للحالة وتحسين عرض المكونات.  
مؤشرات التحميل لإعلام المستخدم.
٠٢. القابلية للتوسع  
بنية نظيفة وواضحة (طبقات النطاق، التطبيق، البنية التحتية).  
قابلية التوسع لدعم نماذج وميزات جديدة وزيادة أحمال المستخدمين.  
تحسين قاعدة البيانات وتصميم مخطط فعال.
٠٣. الأمان  
مصادقة آمنة وحماية المسارات.  
التحقق من المدخلات واتصالات API آمنة.
٠٤. سهولة الاستخدام  
واجهة مستخدم حديثة ونظيفة وسهلة الوصول.  
تنقل بديهي وأنماط تفاعل واضحة.

٥. القابلية للصيانة

هيكّل تعليمات برمجية نظيف وفصل واضح للمسؤوليات.  
مكونات قابلة للاختبار والتعديل.

٦. التوافق

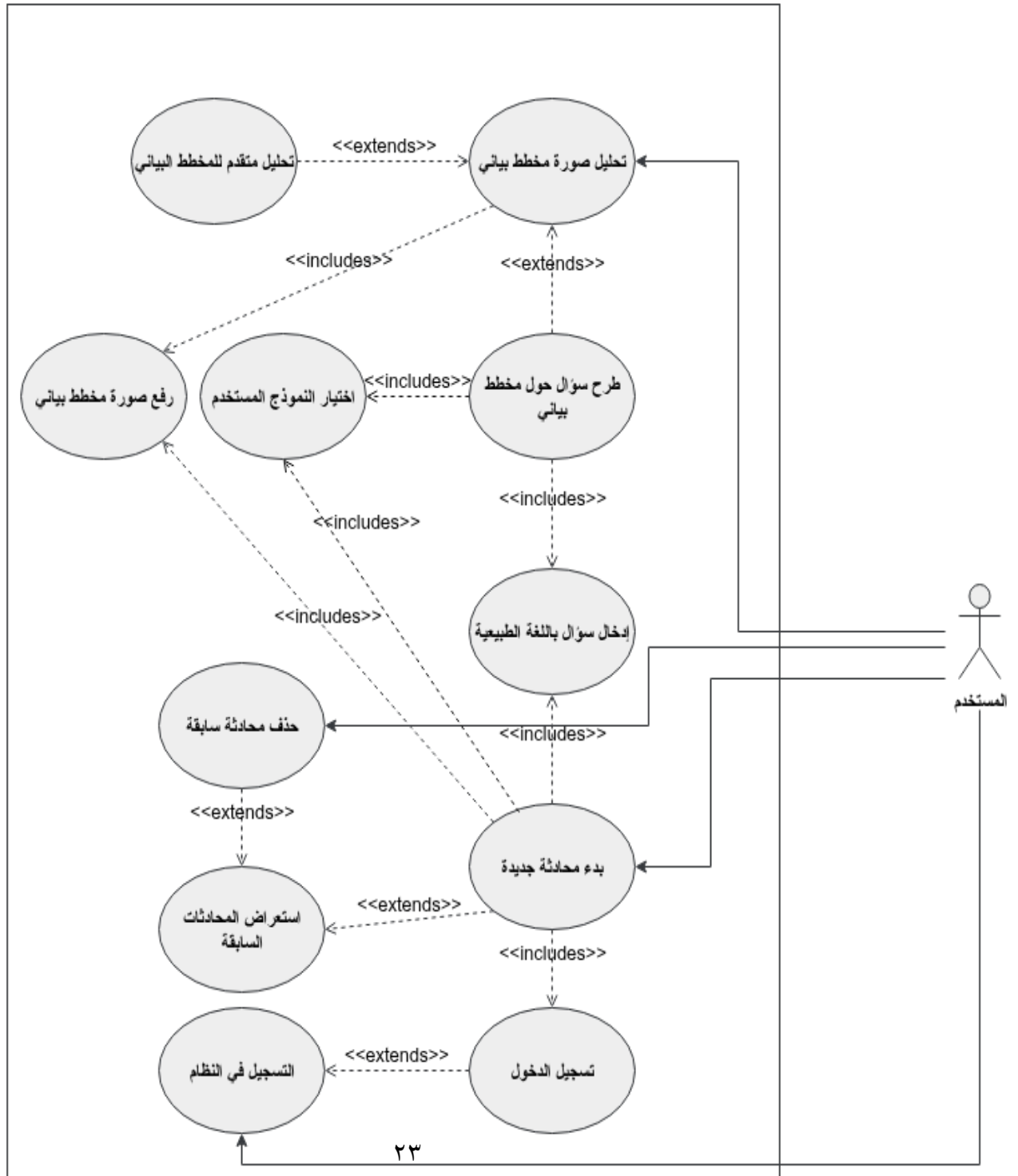
يعمل على جميع المتصفحات الحديثة والأجهزة المحمولة.

## ٢-٤ مخطط حالات الاستخدام

يُبرز المخطط ٧ الوظائف الأساسية التي يتيحها النظام للمستخدمين، بحيث يتمكن المستخدم من اختيار النموذج التحليلي المناسب، ثم رفع صورة المخطط البياني الذي يرغب في تحليله. بعد ذلك، يمكنه كتابة أسئلته بشكل طبيعي لاستخراج المعلومات المطلوبة من المخطط.

كما يقدم النظام خدمات أخرى مثل حفظ سجل المحادثات السابقة، مما يسمح للمستخدم بالعودة إلى التحليلات القديمة أو متابعة النقاشات السابقة بسهولة. ولضمان تجربة مخصصة وآمنة، يتيح النظام إمكانية إنشاء حساب جديد أو تسجيل الدخول للمستخدمين المسجلين.

شكل ٧: حالات الاستخدام

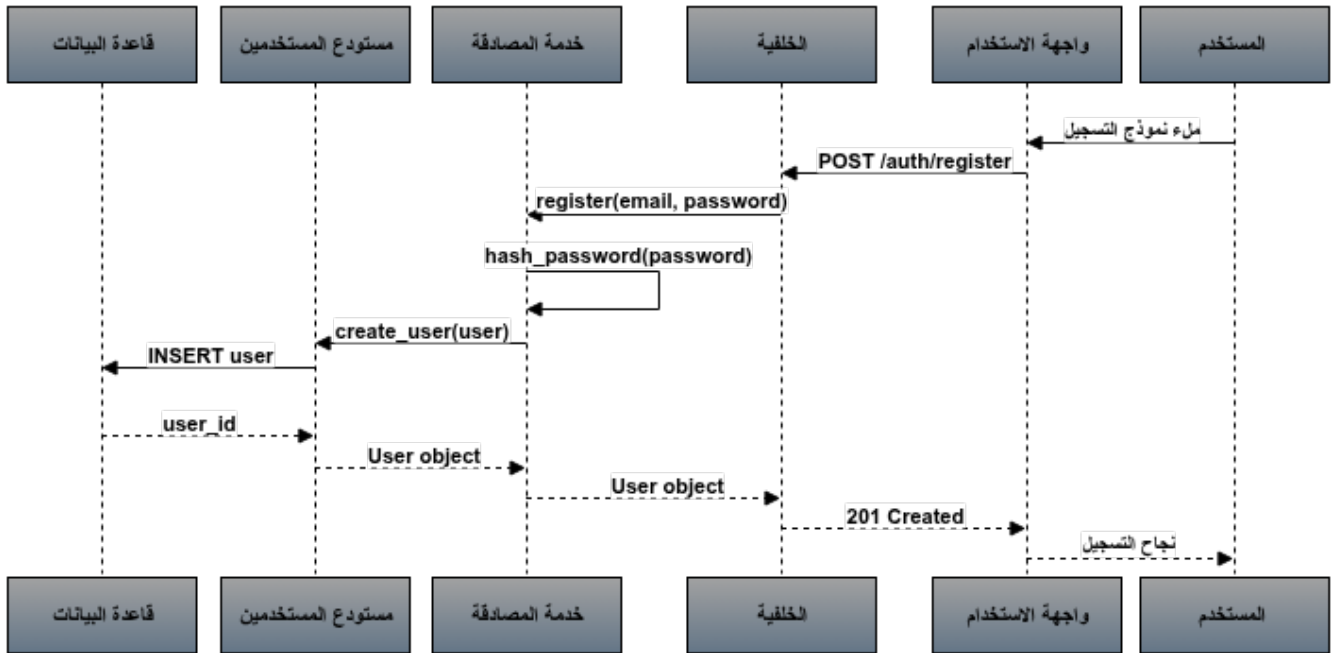


بشكل عام، تم تصميم هذه الوظائف لتقديم تجربة شاملة وسلسلة تبدأ من تحميل المخطط وحتى الحصول على التحليلات المطلوبة، مع الحفاظ على إمكانية الرجوع إلى التاريخ التفاعلي عند الحاجة.

### ٣-٤ مخططات تنامي النظام

١. حالة تسجيل مستخدم جديد يوضح المخطط ٨ عملية تسجيل المستخدم الجديد عبر النظام. يبدأ المستخدم بإدخال بياناته (البريد الإلكتروني وكلمة المرور) عبر واجهة المستخدم، ثم تقوم الخدمة بمعالجة الطلب عبر تشفير كلمة المرور وإنشاء حساب جديد. عند اكتمال العملية بنجاح، يستجيب النظام بإرسال رمز الحالة "201 Created" لتأكيد إنشاء الحساب الجديد.

شكل ٨: تنامي النظام 1

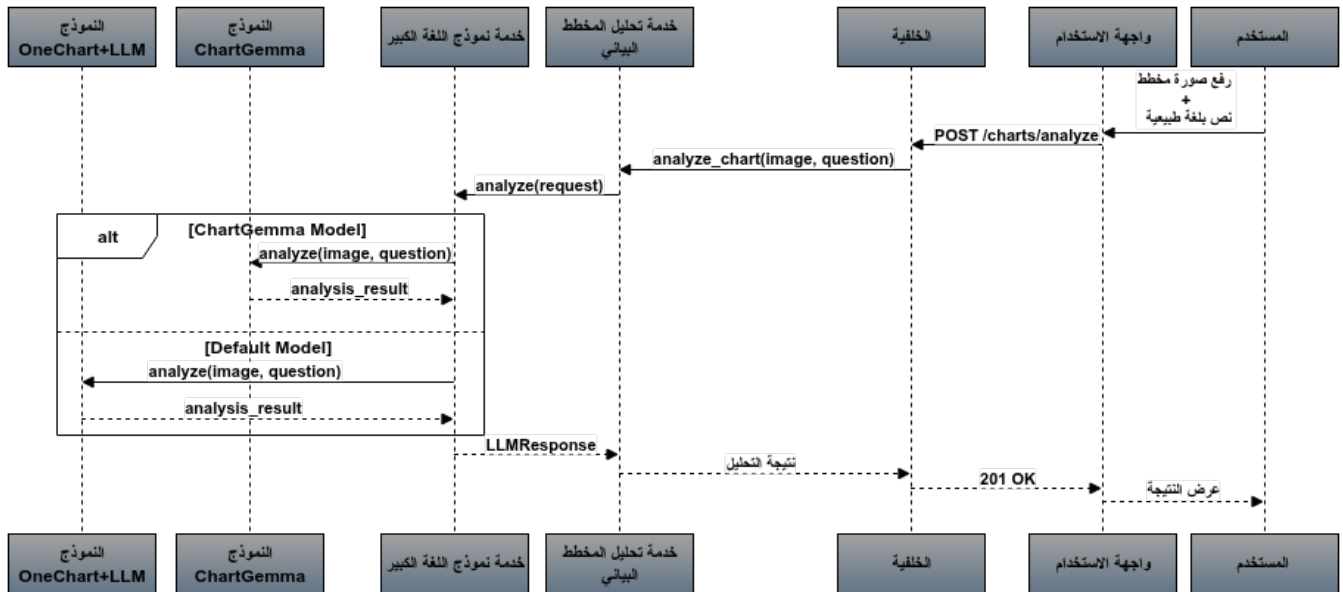


٢. حالة تحليل مخطط بياني يُظهر المخطط ٩ عملية تحليل المخططات البيانية باستخدام نموذجين مختلفين. يقوم المستخدم بتحميل صورة المخطط وإدخال سؤال نصي عبر الواجهة، ثم تصل هذه البيانات إلى خدمة التحليل التي تقوم بمعالجتها باستخدام نموذج ChartGemma أو النموذج الافتراضي.



بعد التحليل، تُعيد الخدمة النتائج بشكل منظم مع رمز حالة "201 OK" لتأكيد نجاح العملية.  
أخيراً، تعرض الواجهة النتائج للمستخدم بطريقة واضحة وسهلة الفهم.

شكل ٩: تسلي النظام ٢



## ٥ تصميم النظام

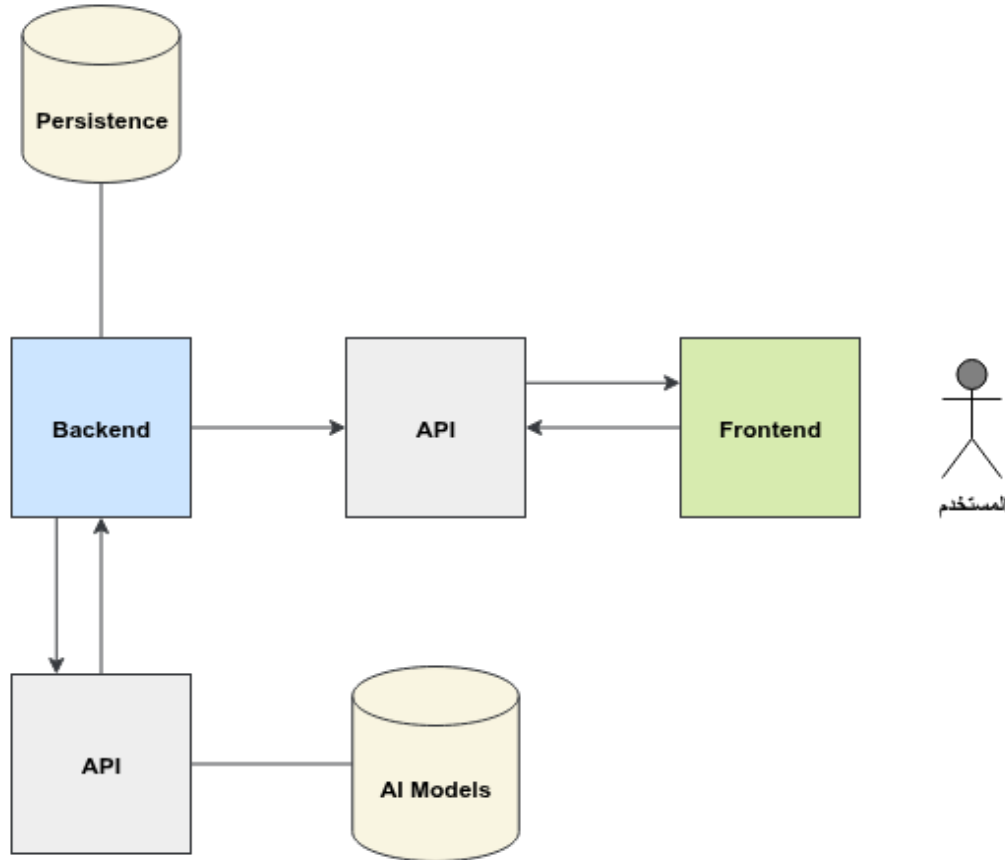
يقدم هذا الفصل التصميم الكامل للنظام المقترح، حيث نستعرض الهيكل العام ومكوناته الأساسية بالتفصيل. سنبدأ بشرح الوحدات الرئيسية ووظائفها، ثم ننتقل إلى كيفية تفاعل هذه المكونات مع بعضها البعض لتحقيق الأهداف المحددة.

### ١-٥ البنية العامة للنظام

يُظهر المخطط ١٠ البنية الأساسية للنظام المقترح الذي يعتمد على تصميم طبقي واضح، حيث تتكون البنية من: واجهة المستخدم الأمامية (Frontend) التي تتفاعل مع المستخدم النهائي، وواجهة برمجة التطبيقات (API) التي تربط الواجهة الأمامية بطبقة الخلفية (Backend) المسؤولة عن معالجة البيانات والمنطق البرمجي.

كما يتضمن النظام وحدات نماذج الذكاء الاصطناعي (AI Models) المتصلة بالواجهة البرمجية لأداء المهام التحليلية المتخصصة، بينما تُعنى طبقة التخزين (Persistence) بإدارة قواعد البيانات وتخزين المعلومات بشكل دائم. يعمل هذا التصميم المتكامل على ضمان تدفق البيانات بسلاسة بين المكونات مع الحفاظ على الفصل الواضح بين المسؤوليات.

شكل ١٠: بنية النظام



## ٢-٥ البنية المعمارية للطبقة الخلفية

تم بناء النظام وفق مبادئ الهندسة المعمارية النظيفة (Clean Architecture) ، لما تعطيه من أولوية لبناء أنظمة مرنة وقابلة للاختبار والصيانة، من خلال فصل المهام وإدارة التبعية بفعالية كبيرة. يتم تحقيق ذلك عن طريق تنظيم الكود البرمجي وفق طبقات متحدة المركز، بحيث يكون منطق العمل المجرد في المركز، وتوضع المهام الأخرى في طبقات خارجية. أهم مبادئ التصميم الجذيرة بالذكر:

### ١. فصل الاهتمامات Separation of Concerns

يتم عزل أجزاء النظام التي تختلف في مجالها/اهتماماتها في طبقات منفصلة، مما يحد من أثر التغيير

في مكون ضمن جزء واحد على مكونات الأجزاء الأخرى.

## ٢. قلب الاعتمادية Dependency Inversion

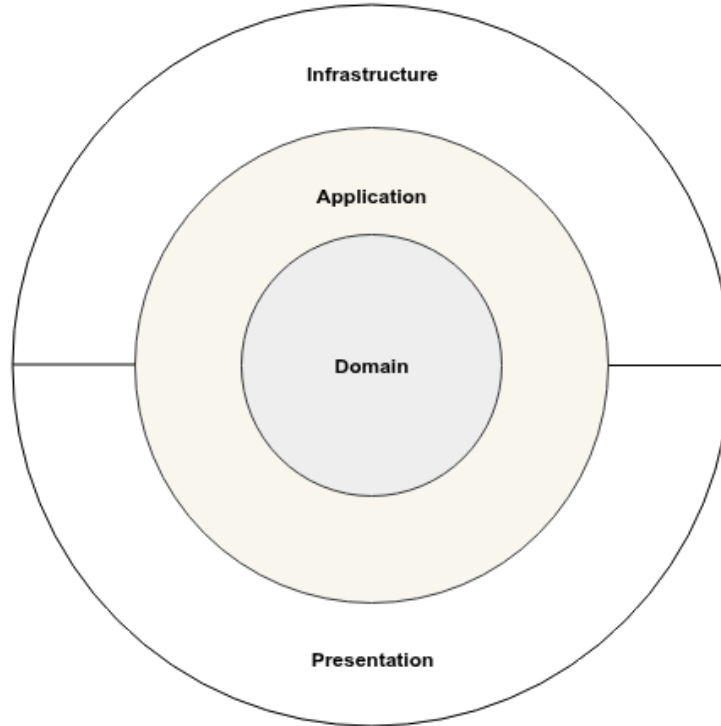
يتم توجيه التبعية/الاعتمادية نحو التجريد الذي يمثل منطق العمل، تعتمد الطبقات الخارجية على الداخلية التي بدورها مستقلة عن كل ما هو خارجها. تضمن هذه المنهجية عدم تأثر منطق العمل الأساسي بالمكونات الخارجية مثل واجهة الاستخدام أو قواعد البيانات.

## ٣. قابلية الاختبار Testability

بعد تحقيق الفصل الواضح بين الاهتمامات، يصبح من السهل اختبار وحدات المكونات على انفراد، وبالتالي اختبار كل وظيفة بدقة أكبر وبمعزل عن الوظائف الأخرى.

يوضح الشكل ١١ بنية المعمارية النظيفة وتموضع الطبقات، بحيث تكون جهة الاعتمادية نحو المركز الموحد.

شكل ١١: بنية المعمارية النظيفة



## ١-٢-٥ أجزاء الطبقة الخلفية

### ١. طبقة النطاق (Domain layer) وتحوي:

- الكيانات وهي صفوف تعرّف الحقول (Fields/Properties) الخاصة بكل كيان في النظام، وتمثل البنية الأساسية للبيانات التي يتم التعامل معها. هذه الكيانات تكون مجردة (Abstract)، أي أنها تُعرّف فقط المنطق الأساسي المتعلق بالبيانات نفسها دون تفاصيل التخزين أو التنفيذ.
- مثل كيان المستخدم، التحليل، المحادثة، المخطط البياني.
- تعريف واجهات المستودعات المختلفة، مثل مستودع صور المخططات ChartsRepository ومستودع التحليلات AnalysisRepository.

### ٢. طبقة التطبيق (Application layer) تتضمن:

- الخدمات هي التي تحوي منطق العمل بشكل مستقل عن التفاصيل الخارجية (مثل طرائق التخاطب مع قاعدة البيانات ونماذج الذكاء)، مثل خدمة المصادقة (AuthenticationService)، وخدمة تحليل المخطط البياني (AnalyzeChartService)، وخدمة المحادثة (ConversationService)، وخدمة LLMService.
- أغراض نقل البيانات (Data Transfer Objects - DTOs) وهي صفوف مخصصة لنقل البيانات بين طبقات النظام أو وحداته المختلفة، حيث تُعرّف حقولاً محدّدة تتوافق مع حالة النقل المطلوبة فقط، دون أي منطق عمل أو معالجة. تُستخدم هذه الكائنات بشكل رئيسي لضمان كفاءة نقل البيانات والفصل الواضح بين الطبقات، حيث تركز فقط على تغليف البيانات ونقلها بأمان بين المكونات (مثل: من الواجهة الأمامية إلى الخلفية أو بين الخدمات).
- مثل: (AnalysisRequestDTO, LoginRequestDTO, LLMResponseDTO...)

### ٣. طبقة البنية التحتية (Infrastructure layer) وتتضمن:

- تمييز المستودعات بما يتناسب مع نظام إدارة قاعدة البيانات المستخدم (DBMS).

- جداول البيانات (Models)، التي استخدمناها لتحقيق التهجير (Migration) إلى جداول وعلاقات حقيقية في قاعدة البيانات.
- إدارة تخزين صور المخططات في نظام الملفات، وحفظ مسارات الصور ضمن قاعدة البيانات بجدول chart\_images.
- تكامل الخدمات الخارجية مثل التخاطب مع نماذج الذكاء المستخدمة.

#### ٤. طبقة واجهات التطبيق البرمجية (APIs)

تحتوي على نقاط نهاية واجهات برمجية التطبيقات (API Endpoints) التي تتعامل مع الطلبات الواردة من الواجهة الأمامية أو الأنظمة الخارجية، وتتضمن موجهات (Routes) منظمة لكل وحدة وظيفية (مثل المصادقة، تحليل الرسوم البيانية، والدردشة)، بالإضافة إلى منطق التحقق (Validation) من صحة البيانات المدخلة، معالجة الأخطاء، وإدارة التبعية (Dependencies). تعمل هذه الطبقة كجسر بين الواجهة الأمامية والخلفية، حيث تستقبل الطلبات، تُحوّلها إلى هياكل بيانات مناسبة (DTOs) ثم توجهها للخدمات المختصة، وأخيراً إعادة الإجابات المنسقة (JSON) مع رموز الحالة المناسبة (HTTP Status Codes).

### ٣-٥ أجزاء الطبقة الأمامية Frontend

تتكون الطبقة الأمامية من واجهة المستخدم التفاعلية المبنية باستخدام React مع TypeScript، والتي تركز على تقديم تجربة مستخدم سلسة وبديهية. تعتمد هذه الطبقة على:

#### ١. المكونات Components

وحدات مستقلة قابلة لإعادة الاستخدام مثل ChartUploader و AnalysisViewer و ChatInterface، كل منها مسؤول عن عرض جزء محدد من الواجهة.

#### ٢. إدارة الحالة State Management

استخدام Context API لإدارة الحالة العامة للتطبيق مثل بيانات المستخدم وسجل المحادثات.

تم تصميم الطبقة الأمامية لتعكس سلاسة التكامل مع الخلفية، مع التركيز على الأداء العالي وسهولة الاستخدام.

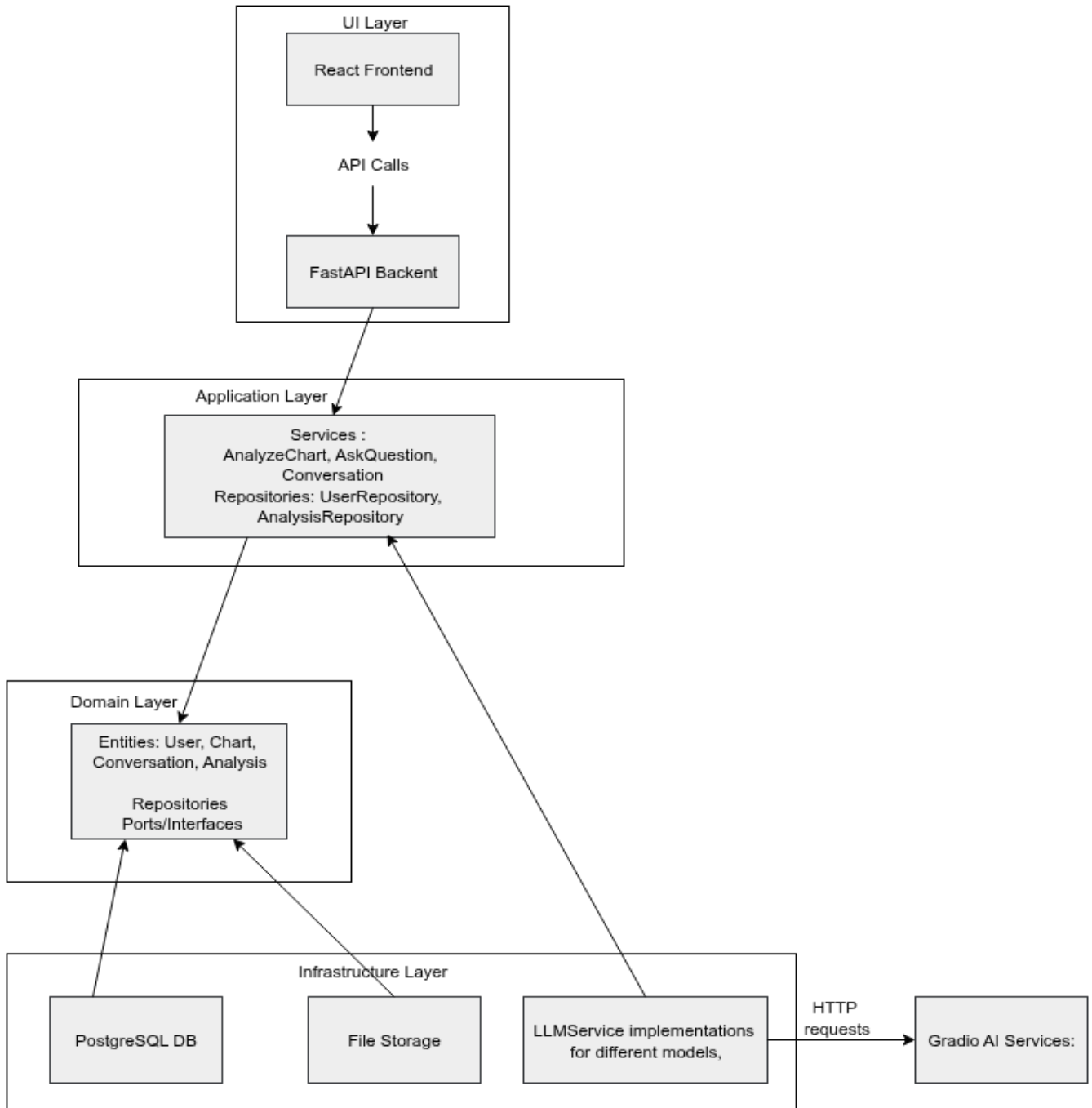
## ٤-٥ الخدمات الخارجية ونماذج الذكاء

تم استخدام النماذج عن طريق التخابط معها وفق واجهات التطبيق البرمجية APIs، وهذا ما جعل التصميم أكثر مرونة بحيث يمكن بسهولة إضافة نماذج أخرى للاستخدام في التحليل، أو تغيير موفر خدمة الاستدلال (Inference).

كان التخابط مع النموذج ChartGemma بسيطاً، عبارة عن طلب من المستخدم، وإجابة (باللغة الطبيعية) من النموذج يتم عرضها مباشرة. أما بالنسبة للنموذج OneChart فتضمنت معالجة الطلب عدة مراحل، أولاً إرسال الصورة كطلب للنموذج، ثم إعادة توجيه الإجابة (Dictionary) مرفقةً بنص سؤال المستخدم إلى نموذج لغة كبير، ليقوم بدوره بالإجابة باللغة الطبيعية.

تم استخدام عدة نماذج لغة كبيرة LLMs لتتكامل مع عمل النموذج، مثل qwen2.5, gemma3, llama, deepseek-r1, mistral-small) يظهر المخطط ١٢ نظرة عامة عن بنية وطبقات النظام، وكيفية تخاطب مكوناته الأساسية فيما بينها.

شكل ١٢: مخطط طبقات ومكونات النظام





## ٦ التنفيذ العملي

يعرض هذا الجزء المراحل العملية التي تم اتباعها لبناء منصة تحليل الرسوم البيانية، بدءاً من تهيئة البيئة البرمجية ووصولاً إلى نشر النظام بالكامل. كما يسلط الضوء على التحديات التقنية التي تم تخطيها والحلول المبتكرة التي تم تبنيها لضمان إنشاء نظام متكامل وفعال.

### ١-٦ الأدوات والبيئات المستخدمة

تم اختيار مجموعة متكاملة من الأدوات والبيئات البرمجية بعناية لتتناسب مع طبيعة المشروع المعقدة ومتطلباته الفنية. يجمع هذا القسم بين منصات التطوير، أطر العمل، المكتبات البرمجية، وأنظمة إدارة قواعد البيانات التي شكلت معاً البنية التحتية التقنية للمشروع. سنعرض كل أداة مع مبررات اختيارها، مزاياها الفريدة، وكيفية مساهمتها في تحقيق أهداف النظام الشاملة، مما يقدم رؤية واضحة عن الأسس التكنولوجية التي بُني عليها المشروع.

#### ١-١-٦ إطار العمل FastAPI

يُعد إطار العمل FastAPI أحد أكثر أطر تطوير واجهات برمجية التطبيقات (APIs) كفاءةً وسرعةً في الوقت الحالي، بفضل بنيته المعتمدة على Python3.7+ دمج مع Pydantic لتحقيق تحقق صارم من أنواع البيانات. يتميز بسرعة أداء قريبة من Go و Node.js بسبب استخدامه لخادم (ASGI Uvicorn) (Asynchronous Server Gateway Interface). يدعم التوثيق التلقائي عبر UI Swagger مما يسهل على الفرق اختبار وتكامل APIs دون جهد إضافي. تكمن قوته في بساطة كتابة الكود مع الحفاظ على القوة، مثل دعمه للمصادقة باستخدام رموز JWT (JSON Web Tokens Authentication). اختيار FastAPI كان أساسياً لهذا المشروع لضمان كفاءة التواصل بين الواجهة الأمامية ونماذج الذكاء الاصطناعي، مع الحفاظ على قابلية الصيانة.

#### ٢-١-٦ منصة Google Colab

كبيئة سحابية مجانية للتعليم الآلي، وفرت (Google Colab) جميع الأدوات اللازمة لتطوير النماذج دون الحاجة إلى موارد محلية قوية. تدعم (Jupyter Notebooks) مع وحدات معالجة رسومات (GPUs, TPUs) مجاناً، وتتكامل مع Google Drive لتخزين البيانات والنماذج.

مميزات مثل التعاون في الوقت الحقيقي والنشر السريع للنماذج جعلتها منصة مثالية لتجربة ChartGemma وOneChart، لكن لا بد من الإشارة إلى أن هذا الخيار ليس الأنسب لإصدار التطبيق بشكله النهائي، لأنها غير مستقرة ولا بد من استخدام وحدات معالجة محلية أو سحابية لكن أكثر استقراراً (خطة مدفوعة)، لتأمين تجربة أكثر موثوقية للمستخدم.

### ٣-١-٦ الحزمة البرمجية Gradio

تمثل مكتبة Gradio جسراً بين نماذج الذكاء الاصطناعي المعقدة والمستخدمين النهائيين، عبر واجهات ويب تفاعلية تُنشأ ببضعة أسطر من الكود. تعتمد على Python وتتكامل بسلاسة مع أطر عمل مثل TensorFlow وPyTorch، مما يجعلها مثالية لنشر نماذج التعلم الآلي كتطبيقات قابلة للاختبار. تقدم مكونات جاهزة مثل (sliders, upload buttons, markdown-supported, textboxes)، مع دعم مدمج لمعالجة الملفات متعددة الوسائط (نصوص، صور، صوت). إحدى أهم ميزاتها هي القدرة على إنشاء روابط عامة يمكن الوصول إليها من أي مكان (public shareable links)، مما يسهل التعاون والتواصل مع أطراف خارجية. في هذا المشروع، ساعدت Gradio في بناء وكشف نقاط نهاية (API endpoints) لإرسال الطلبات إلى نماذج الذكاء المستخدمة وتكاملها مع تطبيق الويب.

### ٤-١-٦ المكتبة البرمجية React مع TypeScript

React مع TypeScript تمثل ثنائياً متكاملًا لتطوير واجهات مستخدم قوية ونوعية، حيث تجمع بين مرونة مكتبة React في بناء واجهات مبنية على المكونات (Components) وقوة نظام الأنواع الثابتة في TypeScript. هذا التكامل يضمن كتابة كود أكثر موثوقية وسهولة في الصيانة، حيث يمكن تعريف أنواع الخصائص (Props) والحالات (State) والأحداث بدقة، مما يقلل من الأخطاء أثناء التطوير ويحسن جودة الكود. تتيح React مع TypeScript تجربة تطوير محسنة مع مميزات مثل الإكمال التلقائي الذكي والتحقق من الأنواع أثناء الكتابة. كل هذه الخصائص السابقة، إضافةً لسهولة تكاملها مع أيًا كانت التقنيات الأخرى المستخدمة معها، يجعلها خياراً ممتازاً لبناء واجهات تفاعلية لتطبيقنا.

## ٥-١-٦ نظام PostgreSQL

هو نظام إدارة قواعد بيانات علائقي مفتوح المصدر يتميز بقدرته الفائقة على التعامل مع أحجام كبيرة ومتنوعة من البيانات، بما في ذلك البيانات المعقدة غير المهيكلة مثل JSON وXML. بالإضافة إلى ذلك، يوفر PostgreSQL بنية علائقية قوية تمكننا من تحديد العلاقات بين الكيانات المختلفة بدقة، مما يضمن سلامة البيانات واتساقها مع تلبية احتياجات التطبيق بكفاءة عالية سواء في عمليات التخزين أو الاسترجاع. تمكن قوة PostgreSQL في دعمه الممتاز للمعاملات المعقدة والاستعلامات المتقدمة، مما يجعله نظاماً مثالياً للتطبيقات التي تتطلب أداءً عالياً وقابلية للتوسع.

## ٦-١-٦ المكتبة SQLAlchemy

تمثلت SQLAlchemy حجر الأساس في التعامل مع قاعدة البيانات في هذا المشروع، حيث وفّرت طبقة تجريدية قوية تربط بين كائنات Python والبنية العلائقية لقاعدة البيانات. تعمل هذه المكتبة كنظام (ORM) (Object-Relational Mapping) متكامل، يسمح بتمثيل الجداول كصفوف في بايثون والعلاقات بينها بخصائص للكائنات، مما يجعل التعامل مع البيانات أكثر سلاسة وقراءة. تميزت SQLAlchemy بقدرتها على دعم مختلف أنواع الاستعلامات المعقدة عبر نظام API Query الذي يجمع بين مرونة لغة SQL وسهولة البرمجة كائنية التوجه. كما ساهمت ميزة Session Management في إدارة فعالة لسياق العمل مع قاعدة البيانات، مع توفير آلية للتحكم في المعاملات (Transactions) تضمن سلامة البيانات عند حدوث أخطاء. تكاملت المكتبة بشكل مثالي مع PostgreSQL لتقديم أداء عالٍ، بينما حافظت على استقلاليتها عن نظام إدارة قواعد البيانات المستخدم، مما يسمح بتبديلها مستقبلاً دون الحاجة لإعادة كتابة شفرات البرمجة. في هذا المشروع، سهّلت SQLAlchemy عمليات الإدخال والاستعلام المعقدة، كما دعمت نظام التهجير عبر Alembic لضمان تزامن نموذج البيانات مع قاعدة البيانات الفعلية في جميع مراحل التطوير.

## ٧-١-٦ الأداة Alembic

لعبت أداة Alembic دوراً محورياً في إدارة التغييرات الهيكلية لقاعدة البيانات بطريقة منهجية وآمنة، حيث تم استخدامها كأداة رئيسية لتنفيذ نظام التهجير (Migration) في المشروع. تعمل Alembic كطبقة إدارة فوق SQLAlchemy مما يتيح تتبع التعديلات على نموذج البيانات وتطبيقها تدريجياً على قاعدة البيانات الفعلية مع الحفاظ على سلامة البيانات الموجودة. تبدأ العملية بإنشاء تعليمة تهجير أولية تحتوي على الهيكل الأساسي، ثم يتم توليد تعليمات لاحقة لكل تعديل يتم إدخاله على

النموذج، سواء أكان إضافة جداول جديدة، تعديل حقول موجودة، أو حتى حذف علاقات. تتميز Alembic بقدرتها على التعامل مع البيئات النموية المختلفة، حيث تسمح بالعكس عن التهجيرات (downgrade) عند الحاجة، كما توفر نظاماً متكاملًا لإدارة الإصدارات يمكن من خلاله تحديد حالة قاعدة البيانات بدقة في أي مرحلة من مراحل التطوير. في هذا المشروع، ساهمت Alembic في تسهيل عملية التكامل المستمر (CI/CD) من خلال إمكانية تطبيق التحديثات الهيكلية على قواعد البيانات بطريقة منتظمة وخالية من الأخطاء، مع الاحتفاظ بسجل كامل لجميع التغييرات التي تم إجراؤها منذ بداية المشروع.

## ٢-٦ تنجيز النظام

فيما يلي سنوضح كيفية تنجيز التصميم المقترح باستخدام الأدوات المذكورة سابقاً.

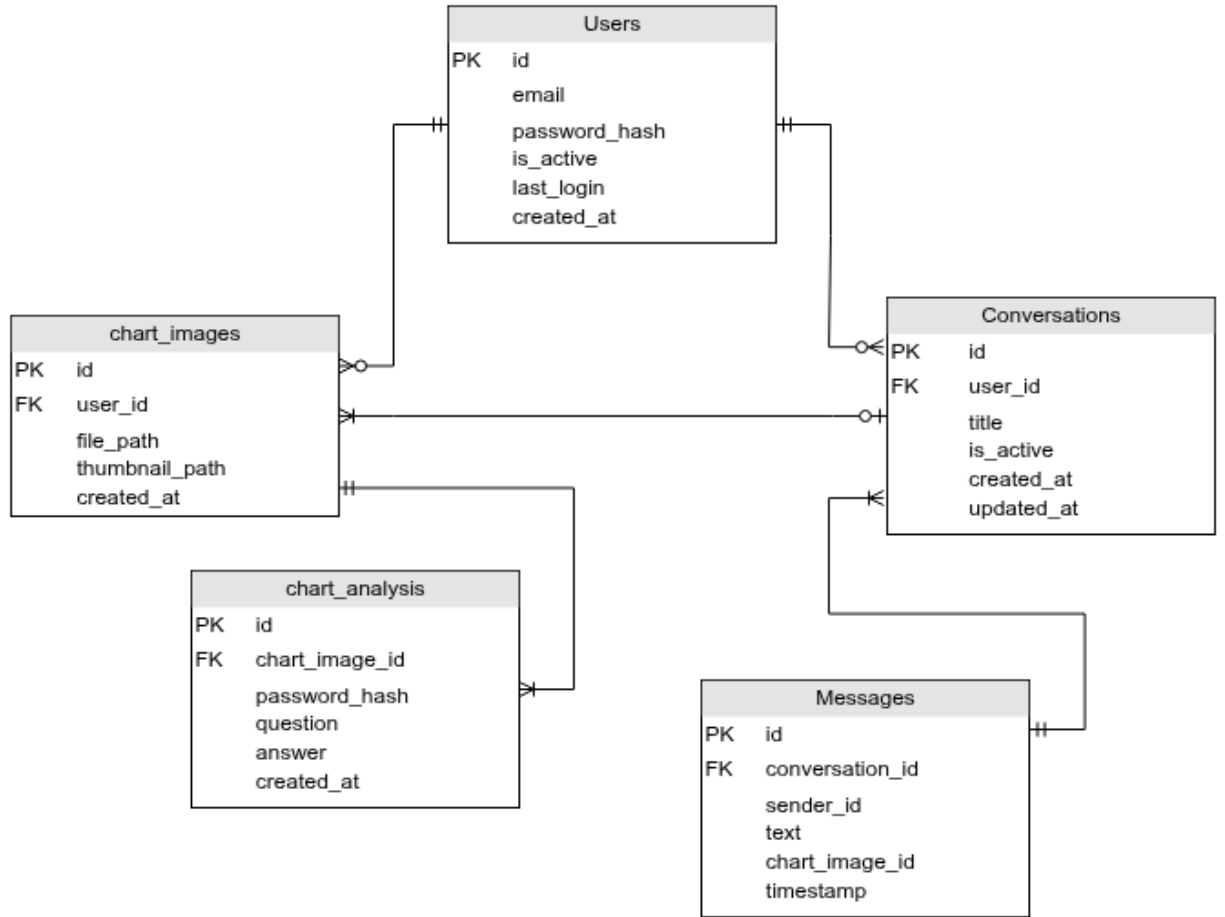
## ٣-٦ التخزين

تم تخزين معلومات المستخدمين، المحادثات والتحليلات في قاعدة بيانات واحد باستخدام PostgreSQL



يُظهر المخطط العلائقي ١٣ الهيكل الأساسي لقاعدة بيانات التطبيق، حيث يركز على أربع كيانات رئيسية: المستخدمون والرسوم البيانية والتحليلات والمحادثات. تبدأ بجدول "المستخدم" الذي يُخزن المعلومات الأساسية للمستخدمين مثل البريد الإلكتروني وكلمة المرور المشفرة. ثم يأتي جدول "صور الرسوم البيانية" الذي يرتبط بالمستخدمين ويحفظ مسارات الملفات المصورة. أما جدول "تحليل المخططات" فيربط كل تحليل بالرسم المقابل ويخزن الأسئلة والأجوبة التحليلية. أخيراً، توجد جداول "المحادثات" و"الرسائل" التي تتيح للمستخدمين إجراء محادثات تفاعلية مع ربط الرسائل بالرسوم البيانية عند الحاجة، مع حفظ تاريخ كل محادثة وتحديثاتها. يعكس هذا التصميم العلاقات المنطقية بين مكونات النظام مع الحفاظ على سلامة البيانات وسهولة الاستعلام.

شكل ١٣: تصميم قاعدة البيانات



تم تعريف النماذج اللازمة لتحقيق قاعدة البيانات السابقة ومن ثم تطبيق التهجير للحصول على قاعدة بيانات منظمة بالتصميم الذي نريده، كانت العملية سلسلة بالإضافة لسهولة تحديث بنية قاعدة البيانات بسهولة وسرعة عند الحاجة.

تم استخدام التخزين في نظام الملفات (Filesystem Storage) لتخزين صور المخططات البيانية، والاحتفاظ بمسار الصورة ضمن جدول صور المخططات في قاعدة البيانات.

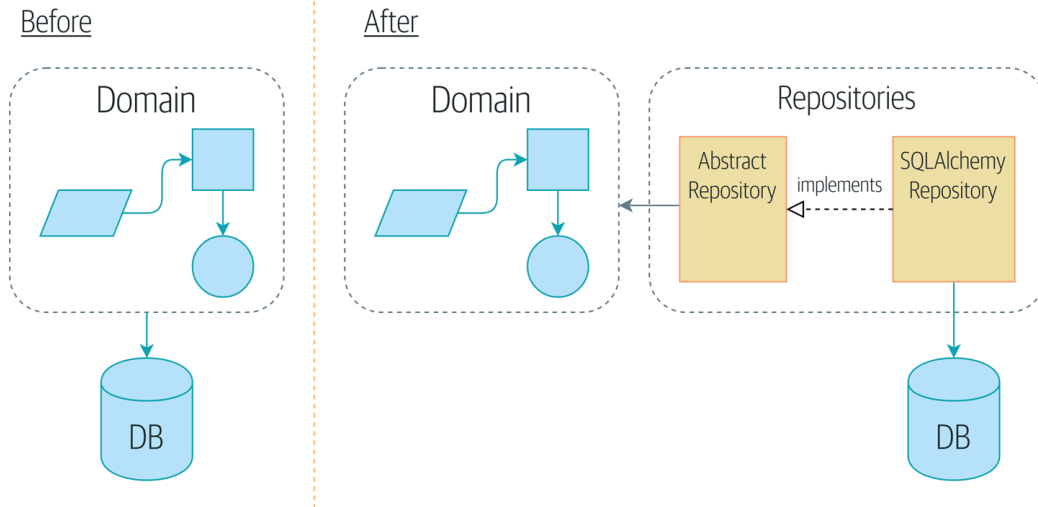
## ٤-٦ الخلفية Backend

كما ذكرنا سابقاً، تم تمييز الطبقة الخلفية وفق مبادئ الهندسة المعمارية النظيفة، لتحقيق قابلية التوسع بالحد الأكبر الممكن. بعد تحقيق بنية الخلفية، تم كشف نقاط النهاية لتكون جاهزة للتعامل مع الطبقة الأمامية. وباستخدام FastAPI كانت نقاط النهاية مقسمة في ثلاث موجّهات أساسية ضمن ثلاث ملفات (auth.py - charts.py - conversations.py) بحيث يمكن للمستخدم قبل التسجيل في النظام، أن يطلب تحليل مخطط بياني بعد رفع صورته، أو إدخال سؤال محدد، ولكن خدمة الحادثة التفاعلية كانت محصورة بالمستخدمين المسجلين في النظام، وذلك لضمان حفظ واسترجاع الحوادث السابقة وتفاصيل التحليلات لكل مستخدم. فيما يلي أهم النقاط في تمييز خلفية التطبيق:

### ١-٤-٦ نمط المستودع Repository

نمط المستودع (Repository) هو وسيط بين طبقة التطبيق وقاعدة البيانات، حيث يوفر واجهة موحدة للوصول إلى البيانات دون الحاجة للتعامل مع تفاصيل التخزين مباشرة. يعمل كمخزن مركزي لإجراء عمليات CRUD (إنشاء، قراءة، تحديث، حذف) على الكائنات، مع إخفاء تعقيدات استعلامات قاعدة البيانات. يسهل هذا النمط اختبار الكود واستبدال مصدر البيانات لاحقاً، كما ينظم عمليات الوصول إلى البيانات بطريقة نظيفة وقابلة للصيانة.

شكل ١٤: نمط المستودع في البنية النظيفة



لعب نمط المستودع دوراً محورياً في تحقيق فصل واضح بين طبقة المنطق التجاري وطبقة الوصول إلى البيانات في مشروعنا، حيث عمل كوسيط يعزل تفاصيل التعامل مع قاعدة البيانات عن باقي مكونات النظام. قننا بتنفيذ هذا النمط من خلال إنشاء واجهات (Interfaces) محددة لكل كيان رئيسي (مثل المستخدمين والرسوم البيانية والتحليلات)، مع توفير تطبيقات ملهوسة لهذه الواجهات تعمل مع قاعدة البيانات عبر SQLAlchemy. وقر مرونة كبيرة في تغيير مصدر البيانات دون الحاجة لتعديل طبقة التطبيق أو النطاق. على سبيل المثال، عند التعامل مع كيان "التحليل"، كان مستودع التحليل (AnalysisRepository) مسؤولاً عن جميع عمليات CRUD، خافياً تفاصيل تنفيذ الاستعلامات المعقدة وراء واجهة بسيطة مثل `save_analysis()` و `get_analysis_by_id()`. هذا النهج لم يحسن قابلية الصيانة فحسب، بل عزز أيضاً مبدأ المسؤولية الواحدة (Single Responsibility Principle) وسهل عملية التوسع المستقبلي للنظام.

## ٢-٤-٦ حقن الاعتمادية Dependency Injection

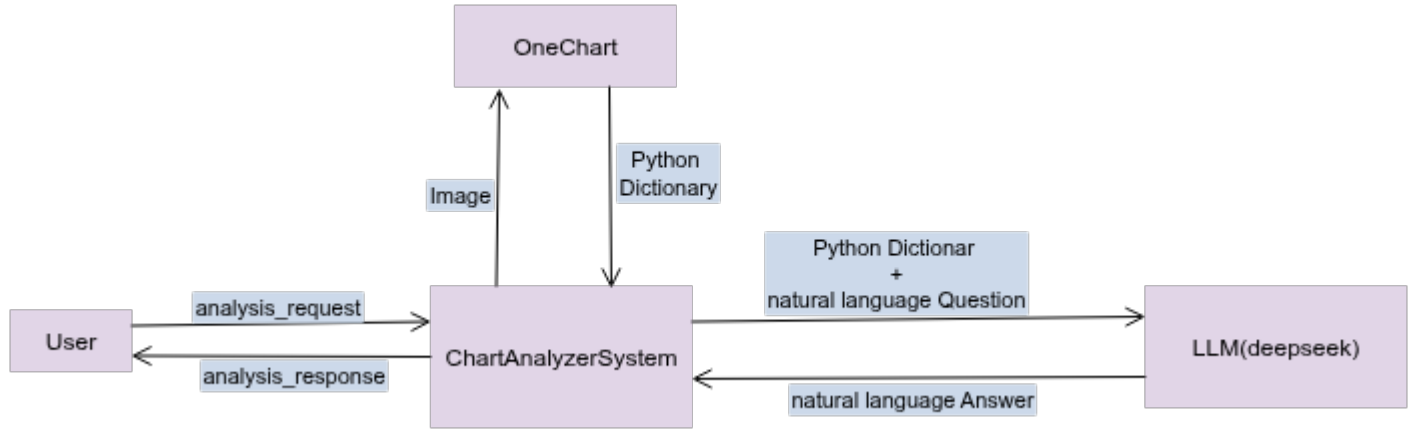
يُطبق مشروعنا مبدأ حقن الاعتمادية (Dependency Injection) بسلاسة عبر فصل تعريف التبعيات عن استخدامها، مما يعزز المرونة والقابلية للاختبار. حققنا ذلك في ملف `dependencies.py` حيث نُحقن جميع المكونات (من مستودعات البيانات مثل `SqlUserRepository` إلى خدمات الذكاء الاصطناعي مثل `ChartGemmaService`) عبر دوال مصنع متخصصة. تعتمد هذه الدوال على نظام `Depends()` في FastAPI لربط التبعيات تلقائياً، مع الحفاظ على تجريد الواجهات (Ports) مما يسمح بتبديل التنفيذات دون تعديل كود الطبقات العليا. يتميز التصميم بقدرته على التكيف مع متطلبات مختلفة، كاختيار نموذج الذكاء الاصطناعي عبر مُعامل استعلام، أو تبديل نظام تخزين الملفات بين المحلي والسحابي عبر إعدادات التطبيق. كما يضمن هذا النهج إدارة مركزية لجلسات قواعد البيانات وعمليات المصادقة، مع الحفاظ على بنية نظيفة تسمح لكل مكون بالتركيز على مسؤوليته دون معرفة تفاصيل تنفيذ التبعيات الأخرى.

## ٥-٦ معالجة طلبات التحليل

يُوضح المخطط ١٥ البنية الأساسية لنظام تحليل الرسوم البيانية، حيث يبدأ المستخدم بإرسال طلب التحليل (`analysis_request`) إلى النظام الذي يعالج الصورة أولاً عبر نموذج `OneChart` لاستخراج البيانات المنظمة بصيغة قاموس بايثون (`Python_Dictionary`). ثم تُدمج هذه البيانات مع الأسئلة النصية الطبيعية من المستخدم (`NaturalLanguageQuestion`) داخل النظام (`ChartAnalyzerSystem`)

لمعالجتها عبر نموذج اللغة الكبيرة مثلاً، DeepSeek والذي يُولد الإجابات النصية الدقيقة (NaturalLanguageAnswer) بناءً على التحليل الهيكلي واللغوي المشترك، ليعيدها أخيراً إلى المستخدم كاستجابة متكاملة (analysis\_response).

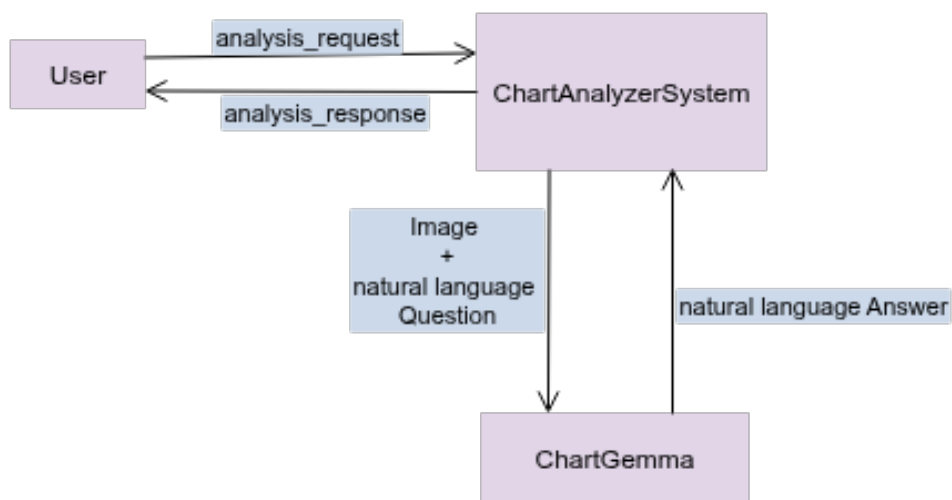
شكل ١٥: طلب تحليل المخطط - حالة OneChart+LLM



يُظهر المخطط ١٨ عملية تحليل الرسوم البيانية بشكل مباشر، حيث يرسل المستخدم طلب التحليل (analysis\_request) مع صورة الرسم البياني والسؤال النصي إلى نظام ChartAnalyzerSystem، الذي يعالجها مباشرةً باستخدام نموذج ChartGemma المتخصص في فهم الرسوم البيانية. يقوم ChartGemma بتحليل الصورة والسؤال معاً لتوليد إجابة نصية دقيقة (NaturalLanguageAnswer)، والتي تُعاد كاستجابة فورية (analysis\_response) إلى المستخدم دون حاجة لتحويل مسبق للبيانات إلى صيغة بسيطة.



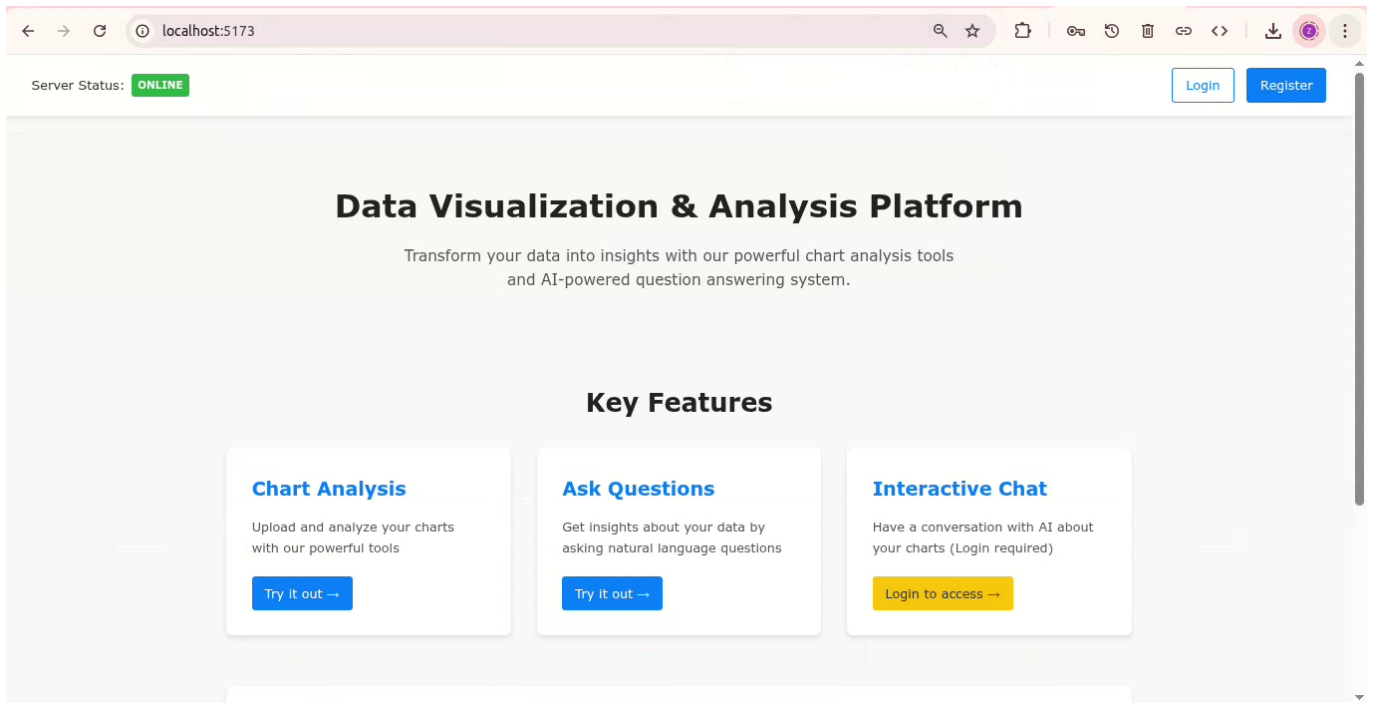
شكل ١٦: طلب تحليل المخطط - حالة ChartGemma



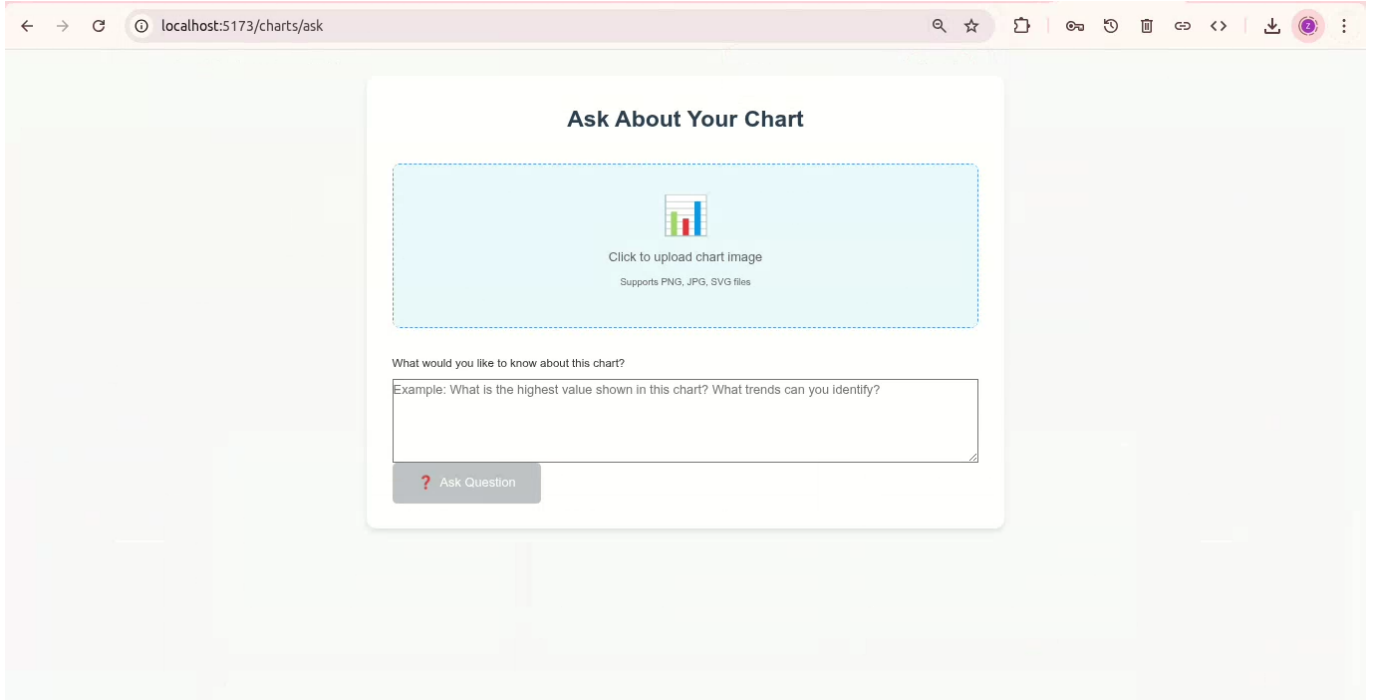
## ٧ تنفيذ واختبار النظام

نعرض فيما يلي حالة استخدام للتطبيق، بعد اختيار Ask Questions من القائمة الابتدائية.

شكل ١٧: الواجهة الابتدائية للتطبيق

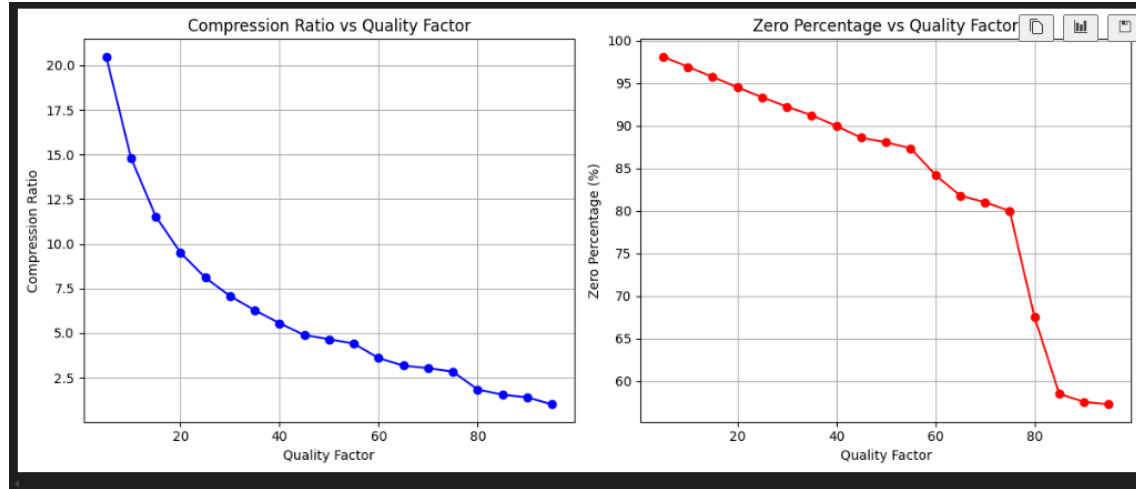


شكل ١٨: واجهة التطبيق لتحميل صورة وإدخال السؤال



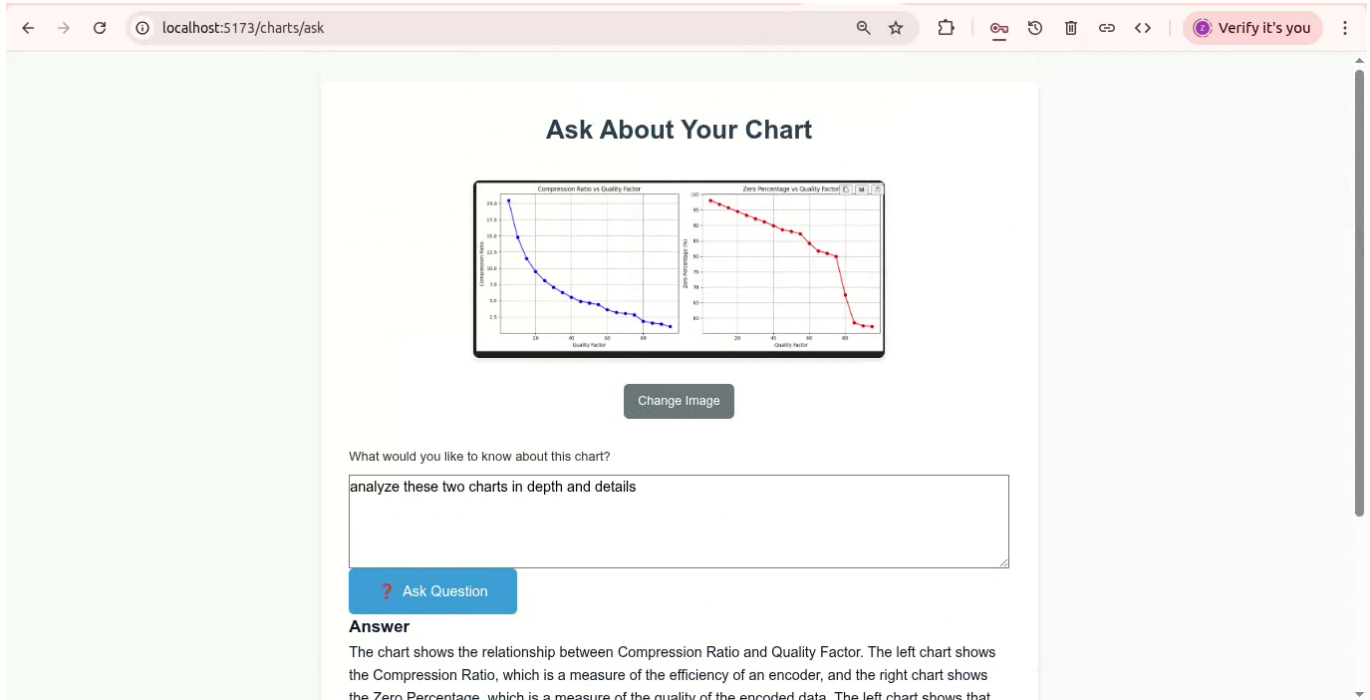
تم رفع صورة لمخططين بيانيين ١٩ بحيث يمثل الخط الأحمر تغير نسبة الأصفر في الصورة مع تغير معامل الجودة، أما المخطط الأزرق، يمثل تغير نسبة الضغط مع تغير معامل جودة الصورة.

شكل ١٩: صورة مخططين بيانيين لتجربة النظام



تم طرح السؤال : قم بتحليل هذين المخططين بعمق وتفصيل (analyze these two charts in depth and details)

## شكل ٢٠: رفع الصورة وإدخال السؤال



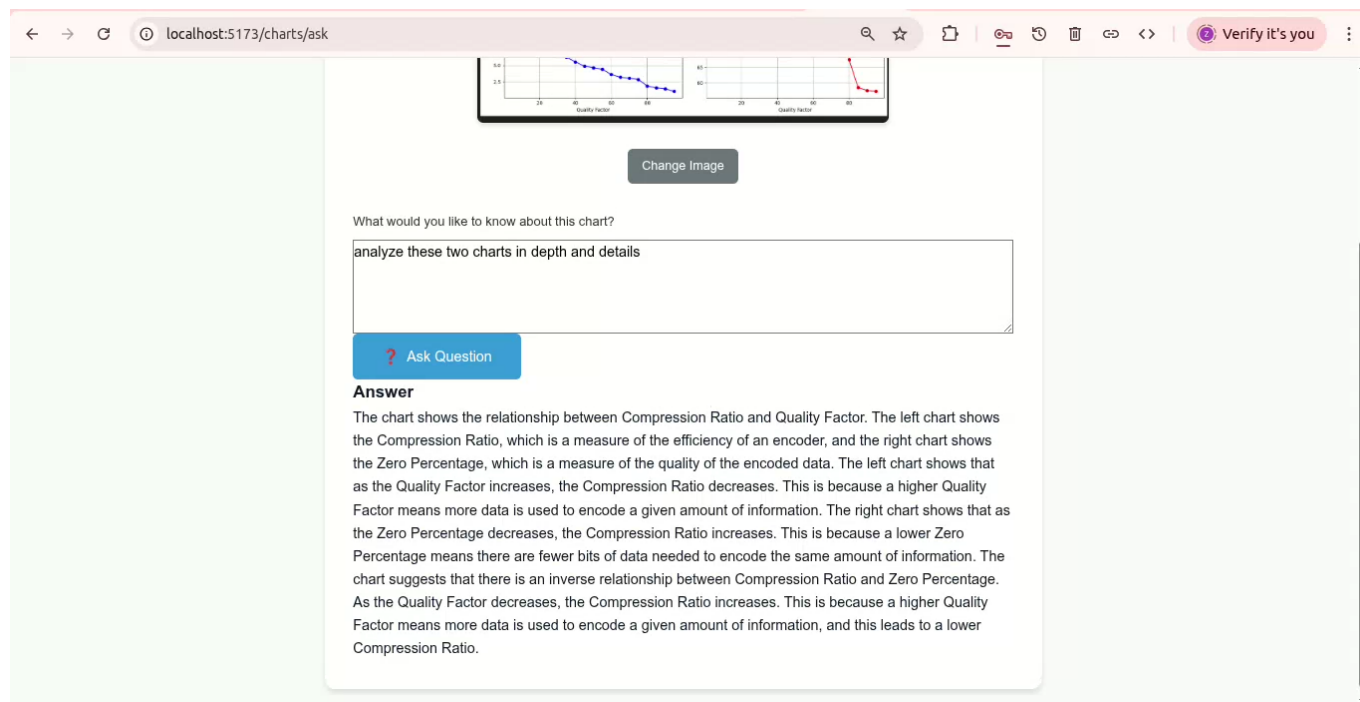
كانت الإجابة كالتالي (موضحة بالشكل ٢١):

### Answer

"The chart shows the relationship between Compression Ratio and Quality Factor. The left chart shows the Compression Ratio, which is a measure of the efficiency of an encoder, and the right chart shows the Zero Percentage, which is a measure of the quality of the encoded data. The left chart shows that as the Quality Factor increases, the Compression Ratio decreases. This is because a higher Quality Factor means more data is used to encode a given amount of information. The right chart shows that as the Zero Percentage decreases, the Compression Ratio increases. This is because a lower Zero Percentage means there are fewer bits of data needed to encode the same amount of information. The chart suggests that there is an inverse relationship between Compression Ratio and Zero Percentage. As the Quality Factor decreases, the Compression Ratio increases. This is because a higher Quality

Factor means more data is used to encode a given amount of information, and this leads to a lower Compression Ratio.”

### شكل ٢١: نتيجة التحليل



## ٨ الخاتمة والخطوات المستقبلية

### ٨-١ الخاتمة

في الختام، ركّز هذا المشروع على الاستفادة من النماذج المدربة مسبقاً في مجال تحليل الرسوم البيانية، دون الحاجة إلى تدريب نماذج جديدة من الصفر. اعتمدنا على حلول الذكاء الاصطناعي الجاهزة التي أثبتت كفاءتها في معالجة المخططات والرسوم البيانية، مع تعديلها لتناسب متطلباتنا الخاصة. هذا النهج وفر وقتاً وجهداً كبيرين، بينما حقق نتائج مرضية في تحويل المخططات الثابتة إلى تحليلات تفاعلية سهلة الفهم. من خلال هذه التجربة، تأكدنا من إمكانية تحقيق قيمة مضافة باستخدام الأدوات المتاحة، مع الحفاظ على المرونة الكافية للتكيف مع مختلف أنواع المخططات. كما أظهر المشروع كيف يمكن توظيف التقنيات الحالية بفعالية لتلبية احتياجات المستخدمين النهائيين. نرى في هذا المنحى فرصاً واعدة لمزيد من التطوير والتحسين، خاصة مع استمرار تطور النماذج الجاهزة وقدراتها. يظل التحدي الأبرز هو اختيار النموذج الأمثل لكل حالة استخدام، وضمان تكامله السلس مع أنظمة العمل الحالية.

### ٨-٢ الصعوبات

واجهنا خلال تنفيذ المشروع تحديات تقنية كبيرة لاسيما في التعامل مع النماذج الضخمة للذكاء الاصطناعي، وفيما يتعلق بضبطها وتكييفها مع المهام المطلوبة دون إعادة تدريب، حيث تطلبت فهماً عميقاً لبنيتها وآليات عملها. كما شكل تعلم أطر العمل الجديدة، تحدياً إضافياً بسبب تعقيدها وضيق الوقت المتاح لإتقانها. بالإضافة إلى ذلك، واجهنا صعوبات في دمج أدوات جديدة (مثل مكتبات معالجة الرسوم البيانية أو واجهات برمجة التطبيقات) مع النماذج الأساسية، مما استهلك وقتاً وجهداً كبيرين في التهيئة والاختبار.

### ٨-٣ الخطوات المستقبلية

يُمثل مشروعنا لتحليل وفهم المخططات البيانية لبنة أساسية في مجال ناشئ يتسع للكثير من التطوير والبحث. فبجانب فهم الرسوم البيانية آلياً لا يزال في مراحله المبكرة، مما يفتح آفاقاً واسعة للتحسين والتطوير، سواء على مستوى النماذج المتكاملة أو أنظمة استخراج المعلومات الجزئية. وفيما يلي أبرز المسارات المستقبلية الواعدة:

١. توسيع نطاق المخططات المدعومة: يشمل ذلك دعم أنواع متقدمة من المخططات (مثل المخططات التفاعلية، والرسوم البيانية ثلاثية الأبعاد، أو المخططات المخصصة لقطاعات محددة كالتمويل والهندسة)، مع تحسين دقة النماذج الحالية في التعامل مع المخططات المعقدة متعددة الطبقات.
  ٢. تعزيز قدرات الفهم السياقي: من خلال تطوير نماذج متعددة المهام قادرة على ربط المخططات بالنصوص المصاحبة (كالتقارير أو الدراسات)، مما يمكنها من تقديم تحليلات أكثر عمقاً تستند إلى السياق العام وليس البيانات المرئية فقط.
  ٣. الانتقال من التحليل إلى التوليد: تمهيد الطريق لنماذج قادرة ليس فقط على فهم المخططات، بل أيضاً على إنشاء مخططات جديدة مبنية على متطلبات المستخدم أو استنتاجات مستخلصة من البيانات النصية، مع تفسير منطق التصميم تلقائياً.
  ٤. حلول مخصصة للقطاعات: تطوير إضافات تخدم قطاعات محددة (مثل التعليم أو الصحافة الاستقصائية)، تتيح مثلاً للمعلمين إنشاء شروحات تفاعلية بناءً على المخططات، أو للصحفيين استخراج القصص الإخبارية من البيانات المرئية تلقائياً.
- مع التطور السريع في نماذج الذكاء الاصطناعي متعددة الوسائط، سيصبح مشروعنا نواة لمنصة شاملة تُعيد تعريف كيفية تفاعل البشر مع البيانات المرئية، بدءاً من التحليل الأساسي ووصولاً إلى الاستنتاجات المعقدة واتخاذ القرارات.



## ٩ قائمة الأشكال

### قائمة الأشكال

٩	بعض أنواع المخططات البيانية	١
١١	بعض نماذج المخططات في مجموعتي Pew و Statista [1]	٢
١٣	بنية النموذج ChartAdapter [4]	٣
١٤	بنية النموذج ChartGemma [5]	٤
١٨	بنية النموذج OneChart [11]	٥
١٩	مثال لاستخدام النموذج OneChart [11]	٦
٢٣	حالات الاستخدام	٧
٢٤	تتالي النظام 1	٨
٢٥	تتالي النظام ٢	٩
٢٧	بنية النظام	١٠
٢٨	بنية المعمارية التنظيمية	١١
٣٢	مخطط طبقات ومكونات النظام	١٢
٣٧	تصميم قاعدة البيانات	١٣
٣٨	نمط المستودع في البنية التنظيمية	١٤
٤٠	طلب تحليل المخطط - حالة OneChart+LLM	١٥
٤١	طلب تحليل المخطط - حالة ChartGemma	١٦
٤٢	الواجهة الابتدائية للتطبيق	١٧
٤٣	واجهة التطبيق لتحميل صورة وإدخال السؤال	١٨
٤٤	صورة مخططين بيانيين لتجربة النظام	١٩
٤٥	رفع الصورة وإدخال السؤال	٢٠
٤٦	نتيجة التحليل	٢١

## المراجع

- [١] Mirna Al-Shetairy, Hanan Hindy, Dina Khattab, and Mostafa M. Aref. Transformers utilization in chart understanding: A review of recent advances and future trends, .2024
- [٢] Shankar Kantharaj, Rixie Tffiany Ko Leong, Xiang Lin, Ahmed Masry, Megh Thakkar, Enamul Hoque, and Shafiq Joty. Chart-to-text: A large-scale benchmark for chart summarization, .2022
- [٣] Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. Chartqa: A benchmark for question answering about charts with visual and logical reasoning, .2022
- [٤] Peixin Xu, Yajuan Ding, and Wenqi Fan. Chartadapter: Large vision-language model for chart summarization. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, Hong Kong SAR, .2024 The Hong Kong Polytechnic University.
- [٥] Ahmed Masry, Megh Thakkar, Aayush Bajaj, Aaryaman Kartha, Enamul Hoque, and Shafiq Joty. Chartgemma: Visual instruction-tuning for chart reasoning in the wild. In *Proceedings of the Conference on Artificial Intelligence*, Canada, .2024 York University, Canada.
- [٦] Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, Thomas Unterthiner, Daniel Keysers, Skanda Koppula, Fangyu Liu, Adam Grycner, Alexey Gritsenko, Neil Houlsby, Manoj Kumar, Keran Rong, Julian Eisenschlos, Rishabh Kabra, Matthias Bauer, Matko Bošnjak, Xi Chen, Matthias Minderer, Paul Voigtlaender, Ioana Bica, Ivana Balazevic, Joan Puigcerver, Pinelopi Papalampidi, Olivier Henaff, Xi Xiong, Radu Soricut, Jeremiah Harmsen, and Xiaohua Zhai. Paligemma: A versatile 3b vlm for transfer, .2024
- [٧] Fanqing Meng, Wenqi Shao, Quanfeng Lu, Peng Gao, Kaipeng Zhang, Yu Qiao, and Ping Luo. Chartassistant: A universal chart multimodal language model via chart-to-table pre-training and multitask instruction tuning. In *Proceedings of the Conference on Artificial Intelligence*, Shanghai, China, .2024 OpenGVLab, Shanghai AI Laboratory.

- [8] Yucheng Han, Chi Zhang, Xin Chen, Xu Yang, Zhibin Wang, Gang Yu, Bin Fu, and Hanwang Zhang. Chartllama: A multimodal llm for chart understanding and generation. In *Proceedings of the Conference on Artificial Intelligence*, Singapore, .2023 Nanyang Technological University.
- [9] Liang Zhang, Anwen Hu, Haiyang Xu, Ming Yan, Yichen Xu, Qin Jin, Ji Zhang, and Fei Huang. Tinychart: Efficient chart understanding with visual token merging and program-of-thoughts learning. In *Proceedings of the Conference on Computer Vision*, Beijing, China, .2024 Renmin University of China.
- [10] Ahmed Masry, Mehrad Shahmohammadi, Md Rizwan Parvez, Enamul Hoque, and Shafiq Joty. Chartinstruct: Instruction tuning for chart comprehension and reasoning. In *Proceedings of the Conference on Artificial Intelligence*, Canada, .2024 York University, Canada.
- [11] Jinyue Chen, Lingyu Kong, Haoran Wei, Chenglong Liu, Zheng Ge, Liang Zhao, Jianjian Sun, Chunrui Han, and Xiangyu Zhang. Onechart: Purfiy the chart structural extraction via one auxiliary token, .2024