بسم الله الرحمن الرحيم

١٢

دولة فلسطين
وزارة التربية والتعليم

تصميم صفحات الويب

عملي ونظري

المسار المهني - الفرع الصّناعي

فريق التّأليف:

م. مؤمن كيلاني (منسقاً)

م. ريما دويكات

م. محمد عبد الهادي

م. وائل عواد

أ. نسرين دويكات

قررت وزارة التربية والتعليم في دولة فلسطين
تدريس هذا الكتاب في مدارسها بدءًا من العام الدراسي ٢٠٢٠/ ٢٠٢١ م

الإشراف العام

رئيس لجنة المناهج	أ. د. مروان عورتاني
نائب رئيس لجنة المناهج	د. بصري صالح
رئيس مركز المناهج	أ. ثروت زيد

الدائرة الفنية

إشراف فني	أ. كمال فحماوي
تصميم فني	أسحار حروب

تحرير لغوي	أ. وفاء جيوسي
متابعة المحافظات الجنوبية	د. سمية النخّالة

الطبعة التجريبية
٢٠٢١ م / ١٤٤٢ هـ

حقوق الطبع محفوظة ©
دولة فلسطين
وزارة التربية والتعليم

مركز المناهج

حي المصيون، شارع المعاهد
ص. ب 719 – رام الله – فلسطين
pcdc.edu.ps | pcdc.mohe@gmail.com

moehe.gov.ps | mohe.pna.ps | mohe.ps
.com/MinistryOfEducationWzartAltrbytWaltlym
هاتف +970-2-2983280 | فاكس +970-2-2983250

تـقـديـم

يتصف الإصلاح التربوي بأنه المدخل العقلاني العلمي النابع من ضرورات الحالة، المستند إلى واقعية النشأة، الأمر الذي انعكس على الرؤية الوطنية المطورة للنظام التعليمي الفلسطيني في محاكاة الخصوصية الفلسطينية والاحتياجات الاجتماعية، والعمل على إرساء قيم تعزز مفهوم المواطنة والمشاركة في بناء دولة القانون، من خلال عقد اجتماعي قائم على الحقوق والواجبات، يتفاعل المواطن معها، ويعي تراكيبها وأدواتها، ويسهم في صياغة برنامج إصلاح يحقق الآمال، ويلامس الأماني، ويرنو لتحقيق الغايات والأهداف.

ولما كانت المناهج أداة التربية في تطوير المشهد التربوي، بوصفها علماً له قواعده ومفاهيمه، فقد جاءت ضمن خطة متكاملة عالجت أركان العملية التعليمية التعلُّميّة بجميع جوانبها، بما يسهم في تجاوز تحديات النوعية بكل اقتدار، والإعداد لجيل قادر على مواجهة متطلبات عصر المعرفة، دون التورط بإشكالية التشتت بين العولمة والبحث عن الأصالة والانتماء، والانتقال إلى المشاركة الفاعلة في عالم يكون العيش فيه أكثر إنسانية وعدالة، وينعم بالرفاهية في وطن نحمله ونعظمه.

ومن منطلق الحرص على تجاوز نمطية تلقّي المعرفة، وصولاً لما يجب أن يكون من إنتاجها، وباستحضار واعٍ لعديد المنطلقات التي تحكم رؤيتنا للطالب الذي نريد، وللبنية المعرفية والفكريّة المتوخّاة، جاء تطوير المناهج الفلسطينية وفق رؤية محكومة بإطار قوامه الوصول إلى مجتمع فلسطيني ممتلك للقيم، والعلم، والثقافة، والتكنولوجيا، وتلبية المتطلبات الكفيلة بجعل تحقيق هذه الرؤية حقيقة واقعة، وهو ما كان له ليكون لولا التناغم بين الأهداف والغايات والمنطلقات والمرجعيات، فقد تآلفت وتكاملت؛ ليكون النتاج تعبيراً عن توليفة تحقّق المطلوب معرفياً وتربوياً وفكرياً.

ثمّة مرجعيات تؤطّر لهذا التطوير، بما يعزّز أخذ جزئية الكتب المقررّة من المنهاج دورها المأمول في التأسيس؛ لتوازن إبداعي خلّاق بين المطلوب معرفياً، وفكرياً، ووطنياً، وفي هذا الإطار جاءت المرجعيات التي تم الاستناد إليها، وفي طليعتها وثيقة الاستقلال والقانون الأساسي الفلسطيني، بالإضافة إلى وثيقة المنهاج الوطني الأول؛ لتوجّه الجهد، وتعكس ذاتها على مجمل المخرجات.

ومع إنجاز هذه المرحلة من الجهد، يغدو إزجاء الشكر للطواقم العاملة جميعها؛ من فرق التأليف والمراجعة، والتدقيق، والإشراف، والتصميم، وللجنة العليا أقل ما يمكن تقديمه، فقد تجاوزنا مرحلة الحديث عن التطوير، ونحن واثقون من تواصل هذه الحالة من العمل.

وزارة التربية والتعليم

مركز المناهج الفلسطينية

أيار / ٢٠١٩

مقدمة

مواكبةً للتطور العلمي والتكنولوجي في عصر المعرفة، أصبحت الحاجة لمهن جديدة ضرورة حتمية، وانطلاقاً من توجهات وزارة التربية والتعليم العالي بتحقيق تعلم وتعليم نوعي وريادي يلبي احتياجات المجتمع الفلسطيني وَفق أولوياته بالتحرير والتنمية، كان لا بد من تطوير مناهج التعليم المهني بِحُلتها الجديدة لتحاكي التوجهات العالمية بنشأة فلسطينية واقعية تعتمد العقلانية العلمية بالدمج بين الجانبين النظري والتطبيقي، مراعياً سوق العمل، وصولاً إلى جيل يمتلك الكفايات والقدرة على اكتشاف المعرفة العالمية، والإسهام في الإنتاج الكوني.

اعتمدت المناهج المهنية الجديدة منهجية الوحدات النمطية القائمة على الكفايات المهنية بمجالاتها الحرفية، والمنهجية، والاجتماعية والشخصية، وارتباط ذلك بسياقات واقعية حياتية تصف مواقف تعليمية تعلمية، توظف حل المشكلات بطريقة علمية وعملية، دون إغفال للجانب النظري المتضمن لأنشطة تعتمد على أداء الطلبة، والتأكيد على استراتيجيات التقويم التربوي الحديثة بما في ذلك التقويم الأصيل، والتحول من التقويم القائم على تحقيق الكفايات إلى تحقيق الجودة والامتياز، من خلال التركيز على شمولية أداء الطلبة، وتعزيز أنماط التفكير النقدي والتأملي.

جاء كتاب تصميم صفحات الويب للصف الثاني عشر في ثلاث وَحدات نمطية، تناولت الموضوعات الوحدة الأولى أساسيات لغة البرمجة (php)، والبرمجة الكينونية في تطوير صفحات الويب، ووضّحت الوَحدة الثانية كيفية التعامل مع قواعد البيانات، وربطها في مواقع الإنترنت، وإنشاء موقع تفاعلي، أمّا الوحدة الثالثة فبيّنت آلية التعامل مع خادمات الويب، ولوحة التحكم، وحجز اسم مجال للموقع، ورفعه، والتحديث عليه، وإنشاء موقع باستخدام الوورد برسّ (WordPress).

وإذ نقدّم هذا الكتاب بين أيدي ذوي العلاقة من معلمات ومعلمين وطلبة وتربويين ومهتمين، نرجو تحقيق الغايات التربوية المنشودة، وبعقلية منفتحة نحو التطوير والتعديل، فإنه يسرّنا استقبال أيّ ملاحظة تهدف إلى تطوير الكتاب وتجويده.

فريق التّأليف

المحتويات

لغة PHP 2

أساسيّات لغة PHP 6

توظيف (البرمجة كائنيّة التوجّه) في PHP 44

قواعد البيانات 68

مفاهيم أساسيّة في قواعد البيانات 71

إنشاء وإدارة قواعد البيانات 91

الموقع التفاعليّ 120

عَوْلمة الموقع 166

خادمات الويب ولوحة التحكُّم بالموقع الإلكترونيّ 169

حجز اسم مجال للموقع ونشره على خادم ويب مجّانيّ، والتحديث عليه. 192

إنشاء موقع من خلال نظام إدارة المحتوى ووردبريس (WordPress). 210

الوَحدة الأولى

لغة PHP

أنـاقــش

تُعدّ لغة PHP من أقوى اللغات البرمجيّة في مجال تطوير مواقع الإنترنت التفاعليّة.

يُتوقَّع من الطلبة بعد دراسة هذه الوحدة، والتفاعل مع أنشطتها أن يكونوا قادرين على توظيف لغة (PHP) في تطوير موقع إنترنت تفاعليّ، من خلال الآتي:

- استخدام أساسيّات لغة PHP.
- برمجة الكائنات oop.

الكفايات المهنيّة

الكفايات المهنيّة المتوقَّع من الطلبة امتلاكها بعد انتهاء دراسة هذه الوحدة، والتفاعل مع أنشطتها:

أوّلاً- الكفايات الحرفيّة:

1- تعريف لغة php (مقدّمة عن php).
2- تنصيب السيرفر xamp.
3- استدعاء الملفات وتحميلها (include, require).
4- جملة الطباعة Echo.
5- المتغيّرات والثوابت.
6- إرسال البيانات get & post.
7- الجمل الشرطيّة (if & switch).
8- جمل التكرار (for....).
9- المصفوفات والمصفوفات ذات المفاتيح، والدوال functions.
10- الخصائص والثوابت.
11- الوراثة.

ثانياً- الكفاياتُ الاجتماعيّةُ والشخصيّةُ:

1- العمل ضمن فريق.
2- تقبُّل التغذية الراجعة.
3- احترام رأي الآخرين.
4- مصداقيّة التّعامُل مع الزبون.
5- حِفْظُ خصوصيّة الزبون.
6- الاستعداد للاستجابة لطلبات الزبون.
7- التحلّي بأخلاقيّات المهنة أثناء العمل.
8- التمتُّع بالفكر الريادي أثناء العمل.
9- استخلاص النتائج ودقّة الملاحظة.
10- الاتّصال والتّواصُل الفعّال.
11- الاستعداد للاستفسار والاستفادة من ذوي الخبرة.
12- الإفادة من مقترحات الآخرين.
13- امتلاك مهارة التأمّل والتأمُّل الذاتيّ.
14- الاستعداد التام لتقبُّل ملحوظات الزبون وانتقاداته.
15- القدرة على الإقناع.

ثالثاً- الكفاياتُ المنهجيّة:

1- توظيف التعلُّم التعاونيّ.
2- القدرة على إدارة الحوار وتنظيم النقاش.
3- القدرة على توظيف البحث العلميّ وأساليبه.
4- القدرة على إعداد خطط.
5- القدرة على استمطار الأفكار.
6- تقويم النتائج وعرضها.
7- جمع البيانات وتحليلها وتبويبها.
8- توظيف المنهج العلميّ في حلّ المشكلات، وتمثيل البيانات، وجمع البيانات وتحليلها وتبويبها.

قواعد الأمن والسلامة:

1- الالتزام بالجلسة الصحيحة أمام الحاسوب.
2- نظافة الأيدي والأسطح التي يتمّ وضعُ أجهزة الحاسوب عليها.
3- أن يكون مكان العمل ذا تهوية وإضاءة جيّدة.
4- عدم استخدام كوابل متعريّة لتوصيل جهاز الحاسوب بالكهرباء.
5- تَرْك العمل على الجهاز كلّ ساعتين مدّة ربع ساعة، والبدء بالمشي في الغرفة.

6- استخدام برامج مكافحة الفيروسات والجدران الناريّ (FireWalls).
7- عدم التجاوب مع الرسائل الإلكترونيّة المشبوهة.
8- أن تعرف جيّداً مع مَن تتعامل قبل الكشف عن أيّة معلومات.
9- استخدام كلمة سِرّ صعبة التخمين تتكوّن من أحرف كبيرة وصغيرة، ورموز وأرقام.
10- عدم استخدام المعلومات العامّة في كلمة السرّ، كتواريخ الميلاد، وأرقام الهواتف، وأسماء الأبناء.
11- عدم استخدام خاصيّة التخزين التلقائيّ للمعلومات الشخصيّة على الحواسيب التي لا تخصّك، في حال استخدامها.

الموقف التعليميّ التعلُّميّ (1-1):

أساسيّات لغة PHP

وصف الموقف التعليميّ:

يرغب صاحب مركز تعليميّ عقد دورة تدريبيّة حول لغة PHP وأساسيّاتها، فطلب من فنّي تصميم صفحات الإنترنت في المدرسة المِهْنيّة المجاورة مساعدته.

العمل الكامل			
خطوات العمل	وصف الموقف الصفّيّ	المنهجيّة (استراتيجيّة التعلُّم)	الموارد
أجمع البيانات وأحلّلها	أجمع البيانات من صاحب المركز التعليميّ عن: • المادّة التعليميّة التي ستتضمّنها الدورة (حجمها، موضوعها، الفئة المستهدفة منها، طريقة عرضها، الأنشطة العمليّة والمادّة النظريّة التي يجب أن تشملها)، ووسائط الإيضاح التي تتضمّنها، صور، روابط، نشرات، ... أجمع البيانات عن لغة PHP: • تعريفها، ومراحل تطوّرها، واستخداماتها. • مميّزات لغة PHP. • تنصيب السيرفر xampp • استدعاء الملفّات وتحميلها (include, require) • Echo جملة الطباعة. • المتغيّرات والثوابت. • إرسال البيانات عبرget & post • الجمل الشرطيّة(if & switch) • جمل التكرار (for....). • المصفوفات. • الدوال functions + دوال مهمّة للمصفوفات.	• الحوار والمناقشة. • التعلّم التعاونيّ (مجموعات العمل). • البحث العلميّ.	• وثائق: الطلب الخاصّ بصاحب المركز التعليميّ، نشرات، مقالات، كتب عن لغة PHP • التكنولوجيا: الإنترنت، مقاطع فيديو، مواقع إنترنت موثوقة عن لغة PHP.

أخطّط وأقرّر	• تصنيف البيانات الخاصّة بـِ : -لغة PHP (تعريفها، ومراحل تطوّرها، ومميّزاتها). • تحديد كيفيّة: 1- تنصيب السيرفر xamp. 2- استدعاء الملفات وتحميلها (include, require). 3- echo. 4- المتغيّرات والثوابت. 5- إرسال البيانات get & post. 6- الجمل الشرطيّة(if & switch). 7- جمل التكرار (for....) 8- المصفوفات والمصفوفات ذات المفاتيح الدوال functions+ دوال مهمّة للمصفوفات. • توضيح استخدامات، ومكوّنات لغة PHP. • تحديد طرق كتابة الأكواد، وإضافة أنماط لغة PHP في تصميم صفحات الويب. • عقد جلسة مع صاحب المركز لمناقشة مقترحات للمادّة الأوّليّة التي تمّ إعدادها على الـ (CD)، من حيث الشكل والمضمون. • إعداد خطّة توضّح الجدول الزمنيّ ومراحل التنفيذ.	• التعلُّم التعاونيّ/ مجموعات العمل. • حوار ومناقشة. • عصف ذهنيّ استمطار الأفكار.	• وثائق: الطلب الخاصّ بصاحب المركز، نشرات، مقالات، كتب عن لغة PHP. • البيانات التي تمّ جمعها.
أنفّذ	إعداد وصف يوضّح نشأة لغة PHP وتطوّرها، ومميّزاتها، وفوائدها، واستخداماتها. تبيان الإجراءات المتّبعة في: • كيفيّة تنزيل وتنصيب السيرفر xamp. • آليّات استدعاء الملفات وتحميلها (include, require). 1- echo. 2- جمل طباعة المتغيّرات والثوابت. 3- جمل طباعة وإرسال البيانات get & post. 4- القواعد الأساسيّة لكتابة الجمل الشرطيّة (if & switch)، وجمل التكرار (for....). 5- المصفوفات والمصفوفات ذات المفاتيح الدوال functions + دوال مهمّة للمصفوفات.	• التعلُّم التعاونيّ / مجموعات العمل. • حوار ومناقشة. • عصف ذهنيّ.	• الوثائق: الطلب الخاصّ بصاحب المركز، نشرات، مقالات، كتب عن لغة PHP. • التكنولوجيا: الإنترنت، مقاطع فيديو، مواقع إنترنت موثوقة عن لغة PHP. • حاسوب.

	• طرق كتابة الأكواد، وإضافة أنماط لغة PHP في تصميم صفحات الويب. • تصميم عرض يوضّح تعريف لغة PHP، ومراحل تطوّرها، وأساسيّاتها واستخداماتها. • عقد جلسة مع صاحب المركز للاتفاق على المادّة التي تمّ إعدادها، من حيث الشكل والمضمون (تطوّر اللغة، استخداماتها، الأمثلة العمليّة). • إجراء التعديلات المقترحة على طريقة تنفيذ الأنشطة، أو حجم المادّة، وطريقة تنظيمها، وإخراجها بشكل نهائيّ.		
أتحقّق	• التحقّق ممّا يأتي: 1- الصياغة المناسبة والصحيحة لتعريف لغة PHP، ونشأتها، وتطوّرها. 2- تحديد الاستخدامات الشائعة للغة PHP وفوائدها. 3- دقّة الأمثلة العمليّة التي تمّ إعدادها. 4- مطابقة وشمول الـ CD على جميع العناصر المتّفق عليها مع صاحب المركز.	• التعلُّم التعاونيّ / مجموعات العمل. • حوار ومناقشة.	• الوثائق: الطلب الخاصّ بطلّاب مدرسة الحريّة ؟؟، نشرات، مقالات، كتب عن لغة PHP. • التكنولوجيا: الإنترنت، مقاطع فيديو، مواقع إنترنت موثوقة عن لغة PHP. • حاسوب.
أوثّق وأقدّم	توثيق ما يأتي: • نشأة لغة PHP وتطوّرها، ومميّزاتها، وفوائدها، واستخداماتها. • آليّة تنصيب السيرفر xamp. • كيفيّة استدعاء الملفّات وتحميلها (include, require) 1- echo 2- جمل طباعة المتغيّرات والثوابت. 3- جمل طباعة وإرسال البيانات get & post. 4- القواعد الأساسيّة لكتابة الجمل الشرطيّة(if & switch) ، وجمل التكرار (for....). 5- المصفوفات والمصفوفات ذات المفاتيح الدوال functions + دوال مهمّة للمصفوفات، والأنشطة العمليّة التي تمّ إعدادها. • إعداد العروض التقديميّة. • إخراج المادّة التعليميّة (الـ CD) بصورتها النهائيّة. • فتح سجلّ خاصّ بلغة PHP استخداماتها ومميّزاتها.	• التعلُّم التعاونيّ / مجموعات العمل. • حوار ومناقشة.	• حاسوب، سجلّات.

أقيّم	• رضا صاحب مركز التدريب عن المادّة التدريبيّة الخاصّة باستخدامات لغة PHP. • مطابقة المادّة التدريبيّة للمعايير والمواصفات.	• حوار ومناقشة. • البحث العلميّ.	• الوثائق: الطلب الخاصّ بصاحب مركز التدريب. • المعايير والمواصفات الخاصّة بإعداد الموادّ التعليميّة.

سؤال: كيف تتمّ كتابة الأكواد البرمجيّة بلغة PHP؟

أتعلّم

أساسيّات لغة PHP

نشاط (1) نظريّ: كيف يمكن استخدام لغة PHP لإنشاء موقع إنترنت تفاعليّ مع المستخدم؟

لغة PHP (Hypertext PreProcesser):

لغة برمجيّة نصّيّة ومفتوحة المصدر، تنفّذ جهة الخادم (server) لتطوير صفحات إنترنت تفاعليّة وديناميكيّة.

أفكّر لغة PHP مفتوحة المصدر.

- من مميّزات لغة (PHP):

1- سهلة الاستخدام.
2- تتوافق مع أنظمة التشغيل، والمِنصّات المختلفة (Mac OS X، Unix، Linux، Windows، إلخ).
3- تُستخدم مع جميع أنواع خوادم الإنترنت المستخدمة حاليّاً، مثل: (IIS ، Apache، إلخ).
4- تدعم مجموعة كبيرة من قواعد البيانات للموقع الإلكترونيّ.
5- يمكن تعلّمها بسهولة ويسر.

- من وظائف لغة (PHP):

1- تطوير صفحات إنترنت ديناميكيّة وتفاعليّة، مع ضبط صلاحيّات الاستخدام.
2- إدارة الملفّات (إنشاء ملفّ جديد، التعديل عليه،...).
3- حفظ وجمع بيانات نماذج HTML.
4- إرسال واستقبال ملفّات تعريف الارتباط (cookies).
5- الربط مع قواعد البيانات الخاصة بالموقع الإلكترونيّ.
6- تأمين وحماية بيانات الموقع الإلكترونيّ.

نشاط (2) نظريّ: باستخدام إحدى مصادرالمعرفة أبحث في إصدارات لغة PHP، وإضافات الإصدار (7).

الأدوات اللازمة لتعلّم لغة PHP:

1- المتصفّح مثل: (Google Chrome, FireFox, Opera, Internet Explorer,...)
2- محرّر الأكواد ويوجد منها أنواع عدّة:
(Visual Code, Komodo Edit, Netbeans, Brackets,...)
3- حزمة التطبيقات التي تتضمّن خادم الويب (web server)، مثل: (Lamp, Mamp, Wamp, Xampp,...).

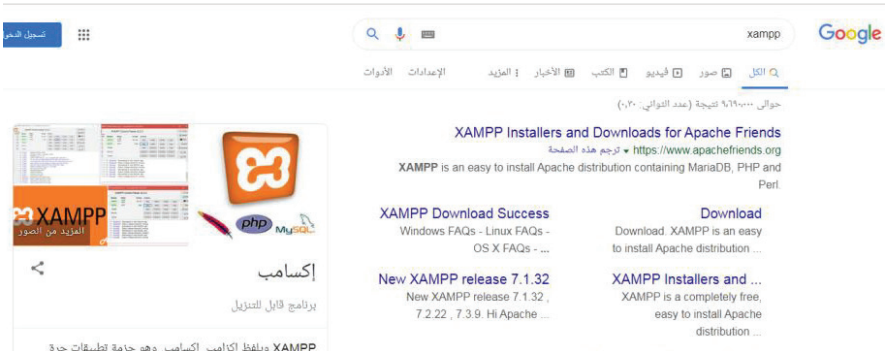
- تهيئة بيئة العمل:

1- محرر الأكواد فيجوال ستوديو كود (Visual Studio Code).
2- حزمة (Xampp):

تنزيل وتنصيب Xampp

يستخدم هذا البرنامج، ويعمل مع كلّ أنواع أنظمة التشغيل (لينكس ، ماك، ويندوز،...)، ويحتوي على خادم ويب مجّانيّ (Apache)، وقاعدة بيانات (mysql) مجّانيّة.

1- كتابة xampp في محرّك البحث، ثمّ اختيار الخيار الأوّل، كما في الصورة الآتية:



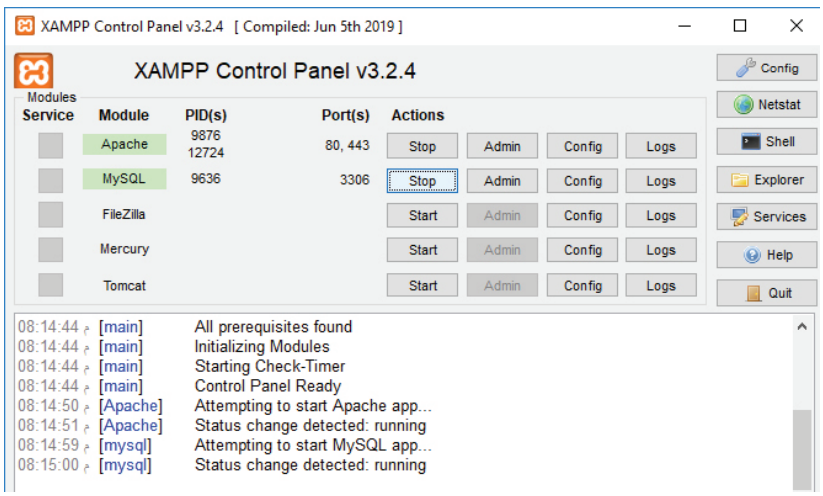
2- اختيار نظام التشغيل بالضغط على XAMPP for Windows 7.3.9 (PHP 7.3.9) كما يأتي:

3- عند اكتمال التنزيل فتح ملفّ التشغيل، ومتابعة التثبيت بالضغط على next.

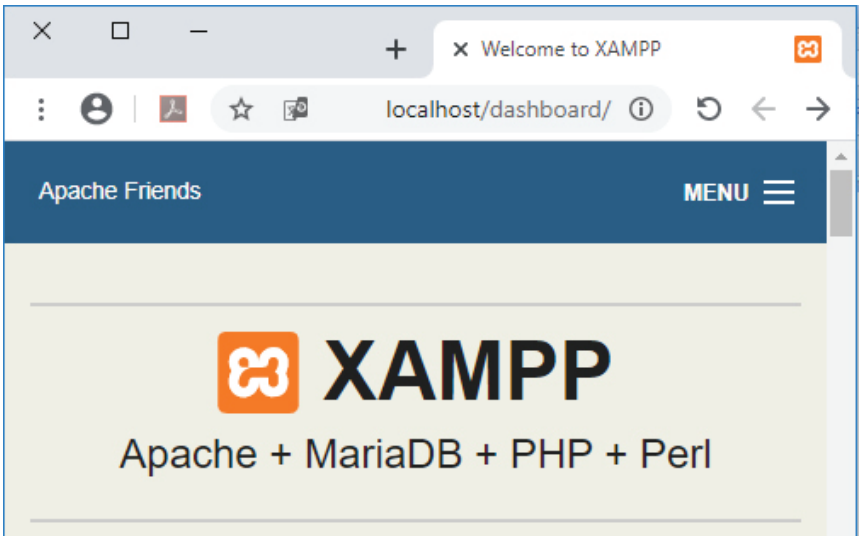
4- بعد انتهاء التثبيت والضغط على كلمة finish تظهر نافذة لاختيار اللغة، كما يأتي:

طريقة استخدام حزمة XAMPP:

1- عند فتح الحزمة تظهر نافذة لوحة التحكُّم بالخادم XAMPP Control Panel، ثمّ تفعيل Apache and MySQL بالضغط على زرّ الأمر start حتّى يظهران باللون الأخضر؛ أي أنّه تمّ تفعيل الخادم Apache وقاعدة البيانات MySQL، كما يأتي:



2- للتأكُّد من عمل الخادم، من خلال المتصفّح كتابة localhost، ثمّ الضغط على زرّ الأمر enter، لتظهر الصفحة الرئيسيّة للخادم كما يأتي:



3- حفظ ملفات مشروع الويب المراد تشغيلها بواسطة الخادم داخل ملفّ htdocs، الموجود ضمن المسار التالي C:\xampp\htdocs كما يأتي:

أساسيّات لغة PHP:

- البنية الأساسيّة للغة PHP (PHP SYNTAX):

تنفّذ أوامر البرنامج النصّيّ PHP على الخادم، ثمّ ترسل نتيجة ملفّ HTML العادي إلى المتصفّح مرّة أخرى. ويتمّ تعريف نصّ PHP ووضعه في أيّ مكان داخل مستند وملفّ الموقع الإلكترونيّ.

- **الصيغة العامّة:** تُكتَب الجمل البرمجيّة بلغة PHP داخل المحدّد الآتي:

```
<?php
    //الاكواد البرمجية
?>
```

- بدون هذا المحدّد لا يتمّ تنفيذ أوامر وأكواد لغة PHP المكتوب، ولا يفهمها الخادم.
- من المهمّ كتابة وسم البداية (<?php)، ثمّ الأكواد، والجمل البرمجيّة بلغة PHP، ثمّ كتابة وسم النهاية (?>)، وعدم نسيانه لتجنُّب الأخطاء في التنفيذ .
- حفظ الملفّ بامتداد (.php)، وإلّا لا تنفّذ جمل PHP، وتنفّذ أكواد HTML فقط.
- عند كتابة الجمل البرمجيّة بلغة PHP يجب أن تنتهي بالفاصلة المنقوطة (;).

جملة الطباعة بلغة PHP:

يمكن استخدام العبارة Echo أوprint لطباعة البيانات بلغة PHP.

يُفضَّل استخدام العبارة Echo أكثر من print؛ لأنّها أسرع في تنفيذ الأوامر. وفيما يأتي خصائصها:

1- تُستعمَل مع كلّ أنواع البيانات النصّيّة والرقميّة و... .
2- ليست من دوال اللغة، بل هي من البنية الأساسيّة الخاصّة بلغة PHP.
3- يمكن طباعة وإخراج كلّ البيانات بالصفحة بأمر echo واحد.

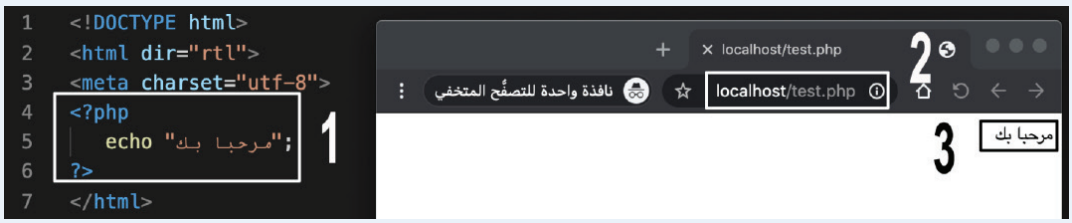
مثال (1): أنشئ صفحة باسم test.php تحتوي على النصّ ("مرحبا بك "):

أوّلاً- تشغيل الخادم في XAMPP.

ثانياً- عمل مجلّد جديد بالمسار التالي C:\xampp\htdocs، مثلاً: (test).

ثالثاً- فتح برنامج الفيجوال ستوديو كود، وإنشاء ملفّ جديد، ثمّ حفظه باسم test.php داخل المجلّد الذي أُنشئ في الخطوة السابقة.

رابعاً- أكتب الكود الآتي:



1- أكتب الجملة باستدعاء كلمة الطباعة echo داخل وسم php.
2- فتح المتصفّح وكتابة localhost/test.php في شريط العنوان، والضغط على entre من لوحة المفاتيح.
3- ظهور الناتج، فإذا ظهرت الكتابة على شكل رموز أضع كود php داخل وسم html وأستخدم العنصر meta، كما في المثال.

أتذكّر

ما فائدة الخاصيّة "dir = "rtl في المثال السابق؟

نشاط (3) عمليّ: أنشئ صفحات الإنترنت التالية، باستخدام echo و print:

1- استخدام جملة الطباعة echo: أفتح الفيجوال ستوديو كود، وأنشئ ملفّاً جديداً وأسمّيه (first.php)، ثمّ أكتب كما يأتي، وألاحظ النتيجة:

```
<?php
echo "<h2>النجاح</h2>";
echo " الحياة مليئة بالحجارة<br>";
echo " .....فلا تتعثر بها.....<br>";
  echo " ... بل اجمعها .....<br>";
  echo " ..وابن بها سلماً.....<br>";
echo " .... تصعد به نحو النجاح ....";
?>
```

النجاح

الحياة مليئة بالحجارة
.....فلا تتعثر بها.....
.....بل اجمعها ...
.....وابن بها سلماً..
.... تصعد به نحو النجاح ....

2- استخدام جملة الطباعة print: أفتح الفيجوال ستوديو كود، وأنشئ ملفّاً جديداً وأسمّيه (second.php)، ثمّ أكتب كما يأتي وألاحظ النتائج:

```
<?php
print "<h2>العلم</h2>";
print "من ذاق ظلمة الجهل <br>";
print " أدرك أن العلم نور ";
?>
```

العلم

من ذاق ظلمة الجهل
أدرك أن العلم نور

كتابة التعليقات بلغة PHP:

التعليق هو الجزء المكتوب داخل الكود البرمجيّ، ولا ينفّذ عند تشغيله.

تكتب التعليقات داخل الكود لتساعد على فهمه من قبل المبرمجين الآخرين، والتعديل عليه.

تكتب التعليقات بلغة PHP بأكثر من طريقة:

1- تعليق سطر واحد (Single-Line comments) داخل الكود البرمجيّ، استخدام: الرمز (//) :

```
<? php
الطريقة الاولى لكتابة تعليق سطر واحد باستخدام هذا الرمز//
?>
```

أو الرمز (#):

```
<? php
#الطريقة الثانية لكتابة تعليق سطر واحد باستخدام هذا الرمز
?>
```

2- تعليق عدّة أسطر (Multiple-Line comments)، أستخدم الرمز (/* */) كما يأتي:

```
<? php
/*طريقة كتابة تعليق
يتكون من
عدة أسطر
*/
?>
```

3- أستخدم رمز التعليق متعدّد الأسطر، لترك جزء من الكود، وعدم تنفيذه بعد تشغيل الكود البرمجيّ، كما في الكود الآتي:

```
<?php
$y = 6 /* -22  */ + 40/2;
echo $y;
?>
```

وتكون نتيجة تنفيذ الكود (الرقم 26).

نشاط (4) عمليّ: أفتح الفيجوال ستوديو كود، وأنشئ ملفّاً جديداً وأسمّيه، ثمّ أكتب كما يأتي، وألاحظ النتيجة، وأدوّنها:

```
<?php
$y = 6  -22   + 40/2;
echo $y;
?>
```

استدعاء الملفّات وتحميلها (include, require):

تُستخدم لتضمين ملفّ معيّن بكلّ ما يحتويه من (نصّ، كود،...) داخل ملفّ آخر يحوي جملة التضمين (include)، أو (require)، ولها ميّزات عدّة، منها:

- تسمح باستدعاء ملفّ PHP أو HTML واستخدامه في صفحات متعدّدة من صفحات موقع الإنترنت. مثلاً: يمكن إنشاء ملفّ لترويسة الموقع يحتوي القائمة الرئيسيّة للموقع، واستخدامه في مختلف صفحات الموقع من خلال استدعائه فقط، وعند الحاجة إلى التعديل على القائمة الرئيسيّة للموقع في مختلف الصفحات يتمّ ذلك على ملفّ الترويسة المستدعى.
- يمكن إدراج محتوى ملفّ PHP واحد كامل في ملفّ PHP آخر، قبل أن ينفّذه الخادم.
- عند استدعاء محتوى ملفّ PHP واحد في ملفّ أخرPHP لا يؤثّر استخدام دالّة التضمين include، أو دالّة الطلب require، فكلاهما تستدعي الملفّ وتحمّله وتنفّذه داخل الملفّ المحتوي الدالّة.
- عند حدوث فشل في تحميل الملفّ المطلوب يظهر الخطأ كما يأتي:
- دالّة الطلب require: يظهر رسالة خطأ (E_COMPILE_ERROR)، ولا يظهر محتويات صفحة الويب الاخرى.
- دالّة التضمين include: يصدر رسالة تحذير (E_WARNING)، وتظهر محتويات صفحة الويب الأخرى.
- استخدام الدالّة include عند تطوير صفحة لإظهار النتيجة فقط للمستخدم حتى لو كان ملفّ التضمين مفقوداً، أمّا الدالّة require فتُستخدم عند تطوير أكواد PHP للتطبيقات المعقّدة.

- الصيغة العامّة لاستدعاء الملفّات وتحميلها:

```
    include 'filename';
// أو
  require 'filename';
```

نشاط (5) عمليّ: أنشئ صفحة إنترنت تحتوي على قائمة التنقّل الرئيسيّة للموقع، ثمّ أستدعيها بملفّات صفحات الإنترنت الأخرى باستخدام جمل التضمين (include, require)، وألاحظ النتيجة.

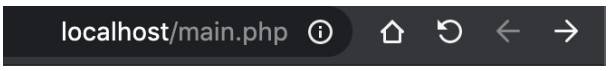
1- أفتح الفيجوال ستوديو كود، وأُنشئ ملفّاً جديداً وأسمّيه (menu.php)، ثمّ أكتب كما يأتي:

```
<?php
    echo '<a href= "/main.php">الرئيسية</a> -
    <a href= "/html.php">HTML</a> -
    <a href= "/css.php">CSS</a> -
    <a href= "/js.php">JavaScript</a> -
    <a href= "/php.php">PHP</a>';
?>
```

2- أنشئ ملفّاً جديداً باسم main.php، وأستدعي ملفّ القائمة (menu.php) كما يأتي:

```php
<!DOCTYPE html>
<html dir="rtl">
<meta charset="utf-8">
<body>
<div class="main">
<?php
  include 'menu.php'
?>
<h1>الصفحة الرئيسية</h1>
</div>
</body>
</html>
```

الناتج:



HTML - CSS - JavaScript - PHP - الرئيسية

الصفحة الرئيسية

3- أستخدم الدالّة require بدلاً من الدالّة include، لاستدعاء الملفّات وتحميلها وألاحظ النتائج.

4- أستخدم الدالّة include، لاستدعاء ملفّ (noFileExists.php) غير الموجود، وأُنفّذ الكود البرمجيّ الآتي، وألاحظ النتيجة:

```php
<!DOCTYPE html>
<html>
<body>

<h1>Welcome to my home page!</h1>
<?php include 'noFileExists.php';
echo "I have a $color $car.";
?>

</body>
</html>
```

النتيجة:

Welcome to my home page!

I have a .

5- أستخدم الدالّة require، لاستدعاء ملفّ (noFileExists.php) غير الموجود، وألاحظ النتيجة.

نشاط (6) عمليّ: أُنشئ صفحة إنترنت تحتوي على بيت الشعر (وما نيل المطالب بالتمنّي **** ولكن تُؤخذ الدنيا غلابا)، باستخدام لغة PHP، وأستخدم جمل التضمين لأستدعي ملفّ الصفحة بصفحات إنترنت أخرى.**

المتغيّرات (Variables) بلغة PHP:

اسم يشير لموقع داخل ذاكرة الحاسوب، ويُستخدم لتخزين البيانات فيه، أو قراءتها منه، أو التعديل عليها، والبيانات المستخدمة لها نمط (نوع) معيّن، مثل: بيانات رقميّة، أو نصّيّة، أو منطقيّة.

تعريف المتغيّر بلغة PHP: كتابة علامة ($) متبوعة باسم المتغيّر ($ Variablename)

مثال (2): تعريف عدّة متغيّرات في لغة PHP:

```php
<?php
$txt = "  PHP لغة  تعلم";
$x = 12;
$y = 1.275;
?>
```

- أهمّ ميّزات المتغيّرات بلغة PHP:

- عندما تكون قيمة المتغيّر نصّيّة توضع القيمة داخل علامة الاقتباس (" ")، أو (' ').
- تختلف لغة PHP عن لغات البرمجة الأخرى بعدم وجود أمر لإعلان المتغيّر، حيث يتمّ إنشاء المتغيّر لأوّل مرّة لحظة تحديد قيمته.

- شروط تعريف المتغيّرات بلغة PHP:

1- يبدأ المتغيّر بعلامة ($) متبوعاً باسم المتغيّر، دون وجود فراغ بينها.
2- يجب أن يبدأ اسم المتغيّر بحرف، أو الشرطة السفليّة (_).
3- لا يبدأ اسم المتغيّر برقم.
4- لا يحتوي اسم المتغيّر على فراغات أو رموز فقط، بل يحتوي على الأحرف الإنجليزيّة والأرقام والشرطة السفليّة (A-z , 0-9, _).
5- أن لا يكون اسم المتغيّر مشابهاً لكلمة محجوزة بلغة، وهي الأوامر، والجمل المعرّفة مسبقاً في اللغة، ولها عمل خاصّ، مثل كلمة echo.
6- يتأثّر اسم المتغيّر بحساسية حالة كتابة الحرف (case-sensitive)، فمثلاً: اسم المتغيّر ($age) يختلف عن اسم المتغيّر ($ AGE).

نشاط (7) نظريّ: أُصنّف أسماء المتغيّرات الآتية إلى اسم متغيّر مسموح، أو غير مسموح تعريفه، مع ذكر السبب:

اسم المتغيّر (Variable name)	مسموح أو غير مسموح تعريفه	السبب
1- 1first_student		
2- final exam	غير مسموح	يحتوي على فراغ
3- employee@com		
4- Year19		
5- #Ebook		

- جملة طباعة المتغيّرات (Variables) بلغة PHP:

استخدام الأمر echo لطباعة بيانات المتغيّر، وإخراجها على الشاشة كما في الكود الآتي:

```
<?php
$txt = "الصف الثاني عشر مهني ";
echo "تصميم صفحات الويب $txt!";
?>
```

تصميم صفحات الويب الصف الثاني عشر مهني !

أو باستخدام (.) النقطة، لدمج نصّين، كما يأتي:

```
<?php
$txt = "الصف الثاني عشر مهني ";
echo "تصميم صفحات الويب" .$txt. "!";
?>
```

تصميم صفحات الويب الصف الثاني عشر مهني !

أنواع البيانات في لغة PHP:

المتغيّرات بلغة PHP تخزّن بيانات من أنواع مختلفة (نصّيّة، أو رقميّة، أو ...)، موضّحة كالآتي:

1- النصّيّة (String): يُكتَب النصّ داخل علامة الاقتباس المفردة (' ')، أو المزدوجة (" ").
2- الرقميّة (Integer): الأعداد الصحيحة.
3- الكسور الحقيقيّة والعشريّة (Double، Float).
4- المنطقيّة (Boolean): تكون الإجابة منطقيّة صواب (true) أو خطأ (false).
5- المصفوفة (Array).
6- الكائنات (Object).
7- القيم الفارغة (NULL).
8- المصادر (Resource).

- **معرفة نوع البيانات:**

1- استخدام الدالّة var_dump:

تُستعمَل هذه الدالّة لمعرفة نوع البيانات وإظهار قيمة البيانات المخزّنة على المتصفّح .

2- الدالّة gettype ( ):

تُستخدَم هذه الدالّة لمعرفة نوع البيانات فقط دون طباعة القيم المخزّنة على المتصفّح.

نشاط (8) عمليّ: معرفة نوع بيانات قيم المتغيّر بلغة PHP.

1- باستخدام الدالّة var_dump ( ): أفتح الفيجوال ستوديو كود، وأنشئ ملفّاً جديداً وأسمّيه، ثمّ أكتب كما يأتي:

```php
<?php
$a = 5.234;   // عدد بفاصلة عشرية
var_dump($a);
echo "<br>";
$b = -214; // رقم سالب
var_dump($b);
echo "<br>";
$c = "العلم بالصغر كالنقش بالحجر";   // نصوص
var_dump($c);
echo "<br>";
$boo=true; // قيم منطقية
var_dump($boo);
echo "<br>";
?>
```

```
float(5.234)
int(-214)
string(49) "العلم بالصغر كالنقش بالحجر"
bool(true)
```

2- باستخدام الدالّة gettype ( ): استبدالها مكان الدالّة var_dump ( ) في الكود السابق، ثمّ أنفّذ الكود، وألاحظ النتيجة.

الثوابت بلغة (PHP Constants):

تعريف قيمة معيّنة باسم محدّد ثابت له (المعرّف)، ولا تتغيّر قيمته.

- يختلف الثابت عن المتغيّر، كلاهما يعرّف قيمة معيّنة، ولكنّ المتغيّر تتغيّر قيمته في الكود الواحد بحسب استخدامه، أمّا الثابت فتبقى قيمته ثابتة ولا تتغيّر.
- يُعدّ الثابت ذا مجال عام global في كلّ البرنامج بعكس المتغيّر.
- يُعرَّف الثابت بلغة PHP باستخدام الكلمة المحجوزة const متبوعة باسم الثابت، أو باستخدام الدالّة define.
- يخضع اسم الثابت لشروط تسمية اسم المتغيّر نفسها، دون البدء بعلامة $ ، ويُفضَّل أن يكتب الثابت بالحروف الكبيرة.

- الصيغة العامّة لتعريف الثابت:

استخدم الدالّة define:

```
define(NAME,valu,cas_insensitive)
```

- name: تحديد اسم الثابت.
- value: تعريف قيمة الثابت.
- case-insensitive: تحدّد إذا كان اسم الثابت غير حسّاس لحالة الأحرف عند كتابته (قيمته false بشكل افتراضيّ).

نشاط (9) عمليّ: أفتح الفيجوال ستوديو كود، وأنشئ ملفّاً جديداً وأسمّيه، ثمّ أكتب الكود الآتي، وأنفّذه:

```
<?php
 const NAME1 ="User1";
 define("NAME2","User2");
 echo NAME1."  ".NAME2;
?>
```

- أُعدّل على جملة الطباعة في الكود البرمجيّ السابق، كما يأتي:

```
 echo NAME1."  ".name2;
```

ألاحظ حدوث خطأ عند تنفيذ الكود السابق؛ لأنّه تمّ تعريف اسم الثابت NAME2 كحالة حسّاسة لكتابة الحرف case-sensitive.

- أعيد تعريف اسم الثابت NAME2 ليصبح غير حسّاس لحالة كتابة الأحرف case-insensitive، كما يأتي:

```
<?php
 const NAME1 ="User1";
 define("NAME2","User2",true);
 echo NAME1."  ".name2;
?>
```

العمليّات بلغة (PHP):

تُعدّ العمليّات ضروريّة في لغة PHP لإجراء تعديلات على قيم البيانات والمتغيّرات، ومن هذه العمليّات:

1- العمليّات الحسابيّة Arithmetic operators:

الجدول الآتي يوضّح أهمّ العمليّات الحسابيّة المستخدمة في لغة PHP:

العملية الحسابيّة	الاسم	النتيجة	مثال(a=6, b=2)
+	الجمع	مجموع المتغيّرين	$a + $b = 8
-	الطرح	طرح المتغيّرين	$a - $b = 4
*	الضرب	ضرب المتغيّرين	$b = 12 * $a
/	القسمة	قسمة المتغيّرين	$b = 3 / $a
%	باقي القسمة	الباقي من $a مقسوماً على $b)	$b = 0 % $a
**	الأس (القوّة)	المتغيّر $a هو الأساس مرفوع له القوّة ($b)	$b = 36 ** $a

2- عمليّات تعيين القيم والإحالة Assignment operators:

يمكن تغيير قيم المتغيّرات، وإعادة تعيينها، باستخدام العلامة(=)، وبالتالي فإنّ المتغيّر في اليسار سوف يُضبَط بحسب القيمة على اليمين، ويمكن تنفيذ عدّة عمليّات حسابيّة بهذه الطريقة، كما في الجدول الآتي:

عمليّة تعيين القيم	طريقة أخرى لكتاباتها	الوصف	مثال
a = b	a = b	المساواة	$a = 10
a+ = b	a = a + b	الجمع	$a = 10 $a+ = 50
a- = b	a = a - b	الطرح	$a =10 $a - = 50
a* = b	a = a * b	الضرب	$a = 10 $a * = 50
a = b	a = a / b	القسمة	$a = 10 $a / = 50
a = b	a = a % b	باقي القسمة	$a = 10 $a % = 50

نشاط (10) عمليّ: أستخدم فيجوال ستوديو كود، وأُنشئ ملفّا جديداً، وأكتب الكود الآتي، ثمّ أنفّذه، وألاحظ النتيجة:

```php
<?php
$a = 40;
echo $a;echo "<br>";
$a += 8;
echo $a;echo "<br>";
$a -= 6;
echo $a;echo "<br>";
$a *= 4;
echo $a;echo "<br>";
$a /= 4;
echo $a;echo "<br>";
$a %= 4;
echo $a;echo "<br>";
$a **= 4;
echo $a;echo "<br>";
?>
```

3- عمليّات المقارنة (Comparison operators):

تُستخدم للمقارنة بين قيمتين، سواء كانت قيماً رقميّة، أو نصّيّة، وتكون النتيجة صحيحة أو خاطئة. ومن عمليّات المقارنة ما يظهر في الجدول الآتي:

عملية المقارنة	الاسم	النتيجة	مثال
==	تساوي	صحيحة، عندما تكون قيمة المتغيّر الأوّل $a تساوي قيمة المتغيّر الثاني $b.	$a == $b
===	متطابقة	صحيحة، عندما تكون قيمة المتغيّر الأوّل $a تساوي قيمة المتغيّر الثاني $b، ولهما نوع البيانات نفسه.	$a === $b
=!	لا تساوي	صحيحة، إذا كانت قيمة المتغيّر الأوّل $a لا تساوي قيمة المتغيّر الثاني $b.	$a != $b
< >	لا تساوي	صحيحة، إذا كانت قيمة المتغيّر الأوّل $a لا تساوي قيمة المتغيّر الثاني $b.	$a < > $b
==!	غير متطابقة	صحيحة، إذا كانت قيمة المتغيّر الأوّل $a لا تساوي قيمة المتغيّر الثاني $b، أو مختلفين بنوع البيانات لهما.	$a !== $b
<	أكبر من	صحيحة، إذا كانت قيمة المتغيّر الأوّل $a أكبر من قيمة المتغيّر الثاني $b.	$a > $b
>	أصغر من	صحيحة، إذا كانت قيمة المتغيّر الأوّل $a أصغر من قيمة المتغيّر الثاني $b.	$a < $b

=<	أكبر أو يساوي	صحيحة، إذا كانت قيمة المتغيّر الأوّل $a أكبر من، أو تساوي قيمة المتغيّر الثاني $b.	$a >= $b
=>	أصغر أو يساوي	صحيحة، إذا كانت قيمة المتغيّر الأوّل $a أصغر من، أو تساوي قيمة المتغيّر الثاني $b.	$a <= $b

نشاط (11) عمليّ: أستخدم فيجوال ستوديو كود، وأُنشئ ملفّا جديداً، وأكتب الكود الآتي، ثمّ أنفّذه، وألاحظ النتيجة:

```php
<?php
$a = 12;
$b = '12';
var_dump($a == $b);echo "<br>";
var_dump($a === $b);echo "<br>";
var_dump($a != $b);echo "<br>";
var_dump($a <> $b);echo "<br>";
var_dump($a !== $b);echo "<br>";

$x = 30;
$y = 7;
var_dump($x > $y);echo "<br>";
var_dump($x < $y);echo "<br>";
var_dump($x >= $y);echo "<br>";
var_dump($x <= $y);echo "<br>";
?>
```

var_dump: تظهر معلومات عن المتغيّرات، نوعها وقيمتها.

4- عمليّات الزيادة أو النقصان لقيم المتغيّرات (Increment/Decrement operators):

تُستخدم عمليّات الزيادة لزيادة قيمة المتغيّر، وعمليّات النقصان لإنقاص القيمة الأصليّة للمتغير، كما في الجدول الآتي:

العملية	الاسم	النتيجة
++$a	الزيادة قبل	زيادة قيمة المتغيّر بواحد ، ثمّ إرجاع قيمته.
$a++	الزيادة بعد	إرجاع قيمة المتغيّر، ثمّ زيادة قيمته بواحد.
--$a	النقصان قبل	نقصان قيمة المتغيّر بواحد ، ثمّ إرجاع قيمته.
$a--	النقصان بعد	إرجاع قيمة المتغيّر، ثم إنقاص قيمة a بواحد.

نشاط (12) عمليّ: أفتح الفيجوال ستوديو كود، وأنشئ ملفّاً جديداً وأسمّيه، ثمّ أكتب كما يأتي، وألاحظ النتيجة بعد تنفيذه:

```php
<?php
$a = 44;
echo ++$a;echo "<br>";
echo $a++;echo "<br>";
echo $a;echo "<br>";
echo --$a;echo "<br>";
echo $a--;echo "<br>";
echo $a;echo "<br>";
?>
```

5- العمليّات النصّيّة (String operators):

لربط النصوص والسلاسل النصّيّة يوجد طريقتان، كما هو موضّح في الجدول الآتي:

العملية	الاسم	النتيجة	مثال
.	Concatenation	دمج النصّ الموجود في المتغيّر $text1، بالنصّ في المتغيّر $text2.	$text1. $text2
.=	Concatenation assignment	يضيف النصّ الموجود في المتغيّر $text2، إلى النصّ في المتغيّر $text1.	$text1.= $text2

مثال (3): توضيح العمليّات النصّيّة، كما يأتي:

```php
<?php
$text1 = 'العلم زين فكن للعلم مكتسبًا';
$text2 = ' وكن له طالبًا ما عشت مقتبسًا';

// ربط النص الأول بالثاني
echo $text1 . $text2;echo "<br>";
//إضافة قيمة المتغير الثاني الى المتغير الاول
$text1 .= $text2;
echo $text1;
?>
```

نشاط (13) عمليّ:

أ- أفتح الفيجوال ستوديو كود، وأنشئ ملفّاً جديداً وأسمّيه، ثمّ أكتب كما يأتي:

```php
<?php
$a = 20;$b = 4;
echo $a + $b; echo "عملية جمع<br>";//عملية جمع
//عملية طرح
echo $a - $b;echo "عملية طرح <br>";
//عملية ضرب
echo $a * $b;echo "عملية ضرب <br>";
//عملية قسمة
echo $a / $b; echo "عملية قسمة<br>";
//عملية Modulus
echo $a % $b; echo "الباقي <br>";
//عملية أسية Exponentiation
echo $a ** $b ;;echo "<br>";
echo $a; echo "<br>";
//بعد تنفيذ عملية جمع a تعيين قيمة جديدة للمتغير
$a += 2;
echo $a;echo "<br>";
//بعد تنفيذ عملية طرح a تعيين قيمة جديدة للمتغير
$a -= 2;echo $a; echo "<br>";
?>
```

```
24عملية جمع
16عملية طرح
80عملية ضرب
5عملية قسمة
0الباقي
160000
20
22
20
```

ب- أبحث عن أكواد أخرى للعمليّات المختلفة بلغة PHP.

إرسال البيانات get & post:

تُنفَّذ لغة php وتعمل جهة الخادم (Web Server) وليس جهة المتصفّح؛ لهذا عند إجراء بعض العمليّات في الموقع الإلكترونيّ مثل تسجيل الدخول، والبحث عن شيء ...إلخ، ترسَل البيانات والمعلومات بين الخادم والمتصفّح.

توجد طريقتان لإرسال البيانات من متصفّح الزبون (Client) إلى خادم الويب (Web Server):

- **طريقة Get:** تُستخدَم بكثرة في محرّكات البحث للموقع الإلكترونيّ للبحث عن موضوع معيّن، عضو، مقطع،...
- **طريقة Post:** تُستخدم لإضافة معلومات لقواعد البيانات، مثل: تسجيل الدخول، وتسجيل حساب جديد.

قبل إرسال المتصفّح البيانات والمعلومات إلى خادم الويب ، يشفّرها بطريقة نظام الترميز (URL)، باستخدامه يتمّ ربط كلّ زوج من الاسم وقيمته بعلامة التساوي، ويفصل بين الأزواج والقيم المختلفة باستخدام علامة الضمّ .ampersand.

```
name1=value1&name2=value2&name3=value3
```

ويتم إزالة المسافات في الكود وتستبدل بإشارة (+)، ويحوّل الأحرف والقيم غير الرقميّة إلى نظام السادس عشر (hexadecimal)، وبعد تشفير (encoded) البيانات يرسلها المتصفّح إلى الخادم.

- طريقة POST:

تنقل هذه الطريقة البيانات وترسلها عبر رأس الصفحة HTTP Header، وتشفّر البيانات قبل إرسالها كما في طريقة GET، وتوضع في رأسHeader ويُسمّى QUERY_STRING.

خصائص استخدام POST طريقة لطلب بيانات الصفحة:

- لا يوجد قيود على حجم البيانات المراد إرسالها بهذه الطريقة.
- يمكن أن تُستخدَم هذه الطريقة لإرسال بيانات بنظام ASCII ونظام الثنائيّ.
- البيانات المرسلة بهذه الطريقة تمرّ عبر HTTP Header؛ لذلك يعتمد حماية البيانات المرسلة وأمنها على بروتوكول HTTP، وبالتالي باستعمال Secure HTTP تأكّد أنّ معلوماتك المرسلة محميّة وآمنة.
- يمكنك استخدام المصفوفة POST _ $ للوصول إلى جميع البيانات والمعلومات المرسلة بطريقة POST.

تُستخدَم هذه الدالّة لطلب البيانات وتمريرها من صفحة إلى أخرى، داخل صفحات الموقع الإلكترونيّ بسهولة، وذلك باستقبال البيانات من خلال المتغيّرات.

مثال (4): أنشئ صفحة تسجيل في موقع، وإرسال البيانات المسجّلة لصفحة أخرى باستخدام post.

أوّلاً- أُنشئ مجلّداً وأسمّيه، ثمّ أُنشئ ملفّاً جديداً في الفيجوال ستوديو كود، وأُسمّيه (registration.html)، ثمّ أكتب كما يأتي:

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="utf-8">
        <title>post</title>
    </head>
  <body>
      <form action="data.php" method="POST">
          <input type="text" name="name" placeholder="name" /><br/>
          <input type="email" name="email" placeholder="example@mail.com"/><br/>
          <input type="password" name="password" placeholder="password"/><br/>
          الجنس:
          <br/>
          <input type="radio" name="gender" value="male"/> ذكر<br/>
          <input type="radio" name="gender" value="fmale"/> انثى<br/>
          الدولة:
          <select name="country">
              <option value="palestine">فلسطين</option>
              <option value="egpt">مصر</option>
              <option value="jordan">الاردن</option>
          </select><br/>
```

```
22          الهوايات:
23          <br/>
24          <input type="checkbox" name="football" value="كرة القدم"/>
25          كرة القدم<br/>
26          <input type="checkbox" name="swim" value="السباحة"/>
27          السباحة<br/>
28          <input type="checkbox" name="sleep" value="النوم"/>
29          النوم<br/>
30          <input type="checkbox" name="read" value=" القراءة"/>
31          القراءة<br/>
32          <input type="submit" value="ارسال"/>
33      </form>
34  </body>
35  </html>
```

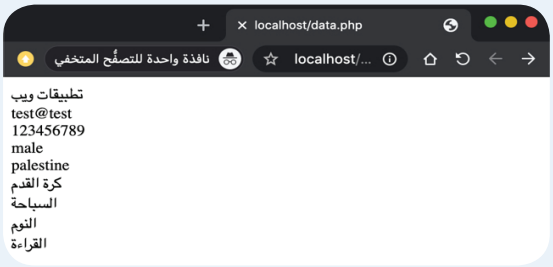
- السطر 8 يكتب في الخاصيّة action اسم الصفحة التي سترسل إليها البيانات، والخاصيّة و method تستقبل الطريقة التي ستُرسل بها.
- السطر 32 استخدام submit وليس button؛ لأنّ الأولى وظيفتها جلب جميع البيانات الموجودة داخل form.

ثانياً- انشاء ملفّ جديد وتسميته data.php، وأكتب فيه كالآتي:

```php
<?php
echo $_POST['name']."<br/>";
echo $_POST['email']."<br/>";
echo $_POST['password']."<br/>";
echo $_POST['gender']."<br/>";
echo $_POST['country']."<br/>";
echo $_POST['football']."<br/>";
echo $_POST['swim']."<br/>";
echo $_POST['sleep']."<br/>";
echo $_POST['read']."<br/>";
?>
```

- بما أنّه تمّ استخدام post أكتب المتغيّر POST _ $ لاستقبال البيانات المرسلة من النموذج.
- للوصول إلى البيانات أستخدم المكتوب داخل خاصيّة name في كلّ مرّة.

ثالثاً- أفتح صفحة التسجيل، وأملأ البيانات، ثمّ أضغط على أرسل، وألاحظ النتائج كما يأتي:



- طريقة GET:

تتمّ هذه الطريقة بإرسال المعلومات والبيانات المشفّرة من متصفّح الزبون، ملحقة بطلب الصفحة. بهذه الطريقة تُستخدَم الإشارة (؟) لتفصل بين الصفحة والبيانات المشفّرة، كما في المثال الآتي:

```
http://www.test.com/index.htm?name1=value1&name2=value2
```

خصائص استخدام طريقة GET لطلب بيانات الصفحة:

- تنتج هذه الطريقة سلسلة نصّيّة طويلة تظهر في سجلّات الخادم الخاصّ بك، في مربّع البحث للمتصفّح المستعرض، أي تظهر البيانات المرسلة في رابط الصفحة، بعد تمريرها وطلبها.
- تسمح بإرسال البيانات حتى 1024 حرفاً فقط.
- إذا أردت إرسال اسم المستخدم وكلمة المرور، أو أي معلومات حسّاسة وسريّة أخرى لا تستخدم طريقة GET، لإرسالها إلى الخادم.
- لا تستطيع استخدام هذه الطريقة لإرسال البيانات الرقميّة الثنائيّة، مثل الصور إلى الخادم.
- من الممكن أن تستخدم (QUERY_STRING)، لمعرفة البيانات المرسلة والوصول إليها.
- من خلال هذه الطريقة يمكن استخدام المصفوفة GET _ $ للوصول إلى البيانات المرسلة بطريقة GET.

نشاط (14) عمليّ: أُعدّل على المثال السابق باستخدام GET، مع العلم أنّ البيانات تنقل عن طريق الرابط كالآتي:

```
/data.php?name=test&email=test%40test&password=1234&gender=male&country=palestine&football=كرة+القدم&swi
```

- بعد اختبار الطريقة، هل تُعدّ GET أكثر أماناً في هذا المثال، ولماذا؟

الجمل الشرطيّة بلغة PHP:

تنفّذ الجملة الشرطيّة الأكواد والعمليّات البرمجيّة إذا تحقّق الشرط، وإذا لم يتحقّق الشرط لا تنفّذ هذه العمليّات البرمجيّة.

- الجملة الشرطيّة (if Statement):

```
if(الشرط){
    //ينفذ الكود إذا كان الشرط صحيحاً
}
```

مثال (5): أنشئ صفحة تسجيل في موقع، وإرسال البيانات المسجّلة لصفحة أخرى باستخدام post.

أكتب الكود البرمجيّ المناسب لطباعة جملة (صباح يوم سعيد مليء بالتفاؤل)، إذا كانت القيمة أقلّ من 12، كما يأتي:

```
$time = 7;
if($time < 12){
    echo "صباح يوم سعيد مليء بالتفاؤل";
}
```

الناتج: صباح يوم سعيد مليء بالتفاؤل

- العمليّات المنطقيّة (Logical operators):

إمّا أن تكون الإجابة صحيحة أو خاطئة فقط، وتُستخدَم غالباً مع الجمل الشرطيّة.

تُستخدَم العمليّات المنطقيّة، لدمج العبارات والجمل الشرطيّة، كما في الجدول الآتي:

العمليّة المنطقيّة	الاسم	النتيجة	مثال
and	و	صحيحة، إذا كانت قيمتا المتغيّرين a و b صحيحتين.	$a and $b
Or	أو	صحيحة، إذا كانت قيمة أحد المتغيّرين سواء (a أو b) صحيحة، ويمكن أن تكون القيمتان صحيحتين.	$a or $b
xor	xor	صحيحة، إذا كانت قيمة أحد المتغيّرين سواء (a أو b) صحيحة فقط، وليس كلاهما قيمتهما صحيحة.	$a xor $b
&&	و	صحيحة، إذا كانت قيمتا المتغيّرين a و b صحيحتين.	$a && $b
||	أو	صحيحة، إذا كانت قيمة أحد المتغيّرين سواء (a أو b) صحيحة. يمكن أن تكون القيمتان صحيحتين.	$a || $b
!	ليس	صحيحة، إذا كانت قيمة المتغيّر الأوّل $a ليست صحيحة.	a$!

نشاط (15) عمليّ:

• أفتح الفيجوال ستوديو كود، وأنشئ ملفّاً جديداً وأسمّيه، ثمّ أكتب كما يأتي، وألاحظ النتيجة بعد تنفيذه:

```php
<?php
$a = 200;  $b = 77;
if ($a == 200 and $b == 77) {
 echo "ليس الجمال بأثواب تزيننا، إن الجمال جمال العلم والادب ";
 echo "<br>";}
if ($a == 200 or $b == 10) {
     echo "العلم نور ";
     echo "<br>";}
if ($a == 200 xor $b == 10) {
     echo "الجهل ظلام ";
     echo "<br>";}
if ($a == 200 && $b == 77) {
     echo "  بالعلم والاخلاق ترتقي الامم ";
     echo "<br>";}
if ($a == 90 || $b == 77) {
     echo " الضيق مفتاح الفرج";echo "<br>";}
if ($a != 28) {
     echo "القناعة كنز لا يفنى";
     echo "<br>";}
?>
```

• أُعدّل على قيم المتغيّرات السابقة، وأُلاحظ النتيجة.

- الجملة الشرطيّة (if...else Statement):

إذا لم يتحقّق الشرط تنفّذ أوامر وتعليمات برمجيّة أخرى، كالآتي:

```php
if(الشرط){
  //ينفذ الكود إذا كان الشرط صحيحاً
}else{
    //ينفذ الكود إذا كان الشرط خاطئاً
}
```

نشاط (16) عمليّ: أنفّذ الكود البرمجيّ الآتي، وألاحظ النتيجة:

```php
<?php
   $time = 15;
   if($time < 12){
     echo "صباح الخير";
   }else{
       echo "مساء الخير";
   }
?>
```

- الجملة الشرطيّة (if...elseif...else Statement):

تنفيذ أوامر عدّة، وتعليمات برمجيّة إذا تحقّق أحد الشروط.

```
if(الشرط الأول){
 //ينفذ الكود إذا كان الشرط الأول صحيحاً
}elseif(الشرط الثاني) {
//ينفذ الكود إذا كان الشرط الثاني صحيحاً
}else{
//ينفذ الكود إذا كانت جميع الشروط خاطئة
}
```

نشاط (17) عمليّ: أُنفّذ الكود البرمجيّ الآتي، وألاحظ النتيجة:

```
$avg = 85;
if($avg >= 90){
  echo "ممتاز";
}elseif ($avg >= 80 && $avg < 90){
    echo "جيد جدا";
}else {
    echo "راسب";
}
```

- الجملة الشرطيّة (switch Statement):

تنفيذ إجراءات مختلفة بناءً على شروط وظروف مختلفة.

القاعدة الأساسيّة:

```
switch(x){
 case l1:
   // x= l1 ينفذ الكود اذا كانت
 break;
 case l2:
    // x= l2 ينفذ الكود اذا كانت
 break;
 .
 .
 .
 default:
 // ينفذ الكود اذا لم يتحقق اي شرط مما سبق
 }
```

نشاط (18) عمليّ: أنفّذ الكود البرمجيّ الآتي، ثمّ أكتب قيماً مختلفة للمتغيّر ($favcolor) وألا حظ النتيجة، وأدوّنها في كلّ مرّة:

```php
<?php
$favcolor = "green";

switch ($favcolor) {
    case "red":
        echo "اللون المفضل هو اللون الاحمر";
        break;
    case "blue":
        echo " اللون المفضل هو اللون الازرق";
        break;
    case "green":
        echo "اللون المفضل هو اللون الاخضر";
        break;
    default:
        echo "لونك المفضل ليس اللون الاحمر او الازرق او الاخضر";
}
?>
```

اللون المفضل هو اللون الاخضر

جمل التكرار بلغة PHP:

تقلّل الجمل التكراريّة حجم الكود البرمجيّ، بحيث إذا وُجدت عمليّات متكرّرة تُجمع في جمله برمجيّة واحدة تفيد التكرار، ومنها:

PHP Loops	الوصف
While	تكرار مجموعة أوامر برمجيّة داخلها ما دام ناتج الشرط صحيحاً.
do...while	تنفّذ مجموعة أكواد وأوامر داخل هذه الشيفرة مرّة واحدة، ثمّ تتمّ عمليّة تكرار الأوامر ما دام الشرط صحيحاً.
for	يتمّ تكرار مجموعة أوامر وأكواد داخل هذه الشيفرة لعدد معيّن من المرّات.
Foreach	تُستخدَم هذه الحلقة أو الشيفرة مع المصفوفات arrays، وباستخدامها يتمّ تكرار مجموعة أكواد وأوامر،على كلّ عنصر من عناصر المصفوفة.

- حلقة التكرار while (Syntax while loop):

علمت سابقاً أنّ جملة تكرار while تنفّذ أكواد وأوامر برمجيّة مرّات عدّة ما دام الشرط متحقّقاً true.

```
while(الشرط){
  //ينفذ الكود متكررا ما دام الشرط صحيحاً
}
```

مثال (6): أكتب الكود البرمجيّ الآتي باستخدام while loop، ثمّ أنفّذه، وألاحظ النتيجة:

```
<!DOCTYPE html>
<html>
<body>

<?php
$y = 2;

while($y <= 10) {
    echo "The number is: $y <br>";
    $y++;
}
?>

</body>
</html>
```

```
The number is: 2
The number is: 3
The number is: 4
The number is: 5
The number is: 6
The number is: 7
The number is: 8
The number is: 9
The number is: 10
```

- تعريف متغيّر باسم y، وإسناد القيمة 2 له، يعمل كعدّاد.
- داخل جملة while وُضع شرط أن تكون قيمة المتغيّر y أقلّ، أو تساوي 10.
- تتكرّر عمليّة تنفيذ الأكواد والأوامر داخل جملة while، مادام ناتج الشرط صحيحاً.
- داخل جملة while تزداد قيمة العدّاد y بواحد، بعد كلّ تنفيذ لجملة الشرط، من خلال المعامل ++.
- بعد تنفيذ حلقة للمرّة التاسعة تصبح قيمة y تساوي 11، وعند التحقّق من أوامر الشرط تكون خاطئة false، وبالتالي لا تنفّذ جملة while.

- جملة التكرار do...while (Syntax do....while loop):

جملة تكرار do...while تنفّذ أكواد وأوامر برمجيّة معيّنة أوّلاً مرّة واحدة على الأقلّ، ثمّ تتحقّق من صحّة الشرط.

```
do{
//ينفذ الكود مرة على الاقل ثم تتكرر العملية ما دام الشرط صحيحاً
}while(الشرط)
```

نشاط (19) عمليّ:

أ- أكتب الكود البرمجيّ الآتي باستخدام do...while loop، ثمّ أنفّذه، وألاحظ النتيجة:

```
<?php
$y = 2;
do{
    echo "The number is: $y <br>";
    $y++;
}
while($y <= 10)
?>
```

```
The number is: 2
The number is: 3
The number is: 4
The number is: 5
The number is: 6
The number is: 7
The number is: 8
The number is: 9
The number is: 10
```

ب) أعدّل على الكود السابق بأن يبدأ المتغيّر y بالقيمة التي تساوي 11، ثمّ أنفّذ النشاط، وأفسّر النتيجة.

- جملة التكرار forLoop (Syntax For loop):

تُستخدَم جملة For loop لتنفيذ أكواد وأوامر برمجيّة معيّنة لعدد محدّد من مرّات التكرار.

```
for(عملية زيادة او نقصان للمتغير ; شرط التكرار ; متغير رقمي){
  //يتكرر الكود ما دام شرط التكرار صحيحاً
}
```

نشاط (20) عمليّ:

أكتب الكود البرمجيّ الآتي باستخدام For loop لإظهار الأعداد الزوجيّة المحصورة بين الأعداد من (0) إلى (10)، ثمّ أنفّذه، وألاحظ النتيجة:

```
<?php
for ($z = 0; $z <= 10; $z+=2) {
    echo "The even number is: $z <br>";
}
?>
```

The even number is: 0
The even number is: 2
The even number is: 4
The even number is: 6
The even number is: 8
The even number is: 10

نشاط (21) عمليّ: أعدّل على الكود السابق لطباعة الأعداد الفرديّة.

الدوال PHP function:

مجموعة من الأوامر والأكواد البرمجيّة المعرّفة بصيغة معيّنة باستخدام كلمة Function، يمكن استدعاؤها داخل ملفّات المشروع البرمجيّ الواحد بأكثر من مكان. لا تنفّذ الدالّة Function عند فتح الملفّ بالمتصفّح مباشرة، بل عندما يتمّ استدعاء الدالّة داخل الملفّ.

- أنواع الدوال في PHP:

1- الدوال المضمّنة (Built-in Functions): دوال مبنيّة داخل لغة PHP ومدمجة بها، يمكن للمبرمج الاستفادة منها واستخدامها.

2- دوال معرّفة من المستخدم (User Defined Functions): هي الدوال التي ينشئها المبرمج ويستخدمها، ومنها:

أ- الدالّة الأساسيّة البسيطة.

ب- دالّة تستقبل قيمة أو أكثر.

ج- دالّة تعيد قيمة.

أوّلاً- الدوال الأساسيّة البسيطة:

تُعرف الدالّة من خلال الكلمة function، ثمّ اسم الدالّة، ثمّ علامتي الأقواس ( )، تليها الأقواس المتعرّجة {}، وداخل هذه الأقواس المتعرّجة تُكتَب الأوامر البرمجيّة المراد تنفيذها عند استدعاء الدالّة، كما يأتي:

```
function اسم الدالة()
{
    # الكود البرمجي
}
```

أناقش تتَّبع أسماء الدوال القواعد نفسها المتبّعة في تسمية المتغيّرات.

مثال (7):

أكتب الكود البرمجيّ الآتي في ملفّ PHP جديد:

```
<?php
function writeMsg() {
    echo "المعرفة أساس النجاح    ";
}
```

عرفت دالّة جديدة باسم writeMsg، تطبع رسالة جديدة، ولكن لا تعمل هذه الدالّة حتّى يتمّ استدعاؤها من خلال كتابة اسم الدالّة متبوعة بأقواس ( )، إمّا قبل الدالّة أو بعدها، كما في الكود الآتي:

```
<?php
function writeMsg() {
    echo "المعرفة أساس النجاح    ";
}

writeMsg(); // استدعاء الدالة writeMsg
?>
```

المعرفة أساس النجاح

نشاط (22) عمليّ: أُعدّل على المثال السابق باستدعاء الدالّة قبلها.

ثانياً- دالّة تستقبل قيمة أو اكثر:

تمرير المعلومات للدالّة Function ، وتُستخدَم بشكل مشابه للمتغيّرات، ويمكن تعريف أكثر من قيمة بكتابة أسمائها مفصولة عن بعضها بفاصلة.

نشاط (23) عمليّ:

أ- أكتب الكود البرمجيّ الآتي، وأُضيف ($firstname) لجعل الدالّة أكثر مرونة بحيث تطبع نصوص تحدّدها عند استدعاء الدالّة، وألاحظ النتيجة:

```php
<?php
username("User1");
username("User2");
username("User3");
function username($firstname)
{
   echo "hello $firstname ! <br>";
}
?>
```

```
hello User1 !
hello User2 !
hello User3 !
```

ب- أضيف مدخلاً جديداً باسم ($age) للكود البرمجيّ السابق كالآتي:

```php
<?php
username("User1",18);
username("User2",20);
username("User3",30);
function username($firstname,$age)
{
   echo "$firstname, $age year old <br>";
}
?>
```

```
User1, 18 year old
User2, 20 year old
User3, 30 year old
```

القيم الافتراضيّة للمدخلات Default Argument Value:

تصبح الدالّة أكثر مرونة عند كتابة قيم افتراضيّة للمتغيّرات داخل المدخلات عند تعريف دالّة ما، لتقليل الأخطاء عند استدعائها وتنفيذها.

مثال (8): أهميّة القيم الافتراضيّة:

1- أكتب الكود البرمجيّ الآتي، ثمّ أُنفّذه، وألاحظ نتيجة عدم كتابة إحدى القيم للمدخلات:

```php
<?php
function username($firstname,$age)
{
   echo "$firstname, $age year old <br>";
}
username("User1");
username("User2",20);
username("User3",30);
?>
```

الناتج: سيظهر خطأ على المتصفّح، يشير إلى مكان الخطأ في الكود، وتظهر باقي النتائج.

2- أضيف قيمة افتراضيّة للمدخل ($age) بالتعديل على الكود البرمجيّ:

```php
<?php
function username($firstname,$age = 22)
{
    echo "$firstname, $age year old <br>";
}
username("User1");
username("User2",20);
username("User3",30);
?>
```

User1, 22 year old
User2, 20 year old
User3, 30 year old

ثالثاً- دالّة تعيد قيمة أو أكثر:

الدوال المختلفة بلغة PHP ليس لطباعة النصوص فقط، بل يمكن استخدامها لإعادة قيم محدّدة، باستخدام return، وتُكتَب بطرق عدّة، منها:

1- إعادة قيمة دون التصريح بنوعها.

مثال (9): حساب مساحة مربّع، كما يأتي:

```php
function square($num)
{
    return $num * $num;
}
echo square(4);   //'16' الناتج

?>
```

- يمكن تعريف متغيّر جديد، وإسناد العمليّة الحسابيّة له ($z = $num * $num)، ثمّ إرجاع قيمة المتغيّر (return $z).

2- إعادة القيمة بالتصريح بنوعها.

مثال (10): التعديل على المثال السابق كالآتي:

```php
function square( $num):int
{
    $z = $num * $num;
    return $z;
}

echo square(4);   //'16' الناتج
```

- إضافة نوع القيمة بعد اسم الدالّة.

المصفوفة array:

تُعرّف المصفوفة بأنّها متغيّر من نوع خاصّ يمكنه الاحتفاظ بأكثر من قيمة في الوقت نفسه، تحتوي المصفوفة على العديد من القيم تحت اسم واحد، ويمكن الوصول إلى قيمة معيّنة فيها بحسب رقمها في الفهرس، أو مفتاح خاصّ بها، وتُعرَف باستخدام الدالّة array().

مثال (11): التعديل على المثال السابق كالآتي:

لو أردت إنشاء قائمة بأسماء ثلاثة طلّاب في مدرسة معيّنة، وتخزينهم على شكل متغيّرات كما يأتي:

```php
$student1 = "student1";
$student2 = "student2";
$student3 = "student3";
```

لكن ماذا لو أردت في المثال إدخال بيانات 100 طالب، فمن الصعب جدّاً إنشاء متغيّرات لعدد كبير من العناصر، من هنا أتت فكرة المصفوفة التي تحتوي عدداً كبيراً من المتغيّرات تحت مسمّى متغيّر واحد.

يمكن تشبيه المصفوفة بالحافلة، حيث تحتوي الحافلة على 40 مقعداً، في كلّ مقعد يجلس شخص معيّن. إذن: الحافلة تشبه المصفوفة، و40 مقعداً هم عناصر المصفوفة (حيث إنّ المقاعد تنتمي للحافلة)، والأشخاص الجالسون على المقاعد هم قيم هذه العناصر.

أنواع المصفوفات بلغة PHP:

1- المصفوفات البسيطة: مصفوفة بمفاتيح وفهرس عدديّ numeric index حيث إنّ كلّ مفتاح أو رقم فهرس يتكوّن من قيمة معيّنة، ويتمّ ترقيم هذه المفاتيح بشكل تلقائيّ، وتبدأ أوّل قيمة لها من الصفر.

الصيغة العامّة:

```
$اسم المصفوفة = array(1القيمة,2القيمة,.....);
```

نشاط (24) عمليّ: أنشئ مصفوفة تحتوي أسماء ثلاثة أنواع سيارات.

• الطريقة الأولى: تعيين الفهرس تلقائيّاً (لأنّ المفاتيح دائماً تبدأ بصفر) . أفتح الفيجوال ستوديو كود، وأنشئ ملفّاً جديداً وأسمّيه، ثمّ أكتب كما يأتي:

```
<?php
$cars = array("Volvo", "BMW", "Toyota");
?>
```

• الطريقة الثانية: تعيين قيمة للفهرس والمفاتيح بشكل يدويّ، كما يأتي:

```
<?php
$cars[0] = "Volvo";
$cars[1] = "BMW";
$cars[2] = "Toyota";
?>
```

طباعة نصّ يحتوي على قيم المصفوفة، كما يأتي:

```
<?php
$cars = array("Volvo", "BMW", "Toyota");
echo "I like " . $cars[0] . ", " . $cars[1] . " and " . $cars[2] . ".";
?>
```

I like Volvo, BMW and Toyota.

- دالّة العدّ The count () Function:

تُستخدَم هذه الدالّة لإرجاع ومعرفة الطول (عدد العناصر) داخل المصفوفة.

نشاط (25) عمليّ: أستخدم دالّة العدّ () count لمعرفة عدد عناصر المصفوفة $cars.

```
3
<?php
$cars = array("Volvo", "BMW", "Toyota");
echo count($cars);
?>
```

نشاط (26) عمليّ: أستخدم الحلقة For loop على عناصر المصفوفة $cars لطباعة عناصرها، كما في الكود الآتي:

```
Volvo
BMW
Toyota
<?php
$cars = array("Volvo", "BMW", "Toyota");
$arrlength = count($cars);

for($x = 0; $x < $arrlength; $x++) {
    echo $cars[$x];
    echo "<br>";
}
?>
```

2- **مصفوفات ذات المفاتيح:** تُستخدَم في هذه المصفوفة مفاتيح تُكتَب يدويّاً.

الصيغة العامّة:

```
$اسم المصفوفة = array (
    مفتاح(key) => القيمة,
    مفتاح(key) => القيمة,
    مفتاح(key) => القيمة,
)
```

- **خصائص مصفوفة ذات المفاتيح:**

1- المفتاح تكون قيمته عدداً صحيحاً، أو سلسلة نصّيّة، وإلّا يحوّل تلقائيّاً في الحالات الآتية:

- السلاسل النصّيّة التي تحتوي على أعداد صحيحة عشريّة صالحة، ما لم يُسبَق الرقم بعلامة + تحوّل إلى نوع عدد صحيح، على سبيل المثال المفتاح "8" يخزّن على شكل 8، ومن ناحية أخرى فإنّ "08" لن يحوّل لأنّه ليس عدداً صحيحاً عشريّاً صالحاً.
- تُحوّل الأعداد العشريّة (float) إلى أعداد صحيحة أيضاً، وهذا معناه أنّ الجزء الذي يأتي بعد الفاصلة سيُقتطع، فعلى سبيل المثال تُخزَّن القيمة التي لها المفتاح 8.7 في المفتاح 8.
- تُحوّل القيم المنطقيّة إلى أعداد صحيحة أيضاً، فعلى سبيل المثال، المفتاح true يخزّن 1، والمفتاح false يخزّن 0.
- القيمة Null تحوّل إلى سلسلة نصّيّة فارغة " ".
- لا يمكن استخدام المصفوفات والكائنات مفاتيح، وسيظهر تحذير عند استخدامها (warning: Illegal offset type).

2- إذا وُجدت عناصر في المصفوفة تمتلك المفتاح نفسه، يُستخدم المفتاح الأخير فقط وتُغيّر القيمة إلى القيمة الأخيرة.

3- القيم يمكن أن تكون من أيّ نوع (نصّ، رقم،... .).

مثال (12): إنشاء مصفوفة ذات المفاتيح لعلامات ثلاث موادّ:

أ- تعريف اسم المفتاح وتخزين قيمة فيه من خلال (=>).

```php
$mark = array (
    "English" => 85,
    "Arabic" => 90,
    "Math" => 95,
)
```

ب- لطباعة علامة مادّة اللغة العربيّة أضيف كالآتي:

```php
$mark = array (
    "English" => 85,
    "Arabic" => 90,
    "Math" => 95,
);
echo $mark['Arabic']; // 90 الناتج
```

ج- التعديل على علامة الرياضيات وطباعتها، كما يأتي:

```php
$mark = array (
    "English" => 85,
    "Arabic" => 90,
    "Math" => 95,
);
$mark['Math']= 80;
echo $mark['Math']; // 80 الناتج
```

د) إضافة علامة جديدة، ثمّ طباعة جميع عناصر المصفوفة باستخدام حلقة التكرار foreach loop:

```php
$mark = array (
    "English" => 85,
    "Arabic" => 90,
    "Math" => 95,
);
$mark['Web']= 93;// تضاف قيمة جديدة للمصفوفة
foreach($mark as $index => $value){
    echo $index." = ".$value."<br>";
}
```

الناتج:

English = 85
Arabic = 90
Math = 95
Web = 93

الأسئلة

1. ما مميّزات لغة PHP؟
2. ما أنواع البيانات المستخدمة بلغة PHP ؟
3. ما الفرق بين جمل الطباعة Print وEcho ؟
4. ما هي الدوال function؟ وما القاعدة العامّة أو البنية الأساسيّة لتعريف دالّة جديدة ؟
5. وضّح أنواع جمل التكرار، وما البنية الأساسيّة لكتابة كلّ نوع ؟
6. عرّف المصفوفة، واذكر أهمّيّتها وأنواعها.

الموقف التعليميّ التعلُّميّ (1-2):

توظيف (البرمجة كائنيّة التوجّه) في PHP

وصف الموقف التعليميّ:

أراد مدير شركة برمجيّات تعليم موظّفي التصميم (Front End) المراحل المتقدّمة في php، حتّى يساعدهم ذلك في إنجاز مهمّات التصميم بما يناسب المشروع بشكل أفضل.

العمل الكامل			
خطوات العمل	وصف الموقف الصفّيّ	المنهجيّة (استراتيجيّة التعلُّم)	الموارد
أجمع البيانات وأحلّلها	جمع البيانات من مصمّمي الصفحات: • المادّة التعليميّة (حجمها، موضوعها، طريقة عرضها، الأنشطة العمليّة والمادّة النظريّة التي يجب أن تشملها، وسائط الإيضاح التي تتضمّنها، (صور، روابط، نشرات، ...)). جمع البيانات عن: • مفهوم البرمجة الكائنيّة وأسسها. • خصائص البرمجة الكائنيّة: 1- الصنف / النوع (class) 2- الكائن (object) 3- الخصائص (المتغيّرات) المستخدمة. 4- الدوال (functions) 5- إمكانيّة الوصول. • خصائص البرمجة الكائنيّة: 1- الوراثة (inheritance). 2- التحميل الزائد (overloading). 3- إعادة الكتابة (overriding). 4- الواجهات والتجريد (interface & abstract). 5- التحزيم (packaging).	• الحوار والمناقشة. • التعلّم التعاونيّ (مجموعات العمل). • البحث العلميّ.	• وثائق (طلب مدير الشركة، مراجع عن البرمجة في PHP). • التكنولوجيا (فيديو، حاسوب، شاشة عرض، إنترنت).

أُخطّط وأُقرّر	أُحدّد كلّ من: • استخدامات، ومكوّنات البرمجة الكينونيّة بلغة PHP، ومفهومها. • المحاور الأساسيّة والمهمّة في البرمجة الكينونيّة، مثل: 1- الأصناف (classes). 2- الكائنات (objects). 3- الوراثة. 4- التغليف. • خطوات تحويل المشروع من html إلى php. • طرق كتابة الأكواد، وإضافة أنماط لغة PHP في صفحات الويب. • عقد جلسة مع مدير الشركة لمناقشة مقترحات للمادّة الأوّليّة التي تمّ إعدادها على الـ (CD) من حيث الشكل والمضمون. • إعداد خطّة توضّح الجدول الزمنيّ، ومراحل التنفيذ.	• التعلُّم التعاونيّ/ مجموعات العمل. • حوار ومناقشة. • عصف ذهنيّ استمطار الأفكار.	• وثائق (طلب المدير، نشرات ومصادر متنوّعة). • قرطاسيّة. • حاسوب. • الشبكة العنكبوتيّة.
أُنفّذ	• توضيح استخدامات، ومكوّنات البرمجة الكينونيّة بلغة PHP ومفهومها، والمحاور الأساسيّة، والمهمّة في البرمجة الكينونيّة، مثل: 1- الأصناف (classes). 2- الكائنات (objects). 3- الوراثة. 4- التغليف. • تبيان إجراءات وخطوات تحويل المشروع من html إلى php، وطرق كتابة الأكواد، وإضافة أنماط لغة PHP في صفحات الويب.	• التعلُّم التعاونيّ / مجموعات العمل. • حوار ومناقشة.	• قرطاسيّة. • الطلب الخاصّ بالمدير. • حاسوب، برامج السيرفر، ومحرّر الأكواد.

أتحقّق	التحقّق من: • استخدامات ومكوّنات البرمجة الكينونيّة بلغة PHP، ومفهومها. • المحاور الأساسيّة والمهمّة في البرمجة الكينونيّة، مثل: 1- الأصناف (classes). 2- الكائنات (objects). 3- الوراثة. 4- التغليف. • خطوات تحويل المشروع من html إلى php • طرق كتابة الأكواد، وإضافة أنماط لغة PHP في صفحات الويب.	• التعلُّم التعاونيّ / مجموعات العمل. • حوار ومناقشة.	• إنترنت. • الخطّة المعدّة سابقاً. • الطلب الخاص بالمدير. • حاسوب، برامج السيرفر، ومحرّر الأكواد.
أوثّق وأقدّم	- توثيق البيانات التي تتعلّق بـ: • بمفهوم البرمجة الكائنيّة وأُسسها. • خصائص البرمجة الكائنيّة: 1- الصنف / النوع (class). 2- الكائن (object). 3- الخصائص (المتغيّرات) المستخدمة. 4- الدوال (functions). 5- إمكانيّة الوصول • خصائص البرمجة الكائنيّة: 1- الوراثة (inheritance). 2- إعادة الكتابة (overriding). 3- التغليف (encapsulation). - إعداد العروض التقديميّة. - اعداد ملفّ بالحالة (طلب مدير الشركة).	• التعلُّم التعاونيّ / مجموعات العمل. • حوار ومناقشة.	• حاسوب. • شاشة عرض. • قرطاسيّة.
أقوّم	• نجاعة البرنامج التدريبيّ في تقوية المصمّمين، وقدرة استيعابهم ومقدرتهم على دمج بين التصميم والبرمجة. • رضا المدير عن البرنامج التدريبيّ.	• حوار ومناقشة.	• نماذج للتقويم (استبانة، ...). • طلب المدير.

سؤال: لماذا يُفضَّل استخدام سماحيّة الوصول (private) للخصائص (المتغيّرات) و(public) للدوال؟

أتعلّم

(البرمجة الكينونيّة (الكائنيّة) في PHP)

البرمجة الكينونيّة ((Object-oriented Programming (OOP)):

في نمط البرمجة الكينونيّة، يمكنك تقسيم مشكلة معقّدة إلى مجموعات أصغر عن طريق إنشاء كائنات وأصناف.

الكيان (Object):

هو نسخة من صنف (Class) معيّن تحتوي على دوال (Functions) وبيانات (Data) ...، وكلّ كيان (Object) له ميّزتان رئيسيّتان:

1- الحالة أو الوضع (state): هي حدث يخصّ الكائن، وينتج عن سلوك معيّن، وله خصائص معيّنة.
2- السلوكيّات (behavior): هي الدوال () الخاصّة بإعطاء أمرٍ ما للوصول إلى حالة، أو وضع محدّد.

مثال (1): افترض أنّ الإنارة كائن (object):

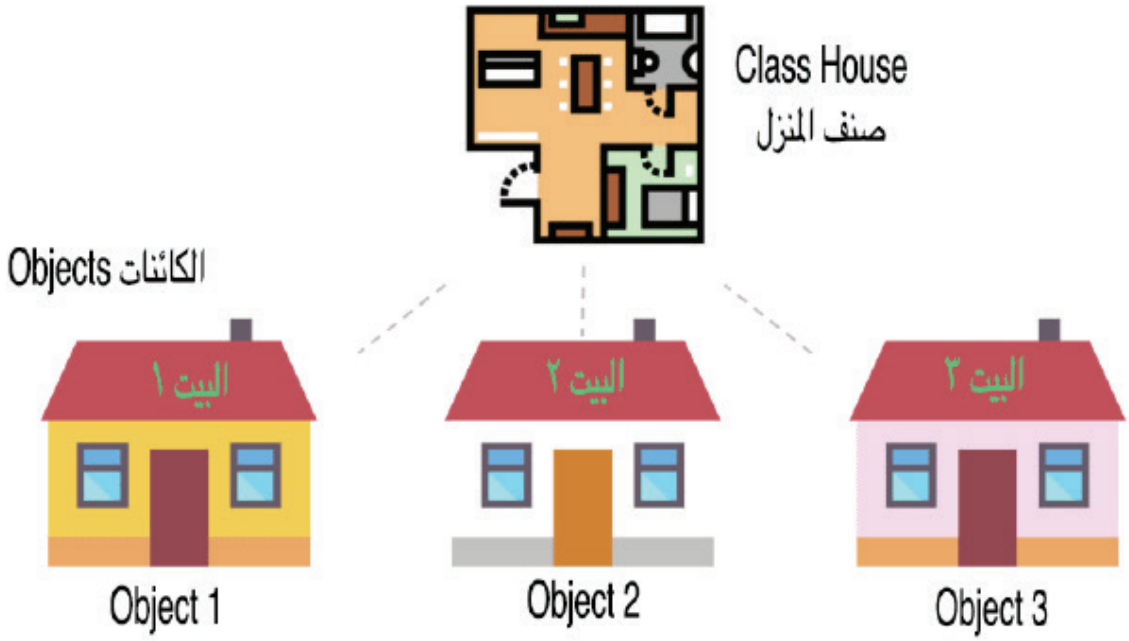
1- إمّا أن تكون الإنارة في وضع تشغيل أو إغلاق، وتُسمّى هذه حالة كائن الإنارة.
2- أستطيع تشغيل أو إغلاق الإنارة، وهذه تُسمّى سلوكيّات.

مثال (2): افترض أنّ سيّارة من نوع (س) كائن (object):

1- السيّارة تحتوي على أربعة إطارات، ومِقوَد،...الخ (خصائص)، إمّا أن تكون في وضع تشغيل وتسير، أو في وضع إيقاف ولا تعمل (الحدث)، هذه هي حالة كائن السيّارة.
2- أستطيع تشغيل السيّارة وقيادتها... إلخ، هذه تُسمّى سلوكيّات.

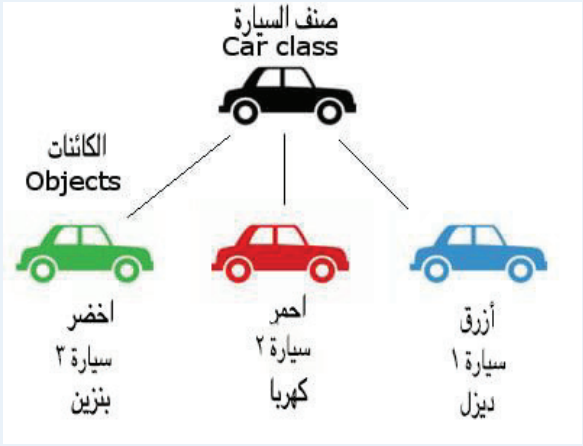
الصنف (Class):

هو القالب، أو البناء، أو الهيكل الأساسيّ الذي يتمّ إنشاء الكائنات (Objects) منه.
مثلاً: أستطيع عمل صنف قالباً أساسيّاً اسمه منزل، المنزل بشكل عام يتكوّن من: مساحة محدّدة، نوافذ، أبواب، ...، وبناءً على هذه الأمور تُبنى المنازل، كلّ بيت تمّ إنشاؤه يُسمّى كائناً (Object)، كما يأتي:



مثال (3):

أستطيع عمل صنف قالباً رئيسيّاً أسمّيه سيّارة، مكونات السيارة: اللون، النوع، ...، وبناءً عليها تُصنَع السيّارات، كلّ سيّارة تمّ تصنيعها تُسمّى كائناً، كالآتي:



مفهوم البرمجة الكينونيّة (Object-oriented):

هي نمط برمجة متقدّمة، يُستخدَم في تمثيل الأشياء في العالم الحقيقيّ عن طريق الصنف (Class) والكيان (Object)، حيث يقسم البرنامج إلى وحدات تُسمّى الكيانات (Objects)، وكلّ كيان عبارة عن حزمة من البيانات، والمتغيّرات، والثوابت، والدوال، ووحدات التنظيم، وواجهات الاستخدام.

نشاط (1) عمليّ: اكتب برنامج بسيط لتشغيل وإغلاق الإنارة باستخدام البرمجة الكينونيّة .

أشغّل السيرفر، ثمّ أنشئ ملفّاً جديداً، وأسمّيه على برنامج الفيجوال ستوديو كود، وأكتب كالآتي:

```php
<?php
 class Lamp{
    public $isOn = False;
    //الحالة الأولى الانارة
     function turnOne()
    {
      $this->$isOn = True;
    }
    //الحاله الثانية الاغلاق
    function turnOf()
    {
        $this->$isOn = False;
    }
    function displayLightStatus($lamp){
        if($this->$isOn == True){
            echo"</br> $lamp : الإنارة مضيئة";
        }else{
            echo"</br>$lamp : الإنارة مغلقة";
        }
    }
}
$lamp1 = new Lamp();//Lamp انشاء كائن من الصنف
$lamp2 = new Lamp();//Lamp انشاء كائن من الصنف

$lamp1->turnOf();
$lamp2->turnOne();

$lamp1->displayLightStatus("lamp1");
$lamp2->displayLightStatus("lamp2");
?>
```

الناتج:

lamp1 : الإنارة مغلقة

lamp2 : الإنارة مضيئة

التوضيح:

- أنشأت صنفاً (class) سميّته Lamp.
- عرفت متغيّراً isOn من نوع Boolean، ويُعدّ خاصيّة للصنف Lamp الذي سيوضّح حال الإنارة.
- أنشأت ثلاث دوال:
  1- turnOn() لتمثّل وضع تشغيل الإنارة، وأسندت قيمة True للمتغيّر isOn.
  2- turnOff() لتمثّل وضع إغلاق الإنارة ، وأسندت قيمة False للمتغيّر isOn.
  3- displayLightStatus($lamp) وتستقبل اسم الإنارة للتمييز بين الإنارات إذا كان هناك أكثر من إنارة.
- خارج الصنف أنشأت كائنين (objects) من الصنف Lamp اسميتهما l1، l2 (l1: تعبّر عن الإنارة 1، l2: تعبّر عن الإنارة 2)، وأستطيع إنشاء عدد لا نهائيّ من الكائنات من الصنف نفسه.
- عند إنشاء كائن من صنف معيّن أستطيع الوصول لجميع الدوال والمتغيّرات التي في داخل ذاك الصنف، ما دام لم يأخذ صفة private التي سأشرحها لاحقاً، هنا استدعيت الدالّة turnOn() من خلال الكائن l1 لتعطي الإنارة الأولى وضع التشغيل، واستدعيت turnOff() من خلال الكائن l2 لإغلاق الإنارة الثانية، ثمّ استدعيت الدالّة displayLightStatus() من خلال l1 و l2.
- للوصول إلى دالّة أو متغيّر داخل الصنف من خلال كائنٍ ما أستخدم (<-) بعد الصنف المنشأ.
- كلمة this تشير إلى الصنف الحاليّ الذي تقع فيه.

خاصيّة الوصول لعناصر الصنف (الخصائص (المتغيّرات) والدوال):

هي كلمات توضع قبل التعريف، تفيد في تحديد إمكانيّة الوصول للعنصر، مثل:

1- **public:** وهي السماحيّة الافتراضيّة للعناصر، وتعني أنّه يمكن الوصول للعنصر واستخدامه في أيّ مكان، سواء داخل الصنف، أو في أصناف الأبناء، أو حتّى أيّ مكان خارج الصنف.
2- **protected:** يمكن الوصول للعنصر واستخدامه في داخل الصنف، أو في أصناف الأبناء.
3- **private:** يمكن الوصول للعنصر واستخدامه في داخل الصنف فقط.

ملحوظة: للوصول لأيّة دالّة أو متغيّر (خاصيّة) في الصنف الرئيسيّ من خلال الكائن الذي تمّ إنشاؤه من ذلك الصنف، أكتب اسم الكائن ثمّ نقطة (<-)، كما في النشاط السابق، مثلاً lamp1->turnOn().

الباني (constructor):

يُعدّ الباني من أهمّ الدوال في الصنف، وهو دالّة خاصّة تسهّل عمليّة إنشاء الكائن، يتمّ استدعاؤها أثناء إنشاء الكائن لتوليد قيم أوّليّة للخصائص الموجودة فيه.

مثال (4):

```php
<?php
class Person{
    public $name;
    public $age;
    //دالة البناء
    function __construct($n,$a){
      //ربط متغيرات البناء مع متغيرات الصنف
      $this->name=$n;
      $this->age=$a;
    }

    //انشاء دوال تستخدم المتغيرات
    function get_name(){
        return $this->name;
    }
    function get_age(){
        return $this->age;
    }
}
// انشاء كائن من الصنف واضافة قيم
$person1 = new Person("Salem",18);
echo  $person1->get_name();
echo "</br>";
echo  $person1->get_age();
?>
```

1- عرفت متغيّرين، واحداً لتخزين قيمة String والآخر Int.

2- أنشأت بانياً (__construct دالّة جاهزة من الضروريّ أن تبدأ بشرطتين سفليّتين)، يستقبل قيمتين، وبداخله أسندت للمتغيّرات التي عرفتها في الخطوة الأولى القيمتين التي يستقبلها هذا الباني.

3- أنشأت دالتين لإرجاع قيمة كلّ متغيّر.

4- خارج الصنف أنشأت كائناً من الصنف Person، وأعطيته قيمتين، ثمّ كتبت جملة طباعة لطباعة قيم المتغيّرات الجديدة لـ (Name,Age) الموجودة في الصنف Person.

نشاط (2) عمليّ: أُعدّل على الكود الآتي لجعل البناء يستقبل قيمتين:

```php
<?php
class Fruit {
  public $name;
  public $color;

  function __construct($name) {
    $this->name = $name;
  }
  function get_name() {
    return $this->name;
  }
}

$apple = new Fruit("Apple");
echo $apple->get_name();
?>
```

الهدّام (Destructor):

يُستدعى destructor عندما ينتهي الكائن، أو يتمّ إتلافه، أو توقّف البرنامج النصّيّ، أو الخروج منه. إذا أنشأت دالّة (__destruct)، فإنّ PHP تنفّذها تلقائيّاً وتستدعيها في نهاية البرنامج النصّيّ.

مثال (5): أستخدم دالّة الهدم في المثال السابق:

```php
<?php
class Person{
    public $name;
    public $age;
    //دالة البناء
    function __construct($n,$a){
      //ربط متغيرات البناء مع متغيرات الصنف
      $this->name=$n;
      $this->age=$a;
    }
    //دالة الهدم
    function __destruct(){
        echo "finish";
```

```php
    }
    //انشاء دوال تستخدم المتغيرات
    function get_name(){
        return $this->name;
    }
    function get_age(){
        return $this->age;
    }
}
// انشاء كائن من الصنف واضافة قيم
$person1 = new Person("Salem",18);
echo  $person1->get_name();
echo "</br>";
echo  $person1->get_age();
echo "</br>";
?>
```

• عند تنفيذ الكود السابق تظهر نتيجة الهدّام بعد آخر موضوع في الصفحة، كالآتي:

Salem
18
finish

الوراثة (Inheritance):

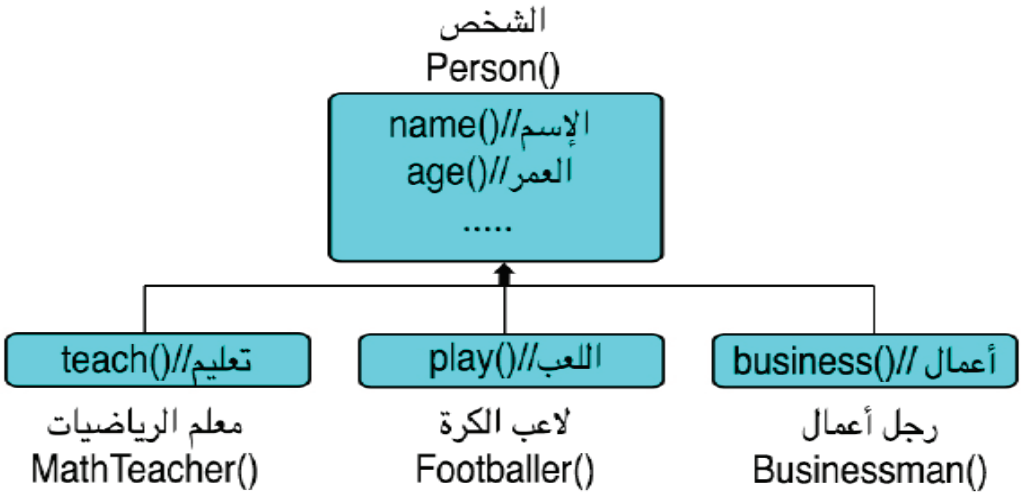
أفترض في مشروعي ثلاث شخصيّات: معلّم رياضيّات، لاعب كرة قدم، رجل أعمال، كلّ هؤلاء الأشخاص تجمعهم خصائص مشتركة، مثل: الأكل، الكلام، الحركة،...إلخ، وكلّ شخص عنده خاصيّة فرديّة، مثلاً: معلّم الرياضيّات يدرّس الرياضيّات، لاعب كرة القدم يلعب الكرة، رجل الأعمال عنده مشاريع استثماريّة.

أستطيع عمل لكلّ شخص صنف (class) خاصّ به، وأضع فيه أموراً عدّة، كما يأتي:

رجل أعمال Businessman()	لاعب الكرة Footballer()	معلم الرياضيات MathTeacher()

name()//الإسم
age()//العمر
.....
business()// أعمال
	name()//الإسم
	age()//العمر
	.....
	play()//اللعب
		name()//الإسم
		age()//العمر
		.....
		taechMath()//تعليم

ألاحظ أنّ هناك أموراً مشتركة، وإذا أردت إنشاء لكلّ شخص صنف خاصّ، سأكرّر في كلّ شخص الأكواد نفسها في كلّ صنف؛ لذلك حتّى أقلّل من التكرار أجمع الأمور المشتركة في صنف أسمّيه Person، وأجعل الأصناف الأخرى ترثه، فيصبح هو الصنف الأب وكلّ صنف يرثه يُسمّى الابن، كما يأتي:



- تعريف الوراثة (Inheritance):

هي إمكانيّة بناء صنف (class) على أساس صنف آخر أو أكثر، ويُسمّى الصنف المُورِّث الصنف الأب، والصنف الوارث يُسمى صنف الابن، يرث الصنف الابن جميع خصائص ودوال صنف الأب.

مثال (6): في هذا المثال أريد ترجمة ما كتبته سابقاً إلى كود برمجيّ، كما يأتي:

أوّلاً- أنشئ ملفّاً جديداً، وأسمّيه، مثلاً:(project.php) وأكتب كما يأتي:

```php
<?php
class Person{//الصنف الرئيسي
    private $name;
    private $age;
    function __construct($name,$age){
        $this->name=$name;
        $this->age=$age;
    }
   protected function get_info(){ // دالة لطباعه المعلومات لكن احدد سماحية الوصول لاستخدامها في الصنف الموروث
        echo "Name : {$this->name} , Age : {$this->age}</br>";
    }
}//انشاء ثلاث اصناف ترث الصنف الرئيسي
class Teacher extends Person{
    function teach(){
        echo "He teach Math in school</br>";
        $this->get_info();//استدعاء الدالة من الصنف الرئيسي
    }
}
class Businessman  extends Person{
    function business(){
        echo "He is a Businessman</br>";
        $this->get_info();
    }
}
class Footballer  extends Person{
    function football(){
        echo "He is a Footballer</br>";
        $this->get_info();
    }
}
$teacher = new Teacher("Hassan",55);//انشاء كائن من صنف
$teacher->teach();
$business = new Businessman("Jod",45);//انشاء كائن من صنف
$business->business();
$footballer = new Footballer("Mar",20);//انشاء كائن من صنف
$footballer->football();
?>
```

```
He teach Math in school
Name : Hassan , Age : 55
He is a Businessman
Name : Jod , Age : 45
He is a Footballer
Name : Mar , Age : 20
```

- أنشأت صنفاً رئيسيّاً يحتوي على بانٍ يستقبل قيمتين: الاسم والعمر، ودالّة تحتوي على جملة طباعة لهما.
- أنشأت ثلاثة أصناف ترث الصنف الرئيسيّ، وتحتوي كلّ منها على دالّة خاصّة بها، باستخدام كلمة extends.
- أنشأت كائنات من الأصناف الثلاثة، وبما أنّها ترث صنفاً يستقبل قيمتين أدخلها.

نشاط (3) (نظريّ): أناقش مع زملائي فائدة الوراثة من المثال السابق.

إعادة الاستخدام (Overriding):

في الوراثة عندما يرث صنف صنفاً آخر، أحتاج أحيانا لدالّة موجودة في صنف الأب، ولكن أريد استخدامها بطريقة مختلفة عن صنف الأب.

مثال (7):

إنشاء صنف للسيّارات يحتوي على دالّة بخصائص السيّارة، واستخدامها في صنف وريث بشكل مختلف كالآتي:

```php
<?php
class car{
    public $type;
    public $color;
    public $model;
    function __construct($type,$color,$model){
        $this->type=$type;
        $this->color=$color;
        $this->model = $model;
      }
    public function properties(){
        echo "Car type : {$this->type} , color : {$this->color} , model : {$this->$model} </br>";
    }
}
class Xcar extends car{
    private $number;
    function __construct($type,$color,$model,$num){
      $this->number = $num;
      $this->type = $type;
      $this->color = $color;
      $this->model = $model;
    }
    //اعادة استخدام الدالة ولكن بشكل مختلف
    public function properties(){
        echo "Car type : {$this->type} , color : {$this->$color} , model : {$this->model} , number : {$this->number} </br>";
        echo "يمكن اضافة و ازالة اي بيانات داخل الدالة";
    }
}
$xcar = new Xcar("c12","Black",2020, 123456);
$xcar->properties()
?>
```

Car type : c12 , color : , model : 2020 , number : 123456
يمكن اضافة و ازالة اي بيانات داخل الدالة

- عند إنشاء صنف الأب أنشأت بانياً من ثلاثة متغيّرات، ثمّ أنشأت دالّة لإعادة استخدامها لاحقاً.
- إنشاء صنف يرث الصنف الرئيسيّ، وإعادة استخدام الدالّة فيه، ولكن أضفت قيمة جديدة في الباني الخاصّ بهذا الصنف، واستخدمته في الدالّة.

عامل تحليل النطاق (::) في PHP:

يسمح عامل تحليل النطاق Scope Resolution Operator بالوصول إلى الخاصيّات والتوابع من نوع static، أو المتجاوز عليها، أو الثوابت الخاصّة بصنف معيّن.

مثال (8): استخدام الرمز (::) خارج الصنف:

```php
?>
<?php
class MyClass {
    const CONST_VALUE = 'قيمة ثابته';
}

$classname = 'MyClass';
// 5.3.0 من الإصدار
echo $classname::CONST_VALUE ."</br>";

echo MyClass::CONST_VALUE;
?>
```

الناتج:

قيمة ثابته
قيمة ثابته

مثال (9): استخدام الرمز (::) من داخل الصنف:

```php
<?php
class MyClass {
    const CONST_VALUE = 'قيمة ثابته';
}

$classname = 'MyClass';
// 5.3.0 من الإصدار
echo $classname::CONST_VALUE ."</br>";

echo MyClass::CONST_VALUE ."</br>";
?>
<?php
class OtherClass extends MyClass
{
    public static $my_static = 'static var';

    public static function doubleColon() {
        echo parent::CONST_VALUE . "</br>";//الاستدعاء داخل الصنف
        echo self::$my_static . "</br>";
    }
}

$classname = 'OtherClass';
// 5.3.0 منذ الإصدار
$classname::doubleColon();

OtherClass::doubleColon();
?>
```

دلالة كلمة (final keyboard):

تُستخدَم كلمة final في بداية الصنف لمنع وراثته، أو منع إعادة كتابة خصائص ودوال منه.

نشاط (4) عمليّ: اكتب الكود البرمجيّ الآتي ونفّذه ولاحظ النتيجة:

1- استخدم كلمة final لمنع الوراثة من الصنف.

```php
<?php
final class Fruit {
}

class Strawberry extends Fruit {
}

?>
```

2- استعمل كلمة final لمنع إعادة الكتابة.

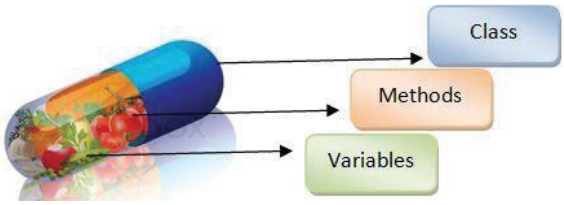
```php
<?php
class Fruit {
  final public function intro() {
  }
}

class Strawberry extends Fruit {
  //  النتيجة وجود خطأ
  public function intro() {
  }
}
?>
```

التغليف Encapsulation:

أسلوب يتمّ اتّباعه لإخفاء الصفات والبيانات الأساسيّة في الصنف (class)، وبالتالي تختفي الصفات العامّة(Global Variables) في الصنف وتفاصيل الكود. الأصناف الأخرى ممكن أن تنادي هذه الخصائص والصفات فقط من خلال دوال ينشئها المبرمج الأساسيّ للصنف.

- الاحتوائيّة أو التغليف: وُجدت لأنّ كلّ كائن يحتوي على البيانات والمعلومات عنه كاملة وطريقته الخاصة لتنفيذ أفعاله؛ لذا يجب حمايته وحماية ما بداخله.
- فكرة التغليف هي فكرة الكبسولة نفسها، تخفي ما بداخلها من دواء ومكوّنات؛ تلاحظ الشكل الخارجيّ للكبسولة فقط ، أمّا أسماء المكوّنات والموادّ التي بداخلها فلا تعرفها، فقط عندما تُستخدَم كمضاد حيويّ، أو علاج لمرض معيّن... وتكون الكبسولة آمنة لهذه الموادّ وتحميها من المؤثّرات الخارجيّة. كذلك التغليف يُستخدم لحماية البيانات وأمنها، فتستطيع إخفاء الكود الخاصّ بك لصنف معيّن عند استنساخ كائنات عنه، لن يتمكّن المبرمج الثاني من استعراض الكود للصنف الذي أنشأته، بل يستطيع رؤية أسماء الخصائص والأفعال فقط.



- باستخدام التغليف تتعامل مع البيانات وتلفّها كوحدة واحدة فقط، فيصبح الصنف للقراءة، أو الكتابة، وبالتالي توفّر السيطرة على البيانات.

مثال (10): توضيح مفهوم التغليف على شركة ما:

يوجد في الشركة أقسام عدّة: قسم الماليّة، قسم المبيعات،...إلخ. فقسم الماليّة يتولّى معالجة المعاملات الماليّة، ويحتفظ بسجلّات لتجميع البيانات المتعلّقة بالتمويل. وبالمثل قسم المبيعات يعالج البيانات المتعلّقة بالمبيعات، ويحتفظ بها في سجلّات خاصّة، ولكن عندما يحتاج مسؤول من قسم الماليّة إلى جميع البيانات المتعلّقة بالمبيعات في شهر معيّن، لا يستطيع الوصول إلى هذه البيانات مباشرة، بل عليه الاتّصال بالمسؤولين في قسم المبيعات، ثمّ يطلب منهم البيانات، وهذا هو التغليف.

التغليف في البرمجة الكينونيّة بلغة PHP (PHP OOP - Encapsulation):

يُستخدَم التغليف بلغة PHP من خلال استخدام الدالّة (private)، التي تخفي الخصائص وتحمي الوصول إليها من خارج الصنف، وللخصائص التي يمكن الوصول إليها من خارج الصنف تُستخدَم الدالّة (public). تبدأ اسم الدالّة التي تجلب القيمة وتحصل من خلالها على قيمة خاصّة بـ (get)، ثم اسم المتغيّر. الدالّة التي تغيّر القيمة وتحدّدها تبدأ بـ (set) يتلوها اسم المتغيّر؛ وذلك للسيطرة على قيمة الخاصيّة، والتأكّد من صلاحيّتها.

- setter: الرسائل التي تحدّد قيم الخصائص الخاصّة.
- getter: الحروف التي تحصل على قيم الممتلكات الخاصّة.

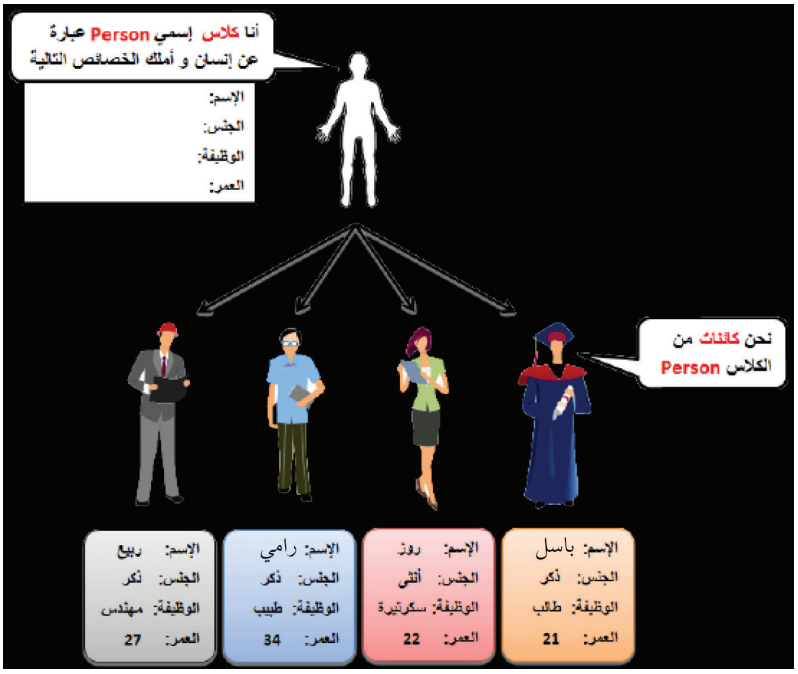
مثال (11): توضيح مفهوم التغليف على شركة ما:

أنشئ صنفاً باسم الطالب، وعرّف فيه المتغيّرات: الاسم الأوّل ونوعه، كما يأتي:

```php
<?php
class Student {
    private $firstname;
    private $gender;

    public function getFirstName() {
        return $this->firstname;
    }
    public function setFirstName($firstname) {
        $this->firstname = $firstname;
        echo("First name is set to ".$firstname);
        echo("<br>");
    }
    public function getGender() {
        return $this->gender; }
    public function setGender($gender) {
        $this->gender = $gender;
        echo("Gender is set to ".$gender);
        echo("<br>");
    }
}
$student = new Student();
$student->setFirstName('Ahmad');
$student->setGender('Male'); ?>
```

نشاط (5) عمليّ: أنشئ صنفاً باسم person وأضِف إليه الخصائص والأفعال التي في الصورة، وعرّفها بطريقة التغليف:



ترتيب ملفّات مشروع php:

عند عمل مشروع كامل وبعد الانتهاء من مرحلة التصميم، لا بدّ من تحويل المشروع إلى php، وإدخال ما يلزم من الأكواد بلغة php، ويمكن عمل تنظيم وتقليل من أكواد التصميم المشتركة، واستدعاؤها باستخدام include، يجب أن تكون الصفحة الرئيسيّة مسمّاة index.php، وإلّا لن يعمل المشروع؛ لأنّ السيرفر يبدأ القراءة التلقائيّة من هذا الملفّ.

- أهمّ خطوات تحويل وترتيب المشروع:

1- تحويل امتداد جميع صفحات html إلى (.php) مثلاً: (index.html) إلى (index.php).
2- إذا كان header و footer في المشروع مشتركاً في كلّ الصفحات من الأفضل فصل كلّ منها في ملفّ مستقلّ، ثمّ استدعاؤها في الصفحات.
3- إذا وُجدت أكواد جافاسربت ذات أهمّيّة عاليه تحويلها إلى كود php لزيادة أمان الموقع.

نشاط (6) عمليّ: إنشاء مشروع php، كما يأتي:

أوّلاً- إنشاء مجلّد جديد باسم المشروع، مثلاً(test)، كالآتي:

ثانياً- إنشاء ملفّ أسمّيه index.php، وأكتب فيه كالآتي:

```
<!DOCTYPE html>
<html dir="rtl" lang="ar">
    <head>
        <meta charset="utf-8"/>
        <title></title>
    </head>
    <body>
1-      <?php
        include "include/header.php";
        ?>
        <hr>
        <main id="about">
            <article>
                <h4> </h4>
                <section>

                </section>
            </article>
        </main>
        <hr>
        <main id="msg">
                <article>
                    <h4></h4>
                    <section>

                    </section>
                </article>
        </main>
```

```
29            <hr>
30            <main id="p">
31                <article>
32                    <h4></h4>
33                    <section>
34
35                    </section>
36                </article>
37            </main>
38            <hr>
39  2-        <?php
40            include "include/footer.php";
41            ?>
42        </body>
43    </html>
44
```

استدعاء ملفَّي header وfooter اللذين سأنشئهما في الخطوات الآتية بواسطة include، ووضع مسار الملفّين.

ثالثاً- عمل مجلّد جديد باسم include، وإنشاء ملفّين باسم (header.php) و (footer.php) كما يأتي:

رابعاً- أكتب في الملفّين الآتي، أو ما يناسب الصفحة المنشأة:

```
<header>
            <mark><b> </b></mark>
            <nav>
           <ul>
               <li><a href="#about"><strong> </strong></a></li>
               <li><a href="#msg"><strong> </strong></a></li>
               <li><a href="#p"><strong> </strong></a></li>
           </ul>
           </nav>
</header>
```

```
<footer>
        <span>
         <small>حقوق النشر محفوظة @ مصدر المعلومات
         </small>
        </span>
</footer>
```

خامساً- اختبار المشروع.

نشاط (7) عمليّ: أُحوّل النشاط (14) من كتاب الصف الحادي عشر الوحدة الرابعة إلى ملفّ php، وأعمل الترتيبات اللازمة.

الأسئلة

1. أعرّف ما يأتي: الصنف ، الكيان أو الكائن، وأوضّح الفرق بينهما.
2. أعدّد أهمّ خطوات تحويل وترتيب المشروع في PHP.

أسئلة الوحدة:

السؤال الأوّل: أضع دائرة حول رمز الإجابة الصحيحة فيما يأتي:

1- لماذا تُستخدم كلمة final؟

أ- لمنع وراثة أيّ صنف، أو منع إعادة كتابة خصائص ودوال منه.

ب- لإضافة قيمة ثابته.

ج- لتوريث صنف لآخر.

د- لا شيء ممّا ذكر.

2- أيّ من الآتية لا يُعدّ أداة لتعلّم php؟

أ- المتصفّح. ب- محرّر الأكواد. ج- برنامج الفوتوشوب. د- حزمة تطبيقات تتضمّن خادم الويب.

3- ما هي جمل الطباعة المستخدمة في php؟

أ- echo ب- ptintln ج- print د- أ + ج.

4- ما الدالّة التي تُستخدم لاستدعاء الملفّات في php؟

أ- require ب- include ج- echo د- لا شيء ممّا ذكر.

5- أيّ من الآتية لا يعدّ من جمل التكرار في php؟

أ- while ب- for ج- foreach د- switch

6- أيّ من خصائص الوصول الآتية لا تسمح باستخدام الدالّة أو المتغيّر إلّا في الصنف نفسه؟

أ- public ب- private ج- protected د- لا شيء ممّا ذكر.

7- ما الذي يُستدعى عندما ينتهي الكائن، أو يتمّ إتلافه، أو توقّف البرنامج النصّيّ، أو الخروج منه؟

أ- الوراثة. ب- البناء. ج- الهدّام. د- التغليف.

السؤال الثاني: ما مميّزات لغة PHP؟

السؤال الثالث: أوضّح الصيغة العامّة لكتابه php.

السؤال الرابع: ما شروط تعريف المتغيّرات في php؟

السؤال الخامس: أُصنّف أسماء المتغيّرات الآتية إلى اسم متغيّر مسموح، أو غير مسموح تعريفه، مع ذكر السبب:

اسم المتغيّر (Variable name)	مسموح أو غير مسموح تعريفه	السبب
1first_student		
final exam		
employee@com		
Year19		
Ebook#		

السؤال السابع: أوضّح الفرق بين get وpost في إرسال البيانات.

السؤال الثامن: ما المقصود بالبرمجة الكينونيّة؟

دراسة الحالة:

توجّه مصمّم صفحات ويب إلى مطوّر ويب لتحليل نظام مدرسة، وتحويله إلى كود php، ثمّ دمج التصميم مع الأكواد.

مشروع الوحدة: أنزل قالب تصميم جاهز من الإنترنت، ثمّ:

أ- أحوّله إلى ملفّات php.

ب- أرتّب الملفّات، وأنسّق الكود، وأفصل الأكواد المتكرّرة.

ج- أنشئ صفحة تسجيل، وأستخدم get و post.

نموذج التقويم الذاتيّ:

المهارات	مستوى الأداء		
	1	2	3
أساسيّات البرمجة في لغة php			
البرمجة الكينونيّة oop			

التقدير:

(3) أداء المَهمّة بشكل متقن.

(2) أداء المهمّة بشكل جيّد.

(1) محاولة أداء المهمّة.

الوحدة الثّانية

قواعد البيانات

أناقش

تلعب قواعد البيانات دوراً مهمّاً في حياتنا.

يُتوقَّعُ من الطلبةِ بعد دراسةِ هذه الوَحْدةِ، والتّفاعُلِ معَ أنشطتِها أن يكونوا قادرين على التعامل مع قواعد البيانات، وربطها في مواقع الإنترنت من خلال الآتي:

- المفاهيم الأساسيّة في قواعد البيانات.
- إنشاء وإدارة قواعد البيانات.
- إنشاء موقع تفاعليّ.

الكفايات المهنيّة

الكفايات المهنيّة المتوقَّع من الطلبة امتلاكها بعد انتهاء دراسة هذه الوحدة، والتفاعل مع أنشطتها:

أوّلاً- الكفايات الحرفيّة:

القدرة على:

1- توضيح المفاهيم الأساسيّة في قواعد البيانات.
2- إنشاء وإدارة قواعد البيانات.
3- إنشاء موقع تفاعليّ.

ثانياً- الكفاياتُ الاجتماعيّةُ والشخصيّةُ:

1- العمل ضمن فريق.
2- تقبُّل التغذية الراجعة.
3- احترام رأي الآخرين.
4- مصداقيّة التّعامُل مع الزبون.
5- حِفْظُ خصوصيّة الزبون.
6- الاستعداد للاستجابة لطلبات الزبون.
7- التحلّي بأخلاقيّات المهنة أثناء العمل.
8- التمتُّع بالفكر الريادي أثناء العمل.
9- استخلاص النتائج ودقّة الملاحظة.
10- الاتّصال والتواصل الفعّال.
11- الاستعداد للاستفسار والاستفادة من ذوي الخبرة.
12- الإفادة من مقترحات الآخرين.
13- امتلاك مهارة التأمُّل الذاتيّ.
14- الاستعداد التامّ لتقبُّل ملحوظات الزبون وانتقاداته.
15- القدرة على الإقناع.

ثالثاً- الكفاياتُ المنهجيّة:

1- توظيف التعَلُّم التعاونيّ.
2- القدرة على إدارة الحوار وتنظيم النقاش.
3- امتلاك مهارة البحث العلميّ، والقدرة على توظيف أساليبه.
4- القدرة على التخطيط.
5- القدرة على استمطار الأفكار.
6- استخدام المصادر والمراجع المختلفة.
7- توثيق نتائج العمل وعرضها.
8- تقويم النتائج وعرضها.
9- جمع البيانات وتحليلها وتبويبها.

قواعد الأمن والسلامة:

1- الالتزام بالجلسة الصحيّة أمام الحاسوب.
2- إراحة العينين كلّ ربع ساعة.
3- الجلوس في غرفة فيها إضاءة مناسبة وتهوية جيّدة.
4- الابتعاد عن الحاسوب كلّ ساعة، أو القيام ببعض التمارين الرياضيّة.

الموقف التعليميّ التعلُّميّ (2-1):

مفاهيم أساسيّة في قواعد البيانات

وصف الموقف التعليميّ:

يرغب طلبة إحدى المدارس الأكاديميّة في إعداد مطويّة عن مفهوم قواعد البيانات وعناصرها المختلفة، فاستشاروا فنّي حاسوب للاستفسار عن مضمون المطويّة.

العمل الكامل			
خطوات العمل	وصف الموقف الصفّيّ	المنهجيّة (استراتيجيّة التعلُّم)	الموارد
أجمع البيانات وأحلّلها	أجمع البيانات من الطلبة عن: • معرفتهم السابقة بمفهوم قواعد البيانات العلائقيّة، وعناصره المختلفة. • محتوى المطويّة. • مواصفات المطويّة (عدد الصفحات، الصور، روابط تعليميّة،...). أجمع البيانات حول: • مفهوم قواعد البيانات العلائقيّة. • عناصر قواعد البيانات العلائقيّة (الجداول، الحقول، السجلّات، المفاتيح، العلاقات). • بيئة العمل (phpMyAdmin)).	• الحوار والمناقشة. • التعلُّم التعاونيّ (مجموعات العمل). • البحث العلميّ.	• وثائق: الطلب الخاصّ بطلبة المدرسة، نشرات، مقالات، كتب عن قواعد البيانات العلائقيّة. • التكنولوجيا: الإنترنت، مقاطع فيديو، مواقع إنترنت موثوقة عن قواعد البيانات.
أخطّط وأقرّر	-أحدّد : • مفهوم قاعدة البيانات العلائقيّة. • مفهوم الجداول. • عناصر الجدول (الحقول، السجلّات). • مفهوم المفاتيح وأنواعها (المفتاح الرئيسيّ، المفتاح الثانويّ، المفتاح الأجنبيّ). • مفهوم العلاقات وأنواعها (علاقة واحد إلى واحد، علاقة واحد إلى متعدّد، علاقة متعدّد إلى متعدّد).	• تعلُّم تعاونيّ/ مجموعات. • حوار ومناقشة.	• وثائق (طلب المدرسة الزائرة، مكتبات، مجلّات تتحدّث عن قواعد البيانات).

	• المقصود ببيئة العمل (phpMyAdmin). • العناصر الأساسيّة في الصفحة الرئيسيّة لِـ phpMyAdmin • إعداد خطّة زمنيّة للتنفيذ.		
أنفّذ	• صياغة مفهوم قواعد البيانات العلائقيّة، ومفهوم الجدول. • إنشاء جدول للمقارنة بين عناصر الجدول (الحقول والسجلّات). • صياغة مفهوم المفاتيح وأنواعها. • إجراء مقارنة بين أنواع المفاتيح المختلفة. • كتابة مفهوم العلاقات. • إجراء مقارنة بين أنواع العلاقات المختلفة. • تعريف المقصود ببيئة العمل (php MyAdmin). • التعرف على العناصر الأساسيّة في الصفحة الرئيسية لِـ phpMyAdmin. • وضع مقترحات عدّة للمطويّة. • عقد جلسة مع طلبة المدرسة الزائرة، والاتّفاق على مضمون المطويّة وشكلها. • إجراء التعديلات المطلوبة وإخراج المطويّة بحلّتها النهائيّة.	• تعلّم تعاونيّ/ مجموعات. • حوار ومناقشة. • عصف ذهنيّ/ استمطار الأفكار.	• إنترنت حاسوب. • البيانات التي تمّ جمعُها. • قرطاسيّة. • الطلب الخاصّ بالزائرين.
أتحقّق	أتحقّق من: • صياغة مفهوم قواعد البيانات العلائقيّة، ومفهوم الجدول، ومفهوم المفاتيح وأنواعها، ومفهوم العلاقات. • دقّة جدول المقارنة بين عناصر الجدول (الحقول والسجلّات)، وجدول المقارنة بين أنواع المفاتيح المختلفة، وجدول المقارنة بين أنواع العلاقات المختلفة. • توضيح المقصود ببيئة العمل (php MyAdmin). • العناصر الأساسيّة في الصفحة الرئيسيّة لـ phpMyAdmin.	• تعلم تعاونيّ / مجموعات. • حوار ومناقشة.	• البيانات التي تمّ جمعُها، عن قواعد البيانات • إنترنت، حاسوب. • الخطّة المعدّة سابقاً.

أوثّق وأقدّم	أوثّق: • مفهوم قواعد البيانات العلائقيّة، ومفهوم الجدول، ومفهوم المفاتيح وأنواعها، ومفهوم العلاقات. • الفرق بين عناصر الجدول (الحقول والسجلّات)، وجدول المقارنة بين أنواع المفاتيح المختلفة، وبين أنواع العلاقات المختلفة. • المقصود ببيئة العمل (php MyAdmin). • العناصر الأساسيّة في الصفحة الرئيسيّة لِـ php-MyAdmin. • إعداد ملفّ بالحالة (طلب المدرسة). • إعداد عرض تقديميّ. • إعداد ملخّص يتضمّن مفهوم قواعد البيانات العلائقيّة وعناصره، ودور كلّ عنصر في قاعدة البيانات، والمقصود ببيئة العمل php MyAdmin ، والعناصر الأساسيّة في الصفحة الرئيسيّة التابعة لها.	• حوار ومناقشة. • تعلّم تعاونيّ/ مجموعات عمل.	• البيانات التي تمّ جمعُها سابقاً عن قواعد البيانات. • حاسوب. • شاشة عرض. • قرطاسيّة.
أقوّم	• رضا طلبة المدرسة عن المطويّة التي تمّ إعدادها. • ملاءمة المطويّة من حيث الشكل والمضمون للمواصفات والمعايير.	• حوار ومناقشة.	• مجلّات تتحدّث عن المواصفات الخاصّة بمطويّة قواعد البيانات. • طلب المدرسة الزائرة.

أناقش يجب أن يحتوي كلّ جدول على مفتاح رئيسيّ.

أتعلّم

مفاهيم أساسيّة في قواعد البيانات

نشاط (1) للاطّلاع: أبحثُ عن استخدامات قواعد البيانات في الحياة اليوميّة.

تتواجد قواعد البيانات في كثير من الأعمال الحياتيّة، ويتمّ التعامل معها بشكل يوميّ، فهي موجودة في تطبيقات الجوال، والحواسيب، وكثير من مواقع الإنترنت، وتُستخدَم في أنظمة البنوك، والجامعات، والأسواق، والمطارات، وغيرها... .

قواعد البيانات Database:

مجموعة من البيانات المنظّمة والمرتبطة بموضوع معيّن داخل جدول أو مجموعة من الجداول؛ بهدف استرجاعها لاتّخاذ القرارات المطلوبة.

مناهج قواعد البيانات:

1- المنهج التقليديّ:

منهج الملفّات والمستندات الورقيّة، كانت تُستخدَم الأوراق لتسجيل الأعمال، والحركات الماليّة والأنشطة وغيرها، لحفظ البيانات والرجوع إليها عند الحاجة.

(عيوبه):

- يقتصر على الأنظمة الصغيرة.
- زيادة الكلفة التشغيليّة.
- قلّة الكفاءة والفاعليّة.
- يستهلك وقتاً كبيراً لاسترجاع البيانات.

2- المنهج الحديث:

مع تطوّر التكنولوجيا، أصبحت عمليّة التعامل مع البيانات أسهل، حيث تمّ تطوير برامج خاصّة تتعامل مع البيانات، ولهذه المنهجيّة مزايا متعدّدة، منها:

- وجود تعريف واضح للبيانات.
- تخزين كميّات كبيرة من البيانات.
- إضافة ملفّات جديدة، والتعديل عليها، أو حذفها، أو تحديثها.
- سهولة الحصول على البيانات واسترجاعها .
- عرض البيانات بشكل منظّم.
- ضمان عدم تكرار البيانات.
- إمكانيّة التشغيل بنظام متعدّد المستخدمين.

من الأمثلة على قواعد البيانات ما يأتي:

- القبول والتسجيل الجامعيّ في المعاهد والجامعات.
- تسجيل ومحاسبة المرضى في المستشفيات العامّة أو الخاصّة.
- المبيعات والمشتريات في الشركات التجاريّة والمصانع وغيرها.
- إدارة المخازن والمستودعات في مؤسّسة ما.
- التأمين بأنواعه: (التأمين الصحّيّ، وتأمين السيّارات، وتأمين البضائع).
- إدارة خدمات البلديّة: (مياه، كهرباء، صرف صحّيّ، ...إلخ) .
- إدارة المكتبات العامّة والخاصّة: (إعارة الكتب، بيع وشراء الكتب).
- العمليّات المصرفيّة في المصارف والبنوك ...إلخ.

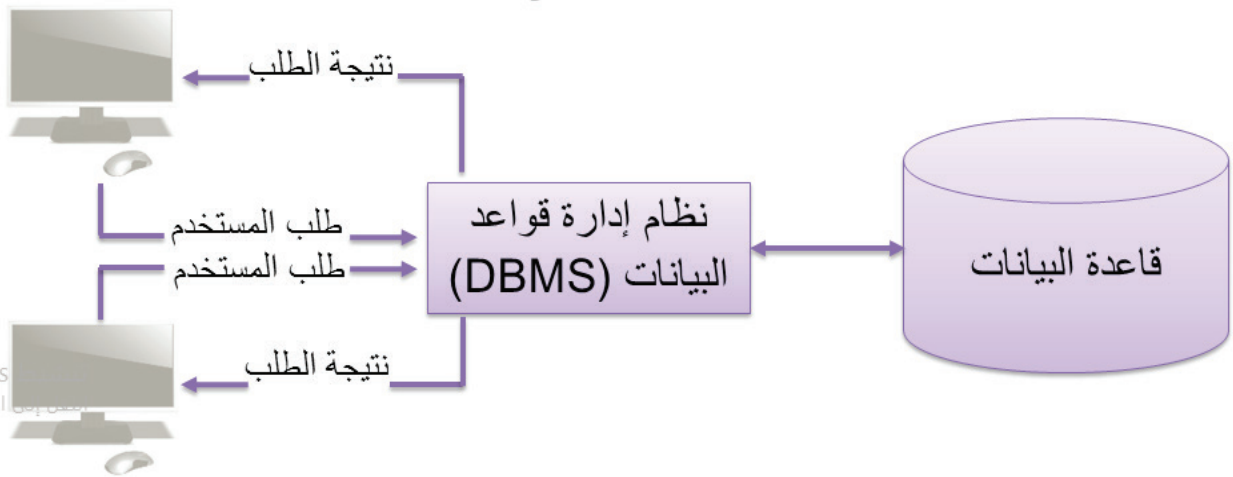
نشاط (2) للاطّلاع: أبحث عن أمثلة إضافيّة لقواعد البيانات، وأناقشها مع زملائي.

نُظم إدارة قواعد البيانات DBMS (Database Management System):

مجموعة من البرامج التي تُمكّن المستخدم من التعامل مع قاعدة البيانات، من حيث القدرة على تعريفها، وإنشائها، وتمثيلها، وتعديلها.

يتمّ التعامل مع البيانات من خلال تعريفها، ووضع القيود عليها، وحذفها، وفرزها، وتخزينها، واسترجاعها، بطرق سهلة.

والصورة الآتية توضّح نُظم إدارة قواعد البيانات:



من أشهر البرامج التي تُمكّن من إدارة قواعد البيانات:

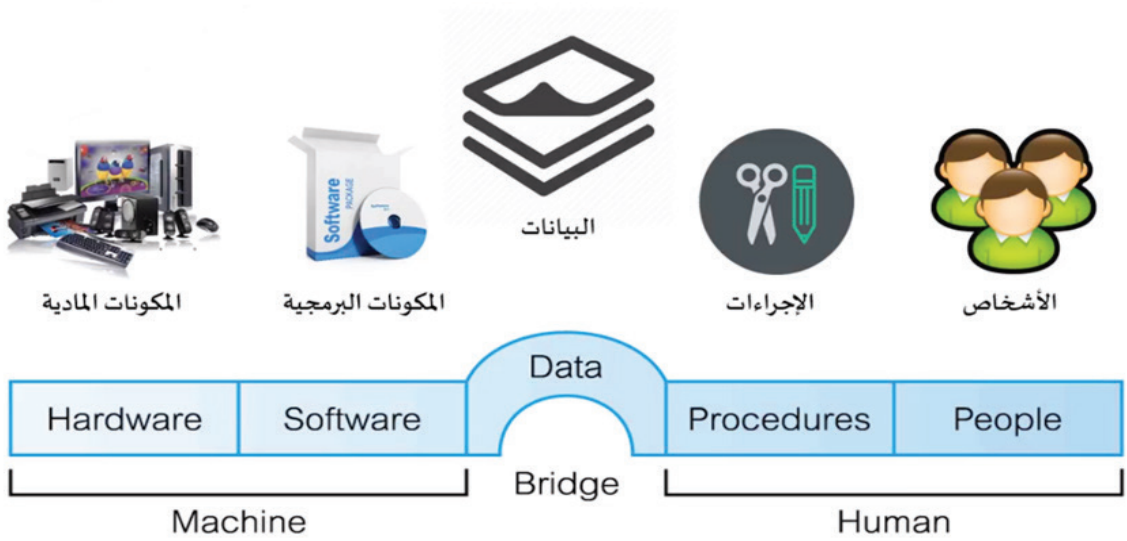
1- أوراكل Oracle.

2- MySQL.

3- SQL Server.

نشاط (3): أبحثُ عن مميّزات كلّ من البرامج السابقة لإدارة قواعد البيانات، وأضعها في جدول.

مكوّنات قاعدة البيانات:



1- **المكوّنات الماديّة:** تشمل جميع الأجهزة الماديّة، مثل: الحواسيب، والأجهزة الطرفيّة، والطابعات.

2- **المكوّنات البرمجيّة:** تنقسم البرامج المستخدمة في قواعد البيانات إلى:

- أنظمة التشغيل: البرامج التي تقوم بإدارة الأجهزة، مثل : Windows, Unix, Linux... .

- برنامج قاعدة البيانات: وهي تتولّى إدارة قواعد البيانات، مثل : Oracle, MySQL... .
- البرامج التطبيقيّة والمساعدة: البرامج التي تقوم بعمليّات الاسترجاع والتخزين وغيرها.

3- **البيانات:** تُعدّ المادّة الخام التي تشمل مجموعة من الحقائق المخزّنة في قاعدة البيانات، (أسماء، أرقام ...).

4- **الأشخاص:**

- مدير قواعد البيانات (Database Admin)

مهمّاته:

- تحديد متطلّبات قواعد البيانات المطلوبة من برمجيّات وتجهيزات.
- التنسيق الدائم في عمليّات استخدام قواعد البيانات .
- توفير الأمن والحماية لقواعد البيانات، وصلاحيّات تفويض الاستخدام .
- الرقابة وضبط أداء النظام ضمن مقياس عمل مثاليّ.

- مصمّم قواعد البيانات (Database Designer)

مهمّاته:

- تحديد البيانات الواجب تخزينها.
- تحديد أفضل التراكيب، وبناء البيانات الواجب استخدامها.
- تصميم قاعدة البيانات، والوصول إلى أقلّ درجة ممكنة من الأخطاء وهدر المصادر.
- تحديد طرق تخاطب المستخدمين مع قاعدة البيانات، ويشمل تعريف وتصميم شاشات التخاطب وتوثيقها.

- مبرمج قواعد البيانات (Database Programmer)

مهمّاته:

- تحويل وترجمة تصاميم قواعد البيانات إلى لغات قواعد بيانات جاهزة لإدخالها إلى الحاسوب.
- تنفيذ تلك الأنظمة والبرمجيّات، والتأكّد من صحّتها، وخلوّها من الأخطاء.
- إنشاء شاشات التخاطب، والإدخال، والإخراج التي تحتاجها نظم قواعد البيانات، وتنفيذها.
- صياغة أنماط وأشكال التقارير المطلوبة، وتنفيذها.

- مستخدمو قواعد البيانات (End User)

وهم تلك المجموعة التي تستخدم أنظمة قواعد البيانات وتطبيقها في مجال محدّد، وقد يكون المستخدمون على مستويات عدّة، اعتماداً على معرفتهم وخبرتهم في الحاسوب على وجه العموم، وبنُظم قواعد البيانات على وجه الخصوص، ويشمل:

- المستخدم البدائيّ.
- المستخدم الخبير.

5- **الإجراءات والتعليمات:**

القوانين والتعليمات التي تحكم عمل قواعد البيانات بشكل صحيح، وتخزّن التعليمات موثّقة بشكل واضح محدّد.

أنواع أنظمة قواعد البيانات:

كانت قواعد البيانات المتعارف عليها في الماضي، هي:

1- قواعد البيانات الشبكيّة.

2- قواعد البيانات الهرميّة.

وظلت هذه الأنواع هي المستخدمة حتى ظهرت قواعد البيانات العلائقيّة.

تتلخّص فكرة قاعدة البيانات العلائقيّة بمجموعة من الجداول (Tables) والعلاقات (Relations)؛ ونظراً لقوّتها، وسهولة تصميمها، وبرمجتها، وتعامل المستخدمين معها فقد طغت على الأنواع الأخرى، وأصبحت النوع الوحيد المستخدم.

نشاط (4): أبحثُ عن المقصود بالاختصار RDBMS.

قواعد البيانات العلائقيّة:

مجموعة من الجداول تربط بينها علاقات، وكلّ جدول يشتمل على صفوف (سجلّات)، وأعمدة (حقول).

يتكوّن الجدول من:

1- أعمدة (Columns): تُستخدَم لإظهار صفات أو خصائص البيانات، وتُسمّى الحقول.

2- صفوف (Rows): تمثّل سجلّات، وكلّ سجلّ يحتوي على بيانات مطابقة لأعمدة الجدول.

مثال (1):

الجدولان التاليان هما جزء من قاعدة بيانات، تربط بين الجدولين علاقات، تُستخدم للاستعلام عن معلومات داخل الجداول واسترجاعها:

الجدول 1:

المواطن	رقم الهويّة	الجنس	العمر	الحالة الاجتماعيّة
محسن	123456789	ذكر	50	متزوّج
حنان	112233445	أنثى	45	متزوّج
خالد	556677889	ذكر	20	أعزب

الجدول 2:

رقم الهويّة	عدد أفراد الأسرة
123456789	3
112233445	2
556677889	1

من المثال شخص رقم هُويّته (112233445)، للاستعلام عن اسمه، وعدد أفراد أسرته أتّبع الخطوات الآتية:

أوّلاً: من الجدول 1 أبحثُ عن رقم الهويّة المطلوب، ومنه أستخرجُ اسم الشخص(حنان).

ثانياً: من الجدول 2 أبحثُ باستخدام رقم الهويّة، وأستخرج عدد أفراد الأسرة (2).

تصميم قاعدة البيانات:

وَصْفٌ مختصر لمتطلبات المستخدم، بواسطة رسم مخطّط المفاهيم، وهو نموذج رسوميّ يصف كيانات النظام، وعلاقتها مع بعضها.

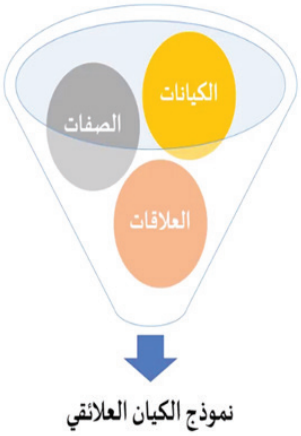
نموذج الكيان العلائقيّ Entity-Relationship Diagram (ERD):

يتكوّن من:

1- الكيانات Entities:

هي الوَحدات الأساسيّة في قاعدة البيانات، وهي تشير إلى شيء حقيقيّ في الحياة، سواء كان له وجود فعليّ، مثل: (مريض - طالب - موظّف - سيارة ...)، أو وجود منطقيّ، مثل: (شركة - وظيفة - مقرّر ...).

يُمثَّل الكيان باستخدام شكل مستطيل يكتب داخله اسم الكيان.



نموذج الكيان العلائقي

2- الصفات أو الخصائص Attributes:

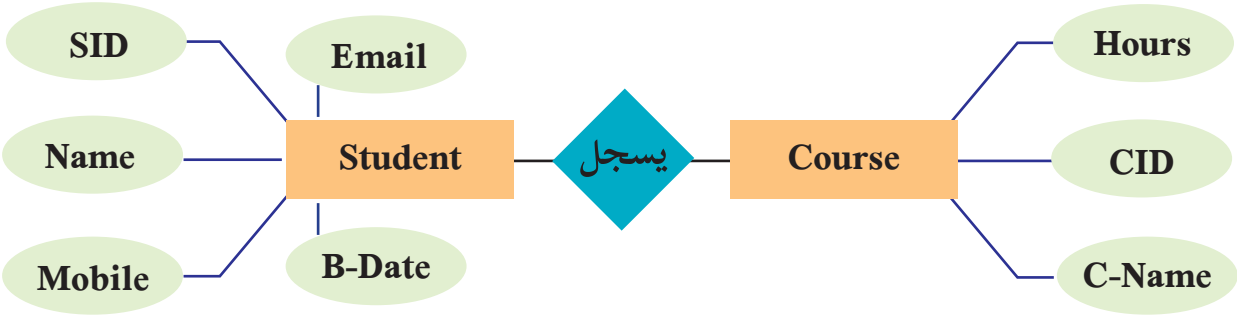
هي صفات معيّنة تصف الكيان أو العلاقة، وتكون تابعه له، ولا تخصّ غيره، مثل: (اسم الطالب، رقم الطالب، عنوان الطالب) تصف الكيان الطالب.

وتُمثّل الصفات بحسب أنواعها بأشكال بيضاويّة مختلفة.

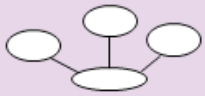
3- العلاقات Relationship:

هي العلاقة التي تربط بين الكيانات في الواقع، وهي عبارة عن فعل يمثّل العلاقة بين كيان ونفسه، أو كيانين، أو ثلاثة كيانات معاً، وتُمثَّل بشكل معيّن.

نشاط (4): أستخرجُ مكوّنات النموذج العلائقيّ من الشكل الآتي:



الرموز المستخدمة في النموذج العلائقيّ:

المفهوم	الرمز
الكيان	
الصفة البسيطة و الأحادية	
الصفة الفريدة أو المفتاح أساسي	
صفة متعددة القيم	
صفة مركبة	
صفة مشتقة	
العلاقة	

أنواع الصفات:

- صفة بسيطة: مثل اسم الطالب، رقم الطالب.
- صفة مركّبة: عند تقسيم الصفة إلى أكثر من قسم: مثل اسم الطالب يقسم إلى: (الاسم الأوّل، اسم الأب، اسم العائلة).
- صفة أحاديّة القيمة: مثل تاريخ ميلاد الطالب.
- صفة متعدّدة القيم: مثل رقم الهاتف يمكن أن يكون للطالب أكثر من رقم هاتف .
- صفة مشتقّة: وهي التي يمكن الحصول عليها عن طريق صفة بسيطة أخرى، مثل عُمْر الطالب، نحصل عليه من تاريخ ميلاده.
- صفة فريدة وتُسمّى المفتاح الأساسيّ: وهي التي لا يمكن أن تتكرّر، وتميّز كلّ سجلّ عن الآخر، ويوضع خطّ تحتها عند الرسم.

مفاتيح الجداول:

تُعدّ المفاتيح من أهمّ خصائص قواعد البيانات العلائقيّة، فهي تميّز جدول معيّن من جهة، والرابط الذي يربط الجداول المختلفة مع بعضها من جهة أخرى، وتنقسم إلى ثلاثة أنواع:

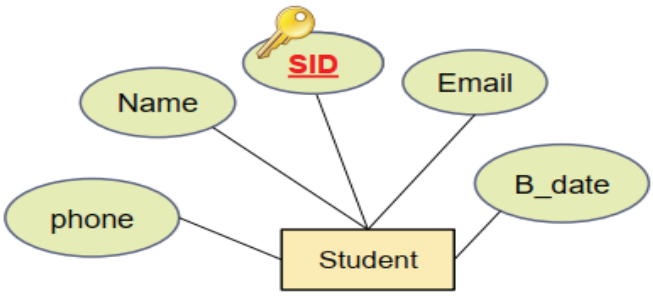
- **المفتاح الأساسيّ PK (Primary Key):**

حقل فريد Unique (غير متكرّر القيمة) يميّز بين سجلّات الجدول، ولا يقبل القيمة الفارغة (NULL). وعند رسم مخطّط ERD يوضع خطّ تحت اسم الخاصيّة التي تعبر عن المفتاح الأساسيّ.

أشكال المفتاح الأساسيّ:

1- بسيط : يتكوّن من عمود واحدة فقط .

2- متفرّع : يتكوّن من مجموعة من الأعمدة.



- **المفتاح الثانويّ (البديل) (Secondary Key):**

حقل تتكرّر فيه المعلومات، ويُستخدم لاسترجاع البيانات، مثلاً مواطن له سجلّ في دائرة الزراعة، وأراد الاستعلام عن المساعدات المقدّمة له، ولم يكن يعرف رقم هويّته (رقم الهويّة هو المفتاح الأساسيّ)، يستخرج البيانات باستخدام الاسم.

- **المفتاح الأجنبيّ FK (Foreign Key):**

حقل يشير إلى مفتاح رئيسيّ، أو قيمة غير مكرّرة (Unique)، تمّ نقل نسخة منه إلى جدول آخر من أجل عمليّات الربط، فأصبح أجنبيّاً في ذلك الجدول.

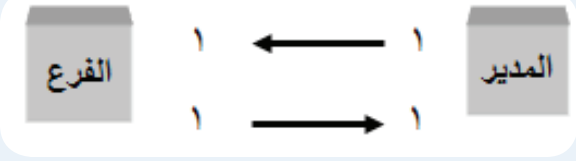
أنواع العلاقات بين الجداول (Type of Entities Relationship):

أوّلاً- علاقة واحد إلى واحد One to One:

كلّ سجلّ في الكيان الأوّل يرتبط بسجلّ واحد فقط في الكيان الثاني وكلّ سجلّ في الكيان الثاني يرتبط بسجلّ واحد فقط في الكيان الأوّل. يُرمز لها بـ 1:1

مثال (2):

البنك لديه فروع عدّة ، كلّ فرع يرأسه مدير واحد فقط، وكلّ مدير يرأس فرعاً واحداً فقط . فتكون العلاقة بين كيان المدير وكيان الفرع علاقة واحد إلى واحد.

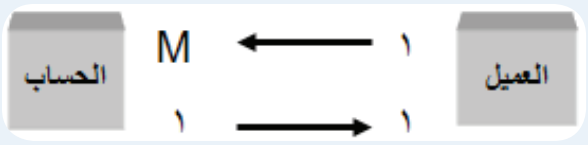


ثانياً - علاقة واحد إلى متعدّد One to Many:

كلّ سجلّ في الكيان الأوّل يرتبط بسجلّات عدّة في الكيان الثاني، وكلّ سجلّ في الكيان الثاني يرتبط بسجلّ واحد فقط في الكيان الأوّل. يُرمز لها بـ : M:1

مثال (3):

في البنك يمكن أن يكون للعميل أكثر من حساب بنكيّ، ولكن يوجد لكلّ حساب بنكيّ عميل واحد فقط.

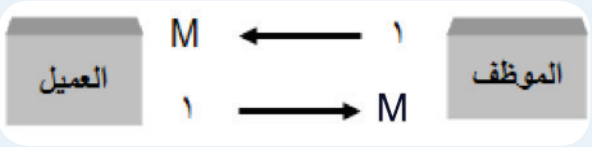


ثالثاً - علاقة متعدّد إلى متعدّد Many to Many:

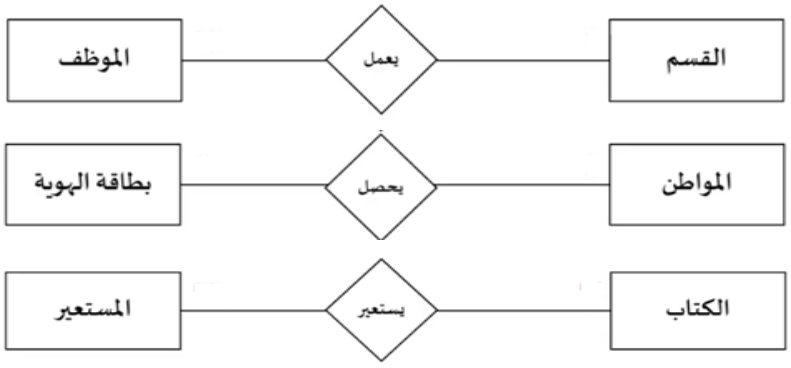
كلّ سجلّ في الكيان الأوّل مرتبط بسجلّات عدّة في الكيان الثاني، وكلّ سجلّ في الكيان الثاني مرتبط بسجلّات عدّة في الكيان الأوّل. يُرمز لها بـ M:M

مثال (4):

في البنك يخدم الموظّف أكثر من عميل، والعميل يمكن أن يخدمه أكثر من موظّف.



نشاط (5) نظريّ: أحدّد نوع العلاقة في كلّ من الأمثلة الآتية، وأعطي أمثلة إضافيّة أخرى:



PhpMyAdmin:

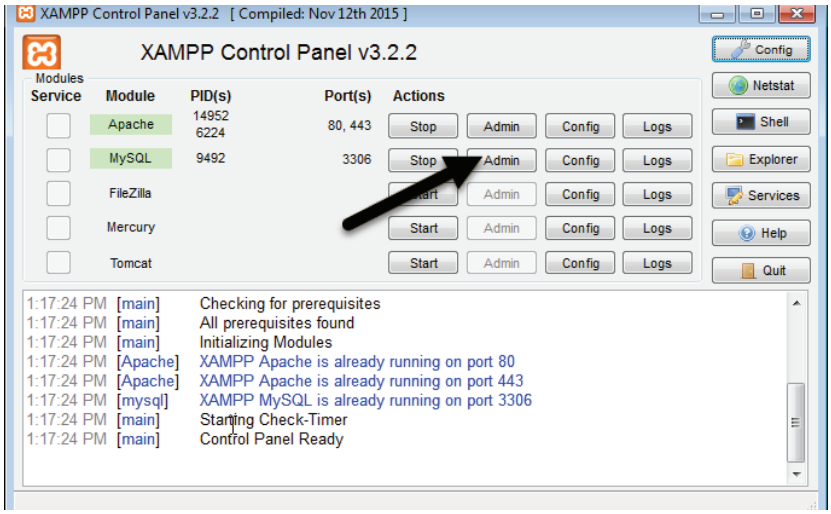
يوجد طُرق عدّة للتعامل مع قواعد البيانات، منها:

1- سطر الأوامر: طريقة متقدّمة، تُستخدم غالباً في السيرفرات التي لا تتوفّر عليها واجهة رسوميّة.

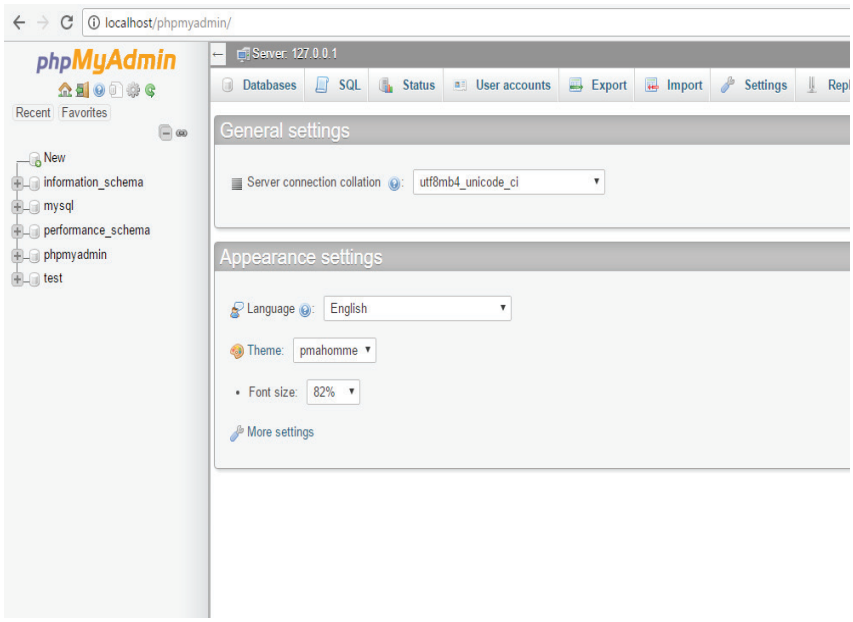
2- phpMyAdmin: واجهة رسوميّة، تمكّن مطوّري مواقع الإنترنت من التعامل مع قواعد بيانات MySql بسهولة.

الواجهة الرسوميّة phpMyAdmin:

1- يتمّ فتح XAMPP وتشغيل Apache وMySQL، ثمّ اختيار Admin كما يأتي:



2- ظهور الواجهة كما في الصورة الآتية:



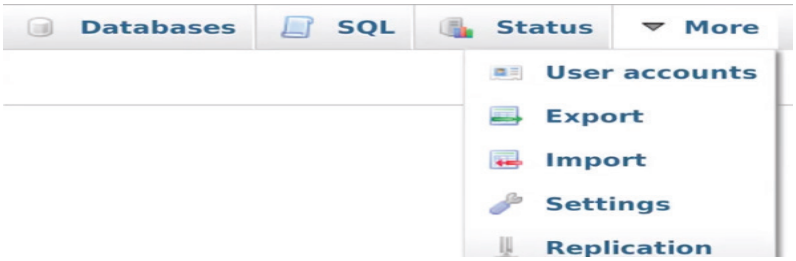
نشاط (6) للاطّلاع: أستنتجُ طرقاً أخرى للوصول إلى صفحة phpMyAdmin.

مكوّنات الصفحة الرئيسيّة لِـ phpMyAdmin:

1- عنوان السيرفر المحليّ يكون موجوداً في أعلى الصفحة، إمّا localhost، أو على شكل أرقام ip (127.0.0.1) كما يأتي:

2- القائمة العلويّة:

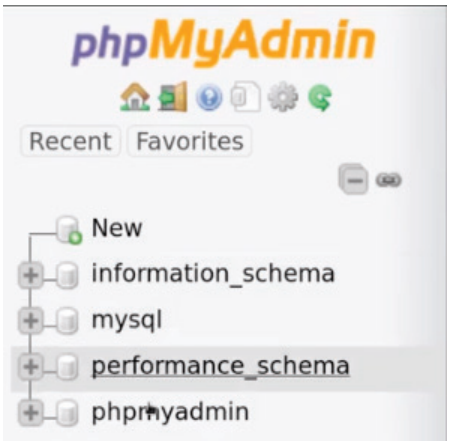
تحتوي القائمة العلويّة على خيارات عدّة، تساعد على إجراء العمليّات المختلفة في قواعد البيانات، كالآتي:



- Databases (قواعد البيانات): تحتوي على قواعد البيانات المنشأة على السيرفر المحليّ، وإنشاء قاعدة بيانات جديدة.
- SQL: كتابة الأوامر النصّيّة للتعامل مع قواعد البيانات.
- الحالة (Status): تعرض حالة السيرفر والعمليّات والإحصائيّات وغيرها، ولا تستخدم في إنشاء المواقع الإلكترونيّة.
- User accounts (حسابات المستخدمين): عرض مستخدمي السيرفر المتاح لهم الوصول إلى قواعد البيانات.
- Export (التصدير): استخراج قاعدة بيانات منشأة على السيرفر وتخزينها على جهاز الحاسوب.
- Import (استيراد): جلب قاعدة بيانات تمّ تصديرها، أو إنشاؤها سابقاً على جهاز الحاسوب.
- Settings (الإعدادات): التعديل في إعدادات PhpMyAdmin.
- Operations (العمليّات): إجراء عمليّات على قاعدة البيانات والجداول، مثل: النسخ، والحذف، وإعادة تسمية، وغيرها.

3- القائمة الجانبيّة:

تضمّ القائمة الجانبيّة على قواعد البيانات الموجودة على السيرفر وتديرها phpMyAdmin:



نشاط (7) عمليّ: أبحثُ عن كيفيّة تشغيل Apache Server و MySQL تلقائيّاً عند بداية تشغيل جهاز الحاسوب.

إنشاء قاعدة بيانات MySQL:

لا بدّ من إنشاء قاعدة بيانات لإدراج الجداول، وحفظ المعلومات داخلها.

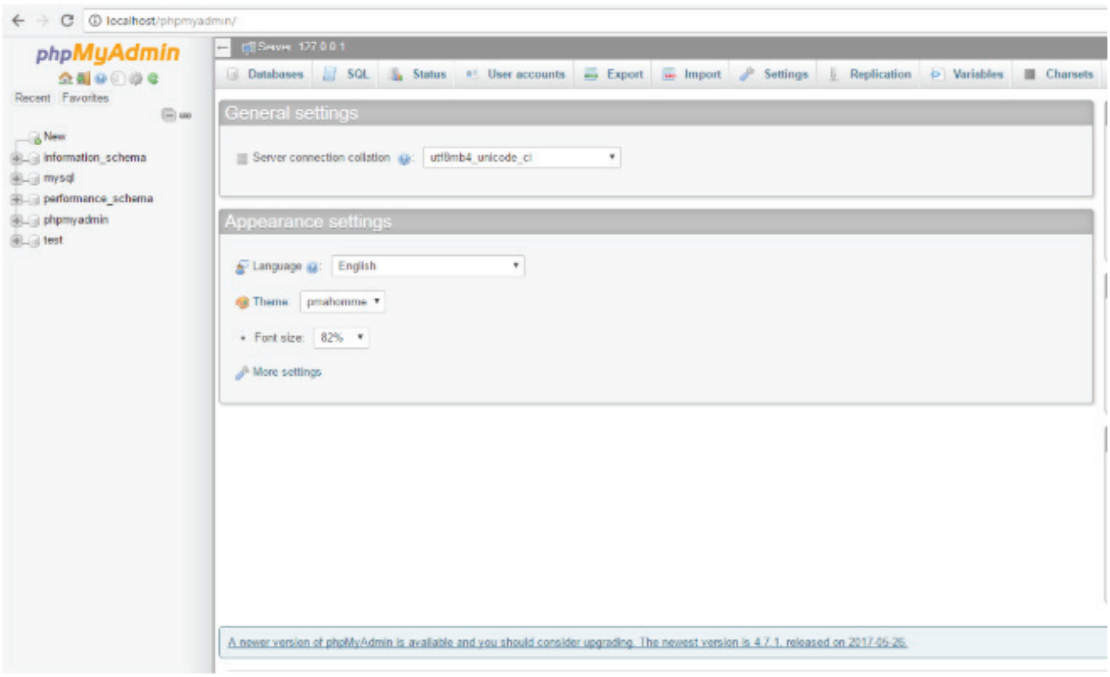
شروط تسمية قاعدة البيانات والجداول والأعمدة:

1- أن يكون الاسم متّصلاً، ويبدأ بحرف.
2- يُفضَّل أن لا يتجاوز طول اسم الجدول أو العمود 30 حرفاً Characters، فإنّ بعض نظم إدارة قواعد البيانات تمنع تجاوز هذا العدد من الأحرف، مثل الأوراكل (oracle).
3- أن يكون الاسم غير مستخدم مسبقاً (يُمنع تكرار الأسماء في المكان نفسه).
4- يجب أن لا يكون من الكلمات المحجوزة.
5- وضع فاصلة منقوطة في نهاية الجملة (;).
6- يفصل بين أسماء الحقول فاصلة (,).

خطوات إنشاء قاعدة بيانات جديدة:

أوّلاً- باستخدام الواجهات:

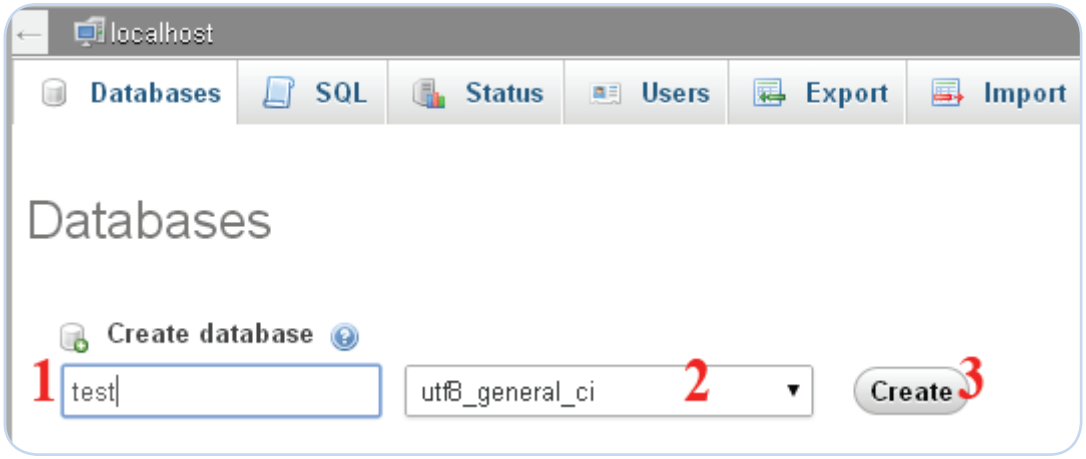
1- تفعيل كلّ من Apache وMySQL، ثمّ الدخول إلى صفحة phpMyAdmin، كالآتي:



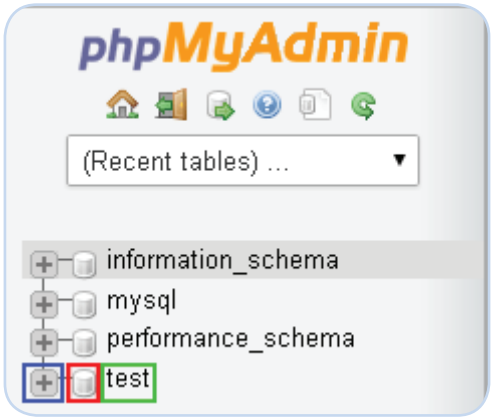
2- اختيار databases (قاعدة البيانات) من القائمة العلويّة.
3- تسمية قاعدة البيانات test.

4- أحدّد الترميز utf8_general_ci كما في الصورة أدناه.

utf8_general_ci: هو ترميز موحّد، يمكّننا من تسجيل بياناتنا بجميع اللغات بعد هذا الاختيار.

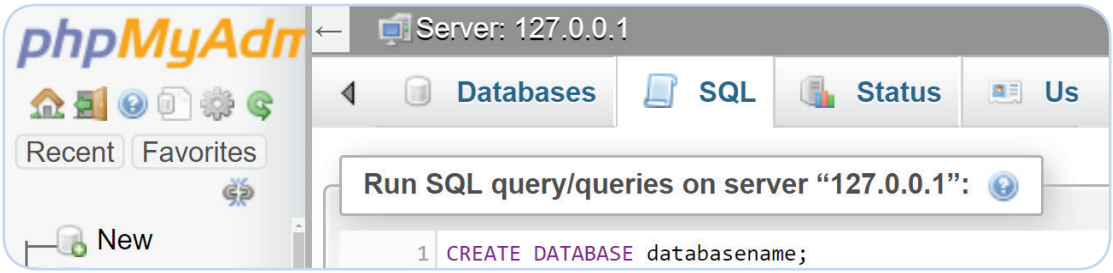


5- أنقر على زر create، سيتمّ إنشاء قاعدة البيانات، وسيظهر اسمها في القائمة العموديّة يسار أو يمين صفحتكم (بحسب اللغة التي اعتمدتها). كما توضّح الصورة أدناه:



ثانياً- طريقة الأوامر النصّيّة:

- أذهب إلى وصلة SQL في القائمة العلويّة.



- أستخدم التركيب الآتي:

```
CREATE DATABASE databasename;
```

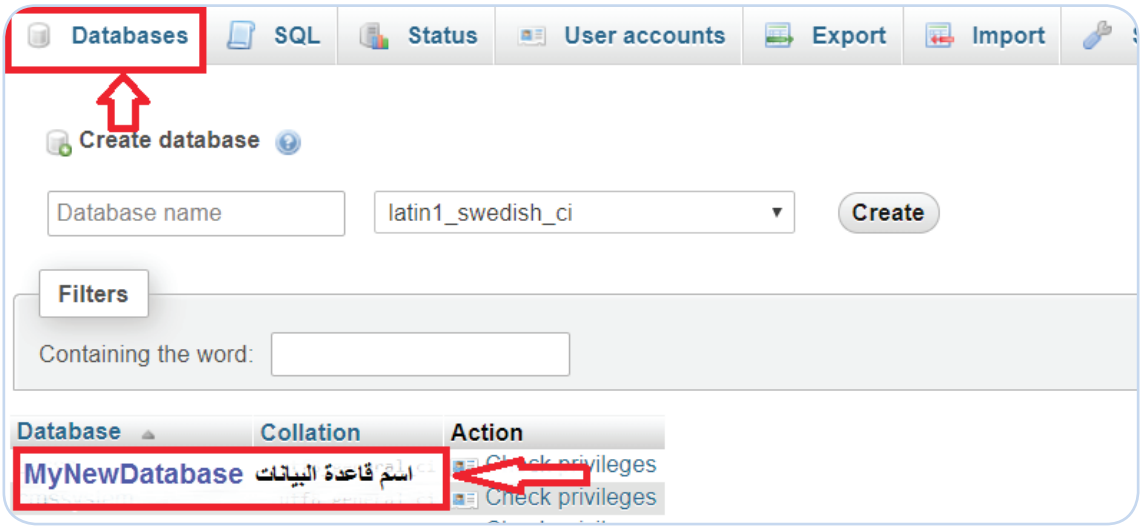
Databasename: اسم قاعدة البيانات

تعيين صلاحيّات المستخدمين في PHPMyAdmin:

حتى تُستخدَم قاعدة البيانات، يجب منح صلاحيّات الوصول إليها لمستخدمي هذه القاعدة، كما يأتي:

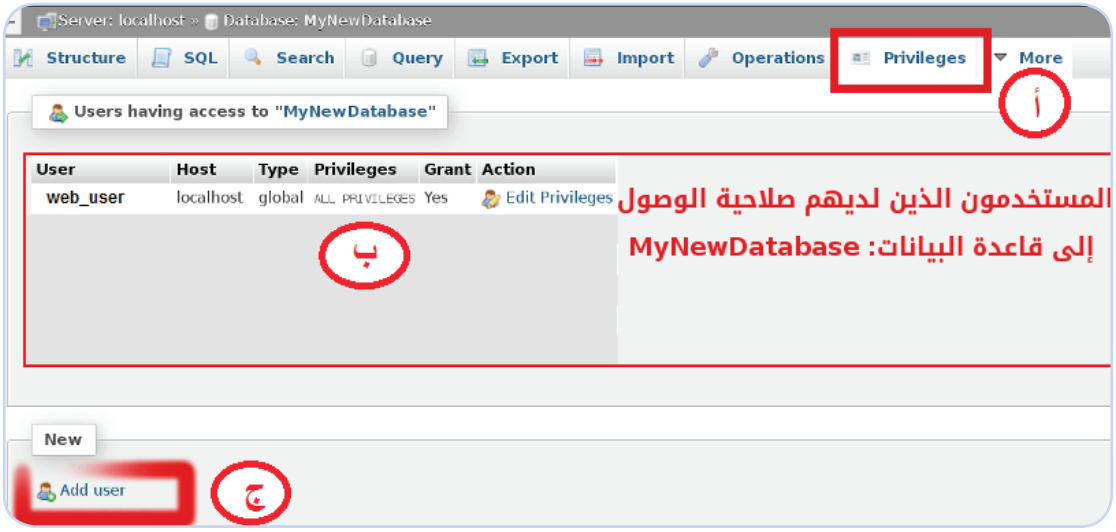
أوّلاً- تحديد قاعدة البيانات:

- النقر على تبويب قواعد البيانات | Databases في الأعلى.
- النقر على اسم قاعدة البيانات المُراد تعيين صلاحيّات الوصول إليها.



ثانياً- إضافة مستخدم جديد:

- أنقرُ على تبويب Privileges | الامتيازات والصلاحيّات في ترويسة الصفحة.
- يظهر المستخدمون الذين لديهم صلاحيّة الوصول إلى قاعدة البيانات MyNewDatabase .
- لإنشاء مستخدم جديد، أنقرُ على Add Userإضافة مستخدم في أسفل الصفحة.

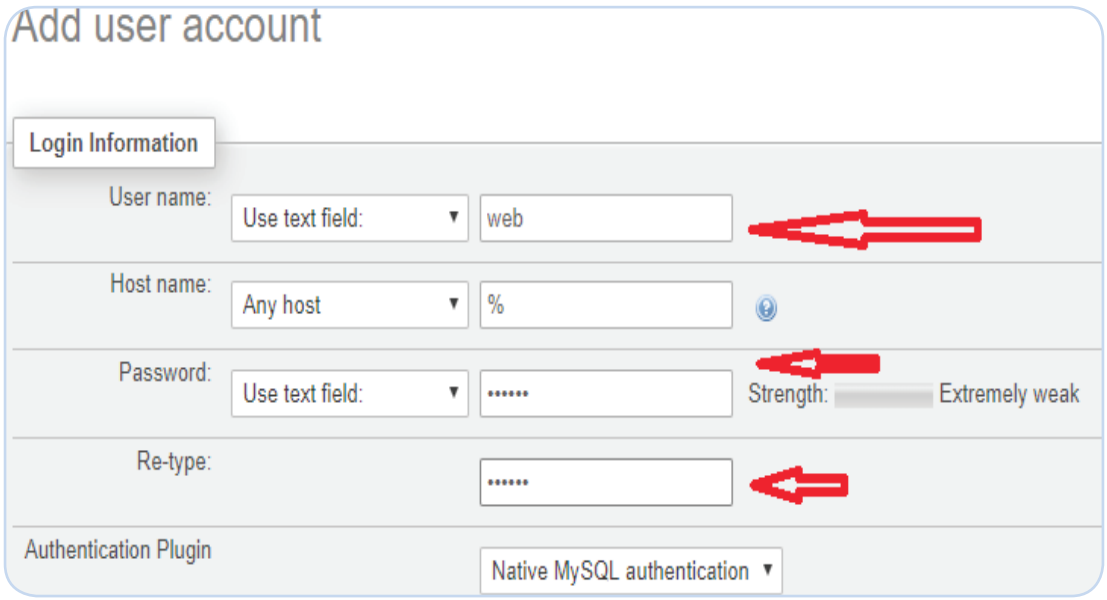


- يمكن الآن إضافة معلومات المستخدم الجديد:

1- اسم المستخدم: مثلاً web.

2- اسم المضيف: أتركه كما هو.

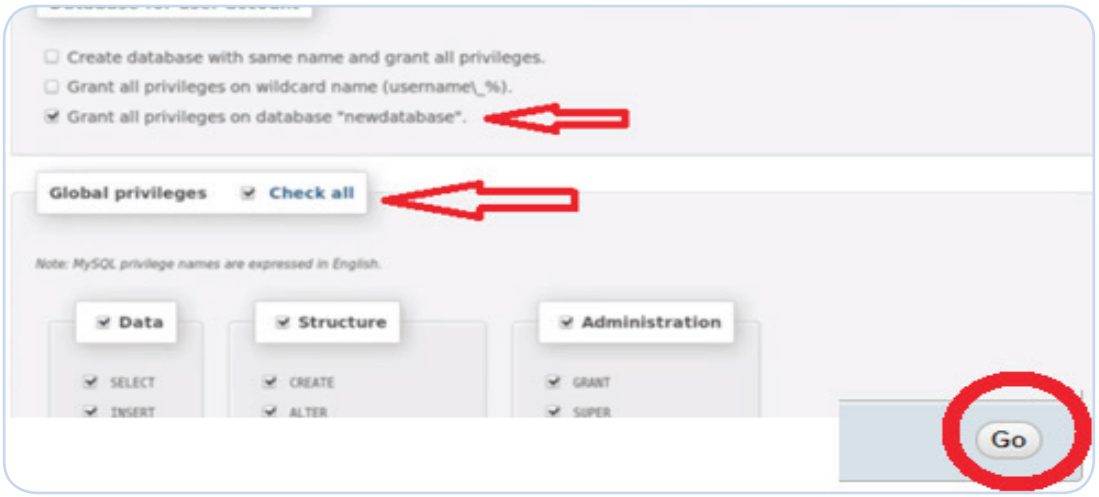
3- كلمة المرور التي تخوّل المستخدم من الوصول إلى قاعدة البيانات.



ثالثاً- تعيين صلاحيّات الوصول إلى قاعدة البيانات:

لتعيين صلاحيّات وصول المستخدم إلى قاعدة البيانات، أنتقل إلى القسم الثاني من صفحة الامتيازات:

- خيار Check All تحديد الكلّ: لمنح صلاحيّات الوصول كافّةً إلى قاعدة البيانات للمستخدم المحدّ
- أنقر على Go استمر.



نشاط (8) عمليّ: أُنشئ قاعدة بيانات باسم School، ثمّ أُضيف مستخدماً جديداً، وأعطيه كامل الصلاحيّات.

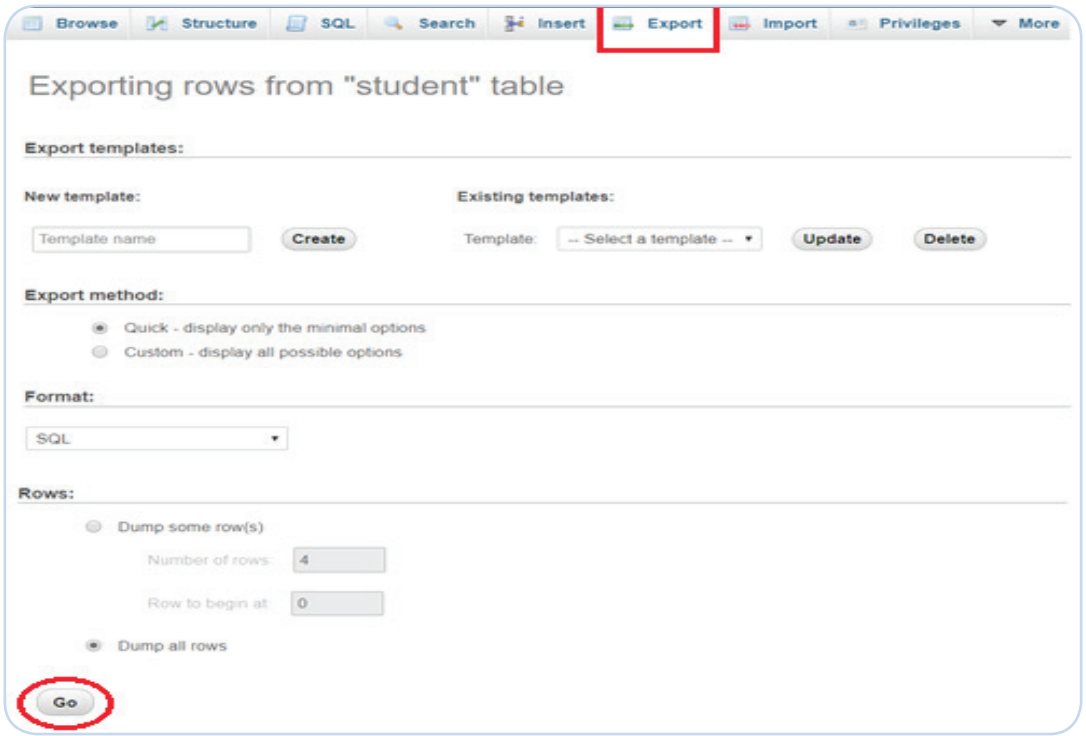
عمل Export وImport لقاعدة البيانات:

نقل قواعد البيانات بسرعة وكفاءة دون الحاجة للتعامل مع الأمر برمجيّاً، أو استخدام سطر الأوامر.

أوّلا: Export (تصدير).

خطوات تصدير قواعد البيانات:

1- الدخول إلى phpMyAdmin.
2- أحدّد قاعدة البيانات التي أريد نقلها.
3- الدخول إلى تبويب Export، يوجد بعض الإعدادات البسيطة تُترَك كما هي.
4- أنقر على Go استمر.
5- سيتمّ تحميل قاعدة البيانات مع الامتداد .sql وأحياناً تكون مضغوطة على هذا الشكل .sql.zip.

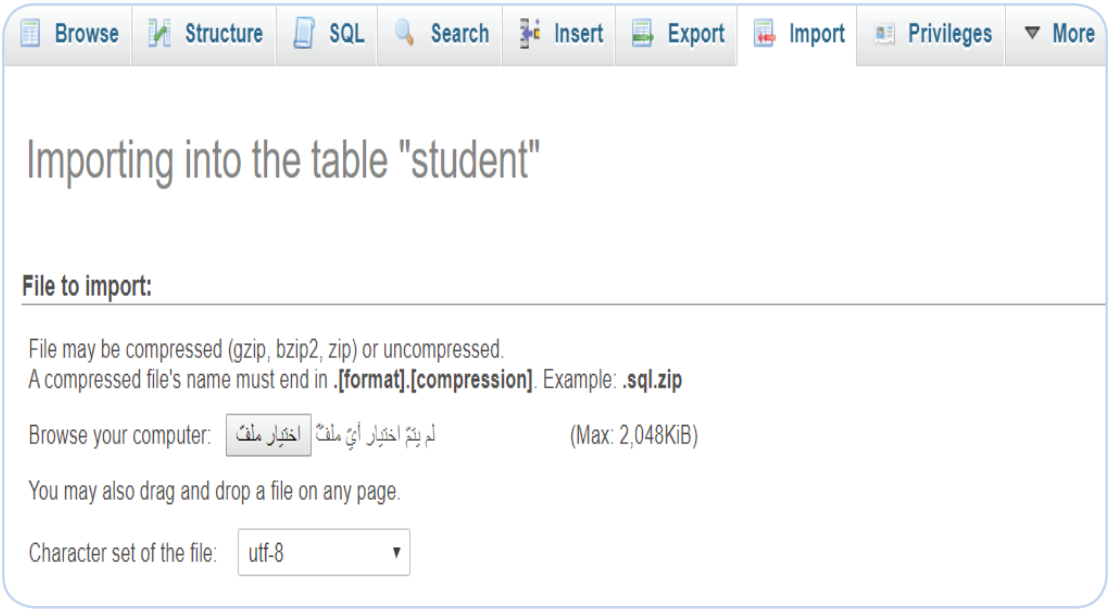


ثانياً: Import (استيراد).

خطوات استيراد قواعد البيانات:

1- الدخول إلى phpMyAdmin.
2- أنشئ قاعدة بيانات جديدة.
3- اختيار التبويت Import من أجل جلب قاعدة البيانات وتفريغها في قاعدة البيانات الجديدة.
4- أنقر على (اختيار ملفّ) Choose File وأحدّد قاعدة البيانات من مكان تواجدها على الجهاز.
5- النقر على GO استمر.

من المفترض أن تتمّ العمليّة بنجاح كما هو موضّح في الصورة التالية، وقد تستغرق عمليّة الجلب بعض الوقت بحسب حجم الملفّ، وأيضاً بحسب سرعة الإنترنت، وتحديداً سرعة الرفع.



نشاط (9) عمليّ: أطبّق Export وImport لقاعدة البيانات School.

الأسئلة

1. أوضّح مناهج قواعد البيانات.
2. أذكرُ ثلاثة أمثلة على قواعد البيانات.
3. أذكر مكوّنات قواعد البيانات.
4. ما شروط تسمية قاعدة البيانات والجداول والأعمدة؟

الموقف التعليميّ التعلُّميّ (2-2):

إنشاء و إدارة قواعد البيانات

وصف الموقف التعليميّ:

كلّف مالك مؤسّسة لتفصيل وتسويق الملابس متخصّصاً في قواعد البيانات إنشاء قاعدة بيانات للموقع الخاصّ بالمؤسّسة.

العمل الكامل			
خطوات العمل	وصف الموقف الصفّيّ	المنهجيّة (استراتيجيّة التعلُّم)	الموارد
أجمع البيانات وأحلّلها	أجمع البيانات من مالك المؤسّسة عن: • محتوى الموقع المراد إنشاء قواعد البيانات له (صفحة تسجيل دخول، صفحة إداريّة، المنتجات، العمّال...). أجمع البيانات: • طرق إنشاء قواعد البيانات • أوامر إنشاء وإدارة جداول قواعد البيانات • أنواع البيانات (نصوص، أرقام،...) • عدد الجداول المستخدمة، وطرق الربط بينها • طريقة ربط قاعدة البيانات بصفحة php.	• الحوار والمناقشة. • التعلُّم التعاونيّ (مجموعات العمل). • البحث العلميّ.	• وثائق: الطلب الخاصّ بمالك المؤسّسة، كتب عن إنشاء قواعد البيانات. • التكنولوجيا: الإنترنت، مقاطع فيديو، مواقع إنترنت موثوقة عن قواعد البيانات، حاسوب.
أخطّط وأقرّر	-أحدّد: • طرق إنشاء قواعد البيانات. • أوامر إنشاء وإدارة جداول قواعد البيانات. • أنواع البيانات (نصوص، أرقام،...). • عدد الجداول المستخدمة، وطرق الربط بينها. • طريقة ربط قاعدة البيانات بصفحة php. • إعداد خطّة زمنيّة للتنفيذ.	• تعلّم تعاونيّ/ مجموعات. • حوار ومناقشة.	• وثائق (طلب مالك المؤسّسة). • حاسوب. • قرطاسيّة.

أُنفّذ	• إنشاء قاعدة البيانات الجديدة باستخدام phpMyAdmin. • توضيح أوامر إنشاء وإدارة جداول قواعد البيانات. • معرفة أنواع البيانات التي ستخزّن في قاعدة البيانات. • إنشاء الجداول المستخدمة في الموقع والربط بينها. • استخدام أوامر إنشاء وإدارة جداول قواعد البيانات. • ربط قاعدة البيانات بصفحة php. • عقد جلسة مع مالك المؤسّسة والاتّفاق على صحّة وجاهزيّة قاعدة البيانات. • إجراء التعديلات المطلوبة، وتجهيز قاعدة البيانات النهائيّة.	• تعلّم تعاونيّ/ مجموعات. • حوار ومناقشة. • عصف ذهنيّ/ استمطار الأفكار.	• إنترنت، حاسوب. • البيانات التي تمّ جمعُها. • قرطاسيّة. • الطلب الخاص بمالك المؤسّسة.
أتحقّق	أتحقّق من: • إنشاء قاعدة البيانات الجديدة باستخدام phpMyAdmin. • توضيح أوامر إنشاء وإدارة جداول قواعد البيانات. • معرفة أنواع البيانات التي ستخزّن في قاعدة البيانات. • إنشاء الجداول المستخدمة في الموقع والربط بينها. • استخدام أوامر إنشاء وإدارة جداول قواعد البيانات. • ربط قاعدة البيانات بصفحة php.	• تعلّم تعاونيّ / مجموعات. • حوار ومناقشة.	• البيانات التي تمّ جمعُها. • إنترنت، حاسوب. • الخطّة المعدّة سابقاً.
أوثّق وأُقدّم	أوثّق: • طرق إنشاء قواعد البيانات. • أوامر إنشاء وإدارة جداول قواعد البيانات. • أنواع البيانات (نصوص، أرقام،...). • عدد الجداول المستخدمة، وطرق الربط بينها. • طريقة ربط قاعدة البيانات بصفحة php. • إعداد ملفّ بالحالة (طلب مالك المؤسّسة). • إعداد عرض تقديميّ. • إعداد ملخّص يتضمّن طرق إنشاء وإدارة قواعد البيانات، وعدد الجداول وأوامرها، وأنواع البيانات المستخدمة فيها، وطريقة ربطها بصفحة php.	• حوار ومناقشة. • تعلُّم تعاونيّ/ مجموعات عمل.	• البيانات التي تمّ جمعُها. • حاسوب. • شاشة عرض. • قرطاسيّة.

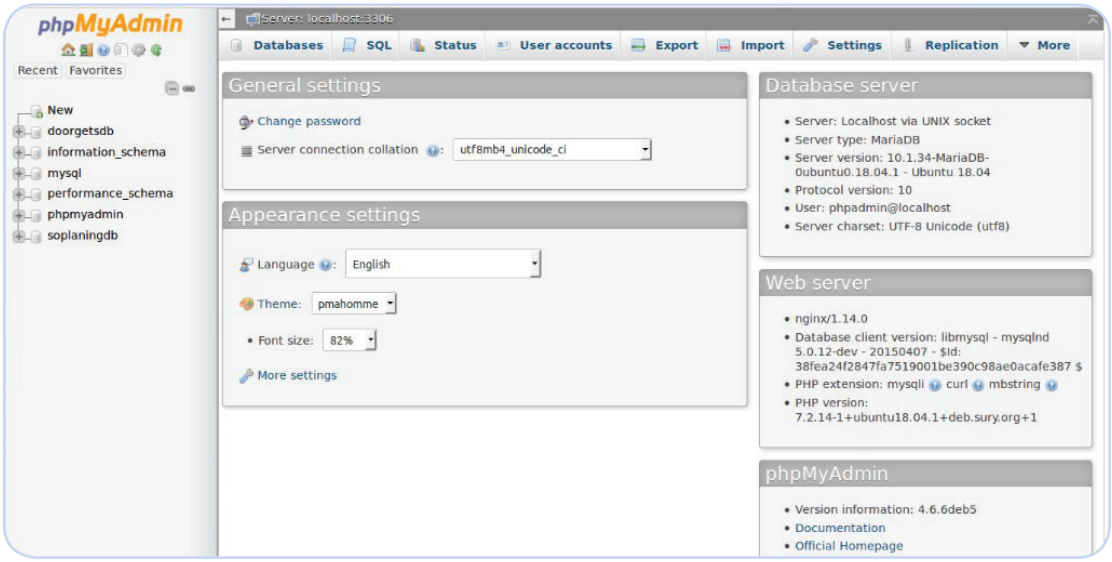
	• رضا مالك المؤسّسة عن قاعدة البيانات المعدّة. • ملاءمة قاعدة البيانات للمواصفات والمعايير.	• حوار ومناقشة.	• مجلّات تتحدّث عن المواصفات الخاصّة بقواعد البيانات. • طلب مالك المؤسّسة.

أناقش حتّى تعمل الصفحة الرئيسيّة لقواعد البيانات phpMyAdmin لابدّ من تفعيل apache.

أتعلّم

إنشاء وإدارة قواعد البيانات

أتامّل وأناقش الصورة الآتية:



لغة الاستعلام البنيويّة Structured Query Language (SQL):

هي اللغة المستخدمة لإجراء عمليّات على قواعد البيانات، بما في ذلك إضافة، أو تحديث، أو حذف البيانات من قاعدة البيانات، أو لتعديل بنية قاعدة البيانات نفسها، وتنقسم إلى:

1- لغة تعريف البيانات Data Definition Language (DDL): هي اللغة المستخدمة لتعريف بنية البيانات ضمن قاعدة بيانات، وتشمل على الكلمات المحجوزة الآتية:

- CREATE: إنشاء قاعدة البيانات والجداول.
- ALTER: التعديل على قاعدة البيانات والجداول.
- DROP: حذف قاعدة البيانات والجداول.
- RENAME: إعادة تسمية قاعدة البيانات والجداول.
- TRUNCATE: إفراغ الجداول من البيانات.

2- لغة التعامل مع البيانات Data manipulation language (DML): وهي اللغة الخاصّة بالتعامل مع البيانات نفسها داخل قواعد البيانات من:

- INSERT: إدخال بيانات جديدة.
- DELETE: حذف بيانات.
- UPDATE: تحديث بيانات.

3- لغة التحكُّم بالبيانات Data control language (DCL): وهي اللغة الخاصّة بمنح المستخدمين صلاحيّات معيّنة، مثل :

- GRANT: تُستخدَم لمنح المستخدمين صلاحيّات معيّنة لأداء مهمّات معيّنة.
- REVOKE: تُستخدَم لإلغاء الصلاحيّات التي تمّ منحها بالأمر السابق.

4- لغة الاستعلام عن البيانات Data Query language (DQL): وتشمل:

- SELECT: استعلام عن بيانات معيّنة .
- SHOW: تُستخدَم لعرض قواعد البيانات المنشأة والجداول.

5- لغة نقل البيانات Data Transaction Language (DTL): وتشمل:

- COMMIT: حفظ التغييرات المُجراة على قاعدة البيانات نتيجة لتنفيذ العمليّات.
- ROLLBACK: يستخدم لإلغاء العمليّات المُجراة على قاعدة البيانات؛ ما يعني تجاهل التغييرات التي طرأت عليها.

محرّكات قواعد البيانات (Storage Engine): صُمّمت للسماح بتخزين البيانات، ولها أنواع مختلفة، أشهرها (InnoDB , MyISAM)، تتعامل MySQL مع الأمور بطريقة مختلفة وراء الستار؛ أي بغضّ النظر عن محرّك التخزين المستخدم فإنّ التعامل مع قواعد البيانات بالطريقة نفسها .

نشاط (1) للاطّلاع: أُعدُّ تقريراً عن مميّزات كلّ من محرّكات البحث (InnoDB, MyISAM).

أنواع البيانات Data Types في قاعدة البيانات:

يُحدّد نوع البيانات طبيعة وشكل القيمة التي من الممكن أن يأخذها العمود في الجدول، ويلزم تحديد اسم العمود ونوع بياناته عند إنشاء هذا الجدول، وذلك في جملة إنشائه.

الوصف	نوع البيانات
نصّ حرفيّ ثابت الطول، طوله (n)	CHARACTER(n)
نصّ حرفيّ مَرِن الطول، أقصى طول (n)	VARCHAR(n) or CHARACTER VARYING(n)

BINARY(n)	نصّ ثنائيّ ثابت الطول، طوله (n)
BOOLEAN	تخزّن قيمة TRUE or FALSE
VARBINARY(n) or BINARY VARYING(n)	نصّ ثنائيّ مرن الطول، أقصى طول (n)
INTEGER(p)	رقم صحيح عدد الخانات أقصاه (p)
SMALLINT	رقم صحيح عدد الخانات أقصاه (6)
INTEGER	رقم صحيح عدد الخانات أقصاه (11)
BIGINT	رقم صحيح عدد الخانات أقصاه (20)
DECIMAL(p,s)	رقم عدد خاناته (p) وعدد الخانات بعد الفاصلة (s)
DATE	تاريخ (Y-M-D)
TIME	وقت (H:M:S)
TIMESTAMP	تاريخ ووقت (Y-M-D H:M:S)

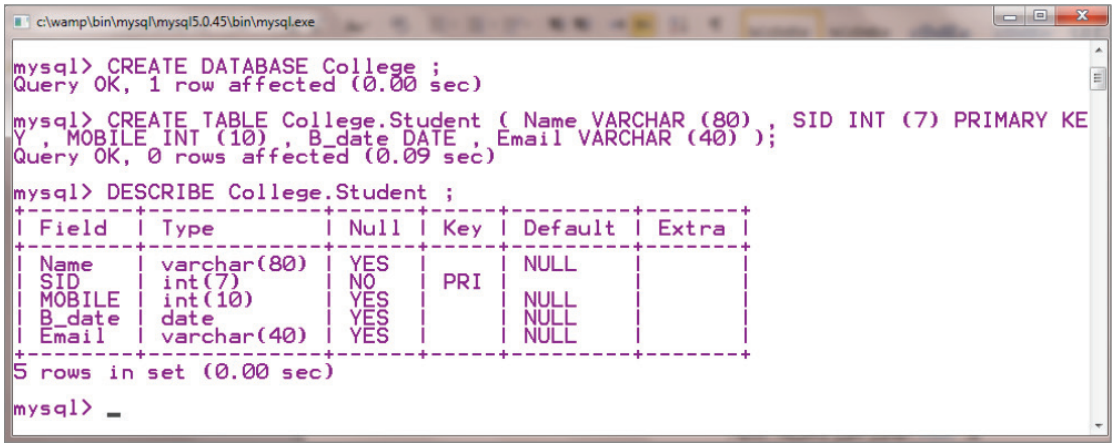
نشاط (2) نظريّ: أحدّد نوع البيانات الأنسب لكلّ من الحقول الآتية:

(الاسم - العمر - السكن - تاريخ الميلاد - رقم الهويّة - الراتب).

التعامل مع قاعدة البيانات:

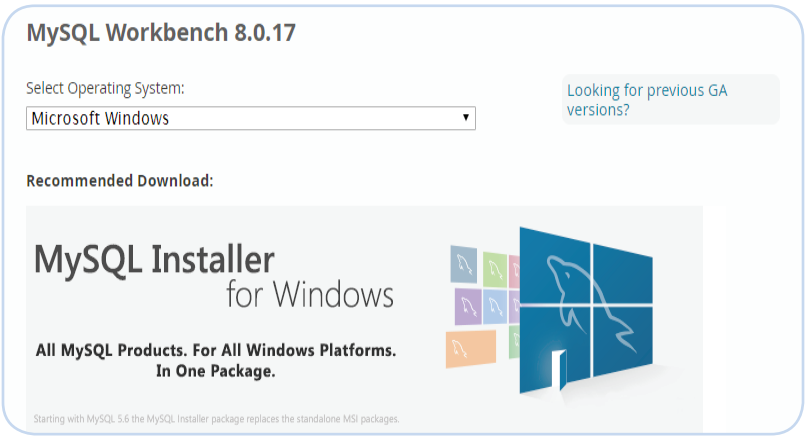
هنالك طرق عدّة للتعامل مع قاعدة البيانات MySQL، منها:

1- طريق سطر الأوامر Command Line، كما في الشكل الآتي:



2- استخدام الواجهة الرسوميّة، وأشهرها PhpMyAdmin الذي تمّ تنصيبه في الموقف الأوّل، وهو ما سيتمّ استخدامه في بناء قاعدة البيانات.

3- برنامج MySQL Workbench.

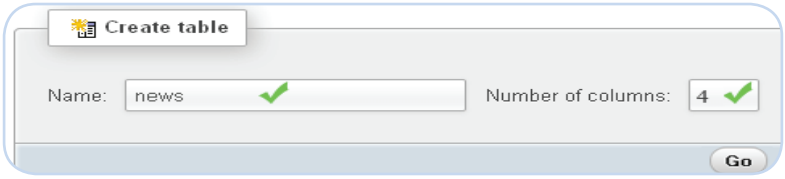


إنشاء جدول داخل قاعدة بيانات:

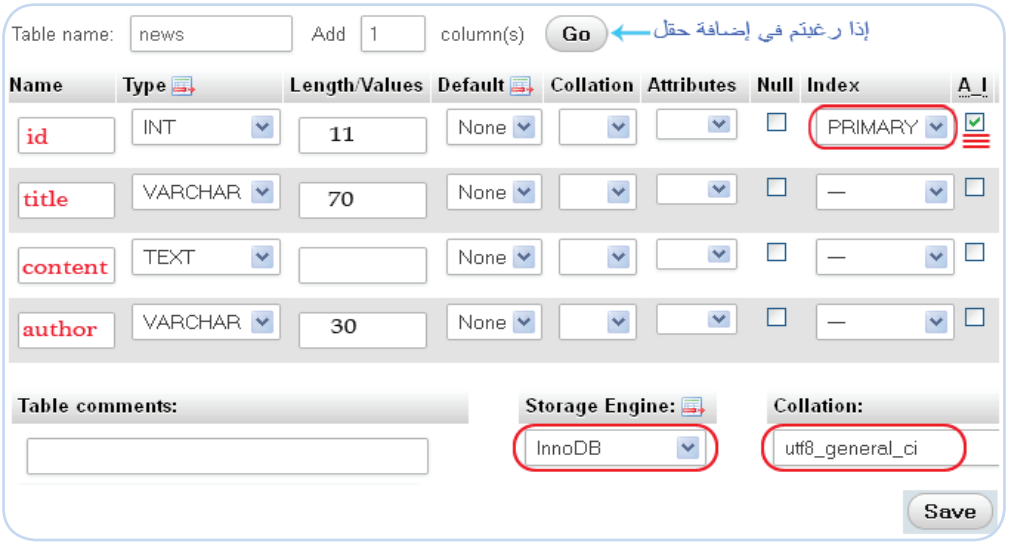
أوّلاً- باستخدام الواجهات:

1- أختار اسم قاعدة البيانات، وبما أنّني لم أنشئ بعد أي جدول ، فبالتالي ليس لديّ أيّ محتوى.

2- ستظهر صفحة تدعونا لإنشاء جدول "Create Table"،أكتب اسم الجدول وعدد الأعمدة، ثمّ أنقر "Go".



3- تظهر صفحة إدخال أسماء الحقول، وأنواعها، وخصائصها، والقيود عليها، كما في الشكل الآتي:



4- أملأ الحقول كما هو موضّح، وذلك تبعاً للجداول التي تمّ تصميمها.

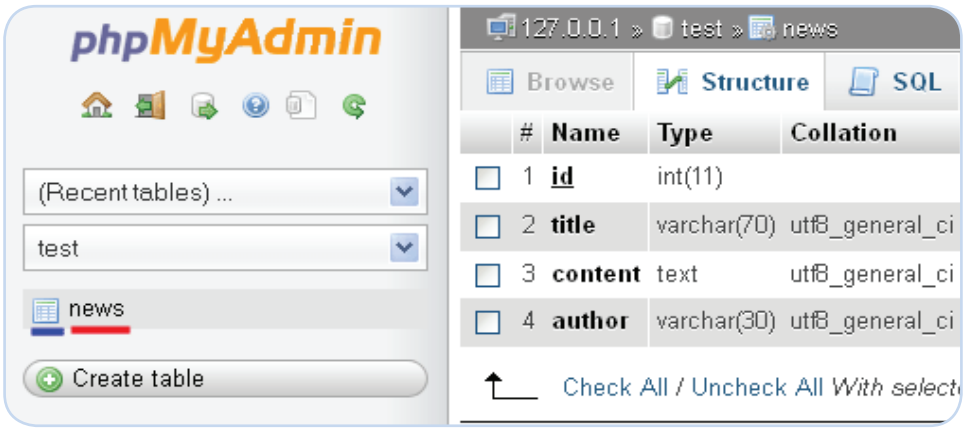
5- أحدّد المفتاح الرئيسيّ في الجدول، وبجانبه خيار (A_I)، ويعني Auto_Increament أي تتمّ زيادة الترقيم تلقائيّاً دون إدخال من المستخدم.

6- أحدّد محرّك قواعد البيانات وأحدّد الترميز.

7- أضغط على save ليتمّ حفظ الجدول.

8- لعرض الجدول بعد إنشائه:

- أختار قاعدة البيانات، وأحدّد الجدول.
- أذهب إلى الوصلة Structure (بناء) من القائمة الأفقيّة العلويّة، كما في الشكل الآتي:



ثانياً- طريقة الأكواد:

1- أضغط على تبويب SQL، ثمّ أكتب في المربّع كما يأتي:

```
CREATE TABLE table_name(
column_name1 data_type(size),
column_name2 data_type(size),
column_name3 data_type(size),
....
);
```

- Table_name: اسم الجدول المنشأ
- Column_name: اسم العمود
- Data_type(size): نوع البيانات وحجمها

مثال (1):

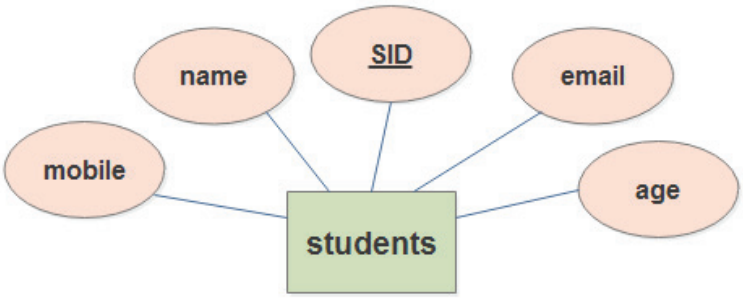
```
CREATE TABLE news (
ID INT (11) PRIMARY KEY,
title varchar (70),
content text,
author VARCHAR (30)
 );
```

2- لعرض معلومات الجدول المنشأ، أكتب كالآتي:

```
DESCRIBE Student ;
```

نشاط (3) عمليّ: أنشئ قاعدة البيانات School.

- أنشئ جدول الطالب Student وفق رسم ERD المعطى، وأحدّد أنواع الحقول:



القيود CONSTRAINS:

هي الشروط التي يجب أن تتحقّق على القيم المدخلة في الجدول (الصفوف)، ويُعدّ موضوع القيود من المواضيع المهمّة جدّاً في قواعد البيانات، ويجب على المبرمج ومسؤول قواعد البيانات الإلمام بأنواع القيود، وكيفيّة تعريفها، والتعامل معها.

الهدف العام من القيود هو الحفاظ على سلامة البيانات ودقّتها.

أنواع القيود:

1- قيود المفاتيح الأساسيّة Primary Key:

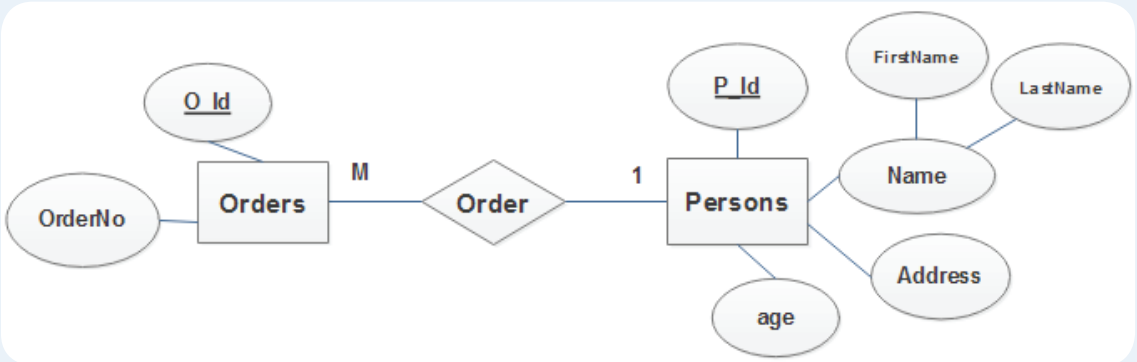
يُطبَّق هذا القيد على عمود أو أكثر، تكون قيمة العمود أو قيمة الأعمدة مجتمعة تُعَرِّف كلّ سجلّ على نحو فريد عن السجلّات الأخرى في الجدول.

خصائص قيد المفتاح الرئيسيّ:

- يمكن تعريف قيد مفتاح رئيسيّ واحد على مستوى الجدول.
- قيم العمود المطبّق عليه قيد المفتاح الرئيسيّ لا بدّ أن تكون فريدة لكلّ سجلّ، وألّا تأخذ قيمة فارغة.
- تُعَرّف قيود المفتاح الأجنبيّ بأخذها مَرجِعاً من مفتاح قيد رئيسيّ في جدول آخر.

مثال (2):

- يُعرَّف قيد المفتاح الرئيسيّ كما يأتي (مستخدما ERD الآتي):



- يمكن استخدام طريقة الواجهات، كما تمّ شرحه في خطوات إنشاء الجدول، أو باستخدام الأوامر النصّيّة كما يأتي :

```
CREATE TABLE Persons (
P_Id int NOT NULL,
LastName varchar(255) NOT NULL,
FirstName varchar(255),
Address varchar(255),
Age int(11),
PRIMARY KEY (P_Id) );
```

- تعريف قيد مفتاح رئيسيّ على أكثر من عمود، ونعطيه الاسم PK_Person :كالآتي:

```
CREATE TABLE Persons (
P_Id int NOT NULL,
LastName varchar(255) NOT NULL,
FirstName varchar(255),
Address varchar(255),
CONSTRAINT PK_Person PRIMARY KEY (P_Id,LastName));
```

2- قيد المفتاح الأجنبيّ Foreign Key Constraint:

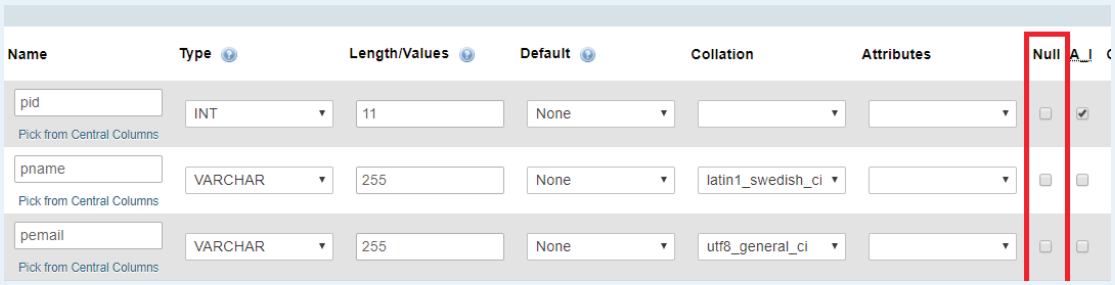
يُعرّف قيد المفتاح الأجنبيّ عموداً أو أكثر في الجدول على أنّه مرجع من عمود يوجد في جدول آخر، يُعرَّف عليه قيد فريد أو قيد مفتاح رئيسيّ، ويُعدّ هذا القيد وسيلة لربط جداول قاعدة البيانات، وبناء علاقات بينها، ومن الممكن تعريف أكثر من قيد أجنبيّ في الجدول الواحد.

3- قيد العمود غير الفارغ Not Null Constraint:

يمكن إعطاء قيمة فارغة لعمود ما لم يُعرَّف هذا القيد عليه، ويُقصَد به منع إدخال أو إعطاء العمود قيمة فارغة Null عند إجراء عمليّات، مثل: الإضافة، أو التعديل على السجلّات.

مثال (3):

1- استخدام طريقة الواجهات كما في الشكل الآتي دون تحديد الخيار Null:



2- طريقة الأكواد كما يأتي:

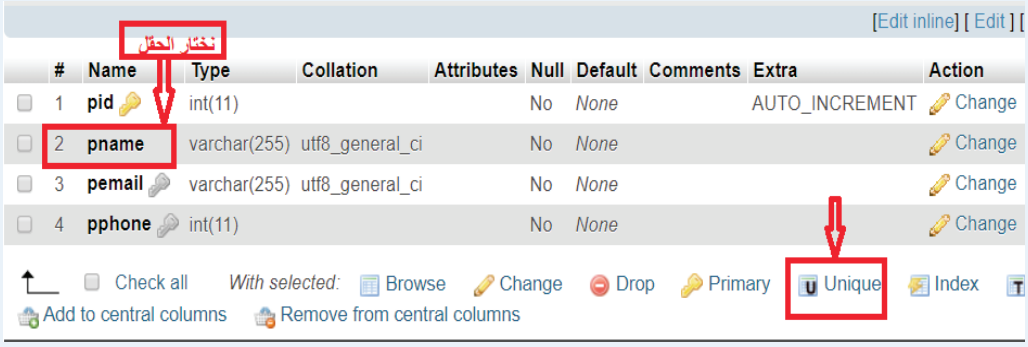
```
CREATE TABLE Persons (
    ID int NOT NULL,
    LastName varchar(255) NOT NULL,
    FirstName varchar(255) NOT NULL,
    Age int
);
```

4- قيد القيمة الفريدة UNIQUE Constraint:

مَهمّة هذا القيد ضمان عدم تكرار قيمة عمود في أيّ من سجلّات الجدول، تكون هذه القيمة فريدة ومختلفة. يمكن تعريف أكثر من قيد فريد في الجدول على أكثر من عمود، كما أنّ قيد المفتاح الرئيسيّ يقدّم ضمان القيمة الفريدة للعمود بجانب القيد الفريد.

مثال (4):

يُعرَّف القيد الفريد بطريقة الواجهات، وذلك بتحديد الحقل، ومن ثمّ الضغط على كلمة UNIQUE، كما هو موضّح في الشكل:



أو باستخدام طريقة الأكواد:

```
CREATE TABLE Persons (
    ID int NOT NULL UNIQUE,
    LastName varchar(255) NOT NULL,
    FirstName varchar(255) NOT NULL,
    Age int
);
```

5- قيد الفحص CHECK Constraint:

يُستخدم قيد الفحص للتحكُّم في القيمة التي من الممكن أن يأخذها العمود الذي طُبّق عليه القيد، وتتمّ عمليّة التحكُّم هذه بإجراء عمليّات مقارنة (<=>)،(>=)، (=)، (< >)،(Like،in) على قيم العمود؛ للتأكّد من أنّها تتوافق مع شروط وحالات معيّنة.

مثال (5):

يُضاف قيدُ الفحص باستخدام الجملة (CHECK)، كالآتي:

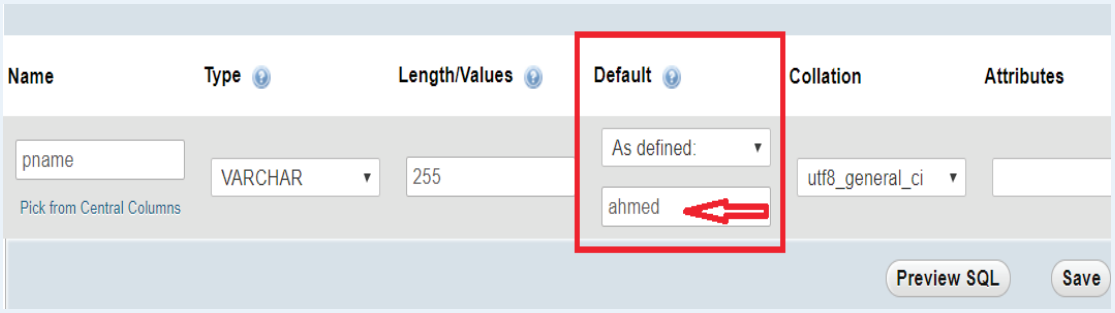
```
CREATE TABLE Persons (
    ID int NOT NULL,
    LastName varchar(255) NOT NULL,
    FirstName varchar(255),
    Age int CHECK (Age>=18)
);
```

6- قيود افتراضيّة Default:

تمكّن القيود الافتراضية من تعريف القيمة التي سيتمّ تزويدها لعمود متى فشل مستخدم في إدخال إحدى القيم.

مثال (6):

أحدّد القيمة الافتراضيّة بطريقة الواجهات كما يأتي:



أحدّد القيمة الافتراضيّة بطريقة الأكواد:

```
CREATE TABLE Persons (
P_Id int NOT NULL,
LastName varchar(255) NOT NULL,
FirstName varchar(255),
Country varchar(255) DEFAULT 'Palestine');
```

تعديل جدول وحذف البيانات:

1- أمر ALTER: يُستخدَم للتعديل في تصميم جدول تمّ إنشاؤه مسبقاً داخل قاعدة بيانات.

الصيغة العامة لأمر ALTER:

```
ALTER TABLE table_name action;
```

Table_name: اسم الجدول المراد التعديل عليه.

Action: الفعل المراد تطبيقه للأمر ALTER.

أفعال الأمر ALTER	أمثلة على الصيغة العامّة
إضافة عمود إلى داخل جدول	`ALTER TABLE students ADD birth-date date;`
حذف عمود من الجدول	`ALTER TABLE students DROP birthdate ;`
تعديل عمود داخل الجدول (تغيير اسم عمود mobile)	`ALTER TABLE students CHANGE Mobile mob int(11);`
تعديل عمود داخل الجدول باستخدام (Modify تغيير نوع البيانات)	`ALTER TABLE students Modify Name char(100);`
إضافة قيد constrain (أضيف قيد القيمة الفريدة للاسم)	`ALTER TABLE students ADD UNIQUE(name);`
تغيير اسم الجدول student	`RENAME TABLE students to stud;`

2- أمر Drop:

يُستخدم لحذف الجدول أو قاعدة البيانات، ولا يمكن استرجاع الجدول أو قاعدة البيانات عند حذفها بواسطة الاستعلام DROP.

الصيغة العامّة لحذف قاعدة البيانات:

```
DROP DATABASE databasename;
```

الصيغة العامّة لحذف جدول:

```
DROP TABLE table_name;
```

3- أمر TRUNCATE: يستخدم لحذف جميع البيانات الموجودة في جدول (الصفوف).

الصيغة العامّة لإفراغ جدول:

```
TRUNCATE TABLE table_name;
```

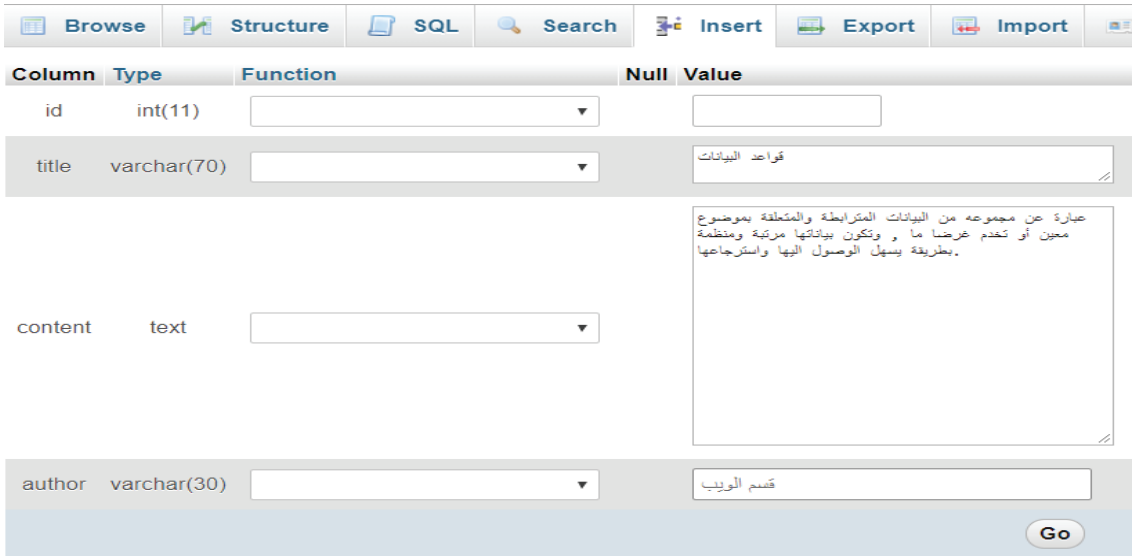
إدخال البيانات:

INSERT: هي إحدى الكلمات المحجوزة التابعة للغة التعامل مع البيانات DML، وهي اللغة المستخدمة للتعامل مع البيانات داخل قاعدة البيانات.

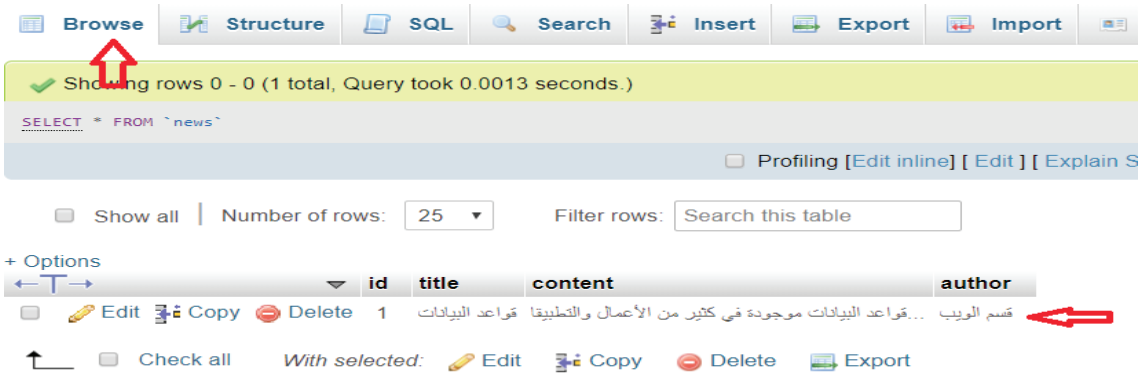
يوجد طريقتان لإدخال البيانات:

الأولى- طريقة الواجهات:

- أنقر على الوصل 'INSERT' (إضافة) في القائمة العلويّة تضيف محتوى جديداً إلى الجدول.
- أملأ حقول الجدول بالبيانات كما هو موضّح.



- ترك حقل "id" فارغاً؛ لأنّه المفتاح الأساسيّ للجدول .
- أنقر على "GO" ليتمّ التخزين .
- لعرض البيانات المخزّنة، أذهب إلى عرض "Browse" في القائمة العلويّة.



الثّانية - كتابة أوامر SQL، وتتمّ بشكلين:

- إدخال القيم بحسب ترتيب الأعمدة

```
INSERT INTO table_name
VALUES (value1,value2,value3,...);
```

- إدخال أسماء الأعمدة، ثمّ إدخال القيم بحسب ترتيب الأعمدة

```
INSERT INTO table_name (column1,column2,column3,...)
VALUES (value1,value2,value3,...);
```

عند إدخال البيانات يجب مراعاة نوع البيانات المخزّنة في العمود:

1- إذا كانت من نوع INT نستطيع إدخال القيمة مباشرة.

2- إذا كانت من نوع نصوص، مثل: VARCHAR، أو تاريخ Date، أو وقت time فيجب إدخال القيمة بين علامتَي التنصيص.

نشاط (4) عمليّ: أدخل بيانات إلى جدول الطلاب Student الذي تمّ إنشاؤه داخل قاعدة البيانات School باستخدام أكثر من طريقة.

الاستعلام عن البيانات:

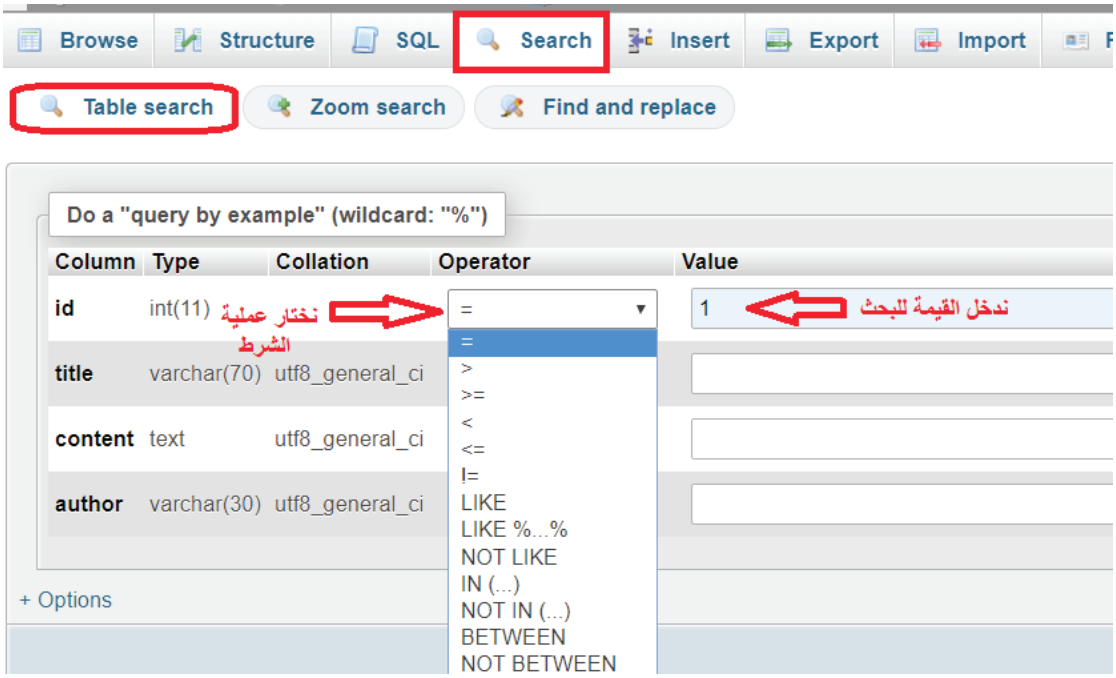
SELECT: هي إحدى الكلمات المحجوزة التابعة للغة الاستعلام عن البيانات DQL.

الاستعلام عن البيانات الموجودة في جدولٍ ما بطريقتين:

أوّلاً- طريقة الواجهات :

1- اختيار من القائمة العلويّة بحث «Search» ، ثمّ «Table search».

2- اختيار عمليّة الشرط والعمود المراد تنفيذ الشرط عليه، والقيمة المراد الاستعلام عنها.



ثانياً - طريقة الأكواد النصّيّة، وتتمّ بشكلين:

- **الشكل الأوّل:** الاستعلام عن البيانات لجميع الأعمدة في الجدول.

```
SELECT * FROM Table_Name ;
```

- **الشكل الثاني:** الاستعلام عن بيانات لأعمدة معيّنة في الجدول بكتابة أسماء أعمدة الجدول المراد عرض بياناتها.

```
SELECT column1, column2 … FROM Table_Name ;
```

عند الرغبة في إضافة شرط معيّن في البيانات المراد الاستعلام عنها تُستخدَم الكلمة المحجوزة where ، ثم بتعريف الشرط باستخدام اسم العمود، ثمّ علامة شرطيّة، ثمّ القيمة كالآتي:

```
SELECT * FROM Table_Name
where column1='value' ;
```

أمثلة: بالاستعانة بجدول Student المُنشأ سابقاً:

- أستعلم عن بيانات الطالب الذي رقمه 2.

```
SELECT * FROM students WHERE  SID= 2 ;
```

- أستخدم علامات المقارنة داخل شرط الاستعلام، وأستعلم عن بيانات الطلّاب الذين أعمارهم أكبر من 17 مرتّبين تبعاً لِـ SID تصاعديّاً .

```
SELECT * FROM students WHERE age > 17 ORDER BY SID;
```

- أستعلم عن بيانات الطلّاب الذين أعمارهم أكبر من 17 مرتّبين تبعا لِـ SID تنازليّاً .

```
SELECT * FROM students WHERE age > 17 ORDER BY SID DESC;
```

Asc: ترتيب تصاعديّ وهو الافتراضيّ .

Desc: ترتيب تنازليّ.

- أستعلم عن بيانات الطلّاب الذين تبدأ أسماؤهم بحرف m.

```
SELECT * FROM students WHERE name like 'm%';
```

%: معناها قد يكون بعد حرف m حرف، أو أكثر، أو لا يوجد حروف.

- أستعلم عن بيانات الطلاب الذين أسماؤهم يكون فيها حرف h هو الحرف الثّاني.

```
SELECT * FROM students WHERE name like '-h%';
```

- : معناها أنه يوجد حرف واحد قبل حرف h.

- أستعلم عن بيانات الطلّاب الذين أعمارهم بين 16, 20 .

```
SELECT * FROM students WHERE age between 16 and 20;
```

- أستعلم عن بيانات الطلّاب الذين أعمارهم تكون في: 16, 18, 20.

```
SELECT * FROM students WHERE age in(16,18,20);
```

أشهر الدوال المستخدمة في SQL:

الدالّة (FUNCTIONS)	الصيغة العامّة
AVG() متوسّط	SELECT AVG(column_name) FROM table_name;
COUNT() عدد القيم	SELECT COUNT(column_name) FROM table_name;
FIRST (), LAST()	SELECT FIRST(column_name) FROM table_name;
MIN (), MAX()	SELECT MIN(column_name) FROM table_name;
SUM () مجموع	SELECT SUM(column_name) FROM table_name;

نشاط (5) عمليّ: أستخدم جدول الطالب Student الذي تم إنشاؤه داخل قاعدة البيانات School، وأعطي أمثلة على كلّ من الدوال في الجدول السابق.

تحديث البيانات UPDATE:

تتمّ عمليّة الـ Update بطريقتين:

أوّلاً- طريقة الواجهات:

+ Options

	StudentID	Name	Spassword	Phone
Edit Copy Delete	2	Ahmed	1234	59111111
Edit Copy Delete	3	ali	1234	2533
Edit Copy Delete	4	zaki	1234	2533

Check all *With selected:* Edit Copy Delete Export

بالإمكان تعديل محتويات الحقول، وذلك بالضغط على أيقونة Edit الموضّحة في الشكل الآتي، عند الضغط عليها تفتح صفحة insert تمكّن من تعديل البيانات.

ثانياً- طريقة الأكواد البرمجيّة:

الصيغة العامّة لتعديل سجلّات في جدول:

```
UPDATE table_name
SET column1 = ‘value1’, column2 = ‘value2’, ...
WHERE condition;
```

- UPDATE: كلمة محجوزة تعني تحديث.
- SET: كلمة محجوزة تعني وضع.
- column1: اسم العمود المراد تعديل قيمة السجلّ الموجود داخله.
- Value: القيمة الجديدة (يجب أن تكون القيمة داخل علامات التنصيص ’ ’).
- Where: كلمة محجوزة تعني حيث (تستخدم لإدراج شرط).
- Condition: عبارة عن نصّ الشرط (يمكن أن يستخدم في الشرط العلامات = > < ...إلخ).

مثال (7):

أعدّل على جدول الطالب (تعديل email, age) للطالب الذي رقمه 3:

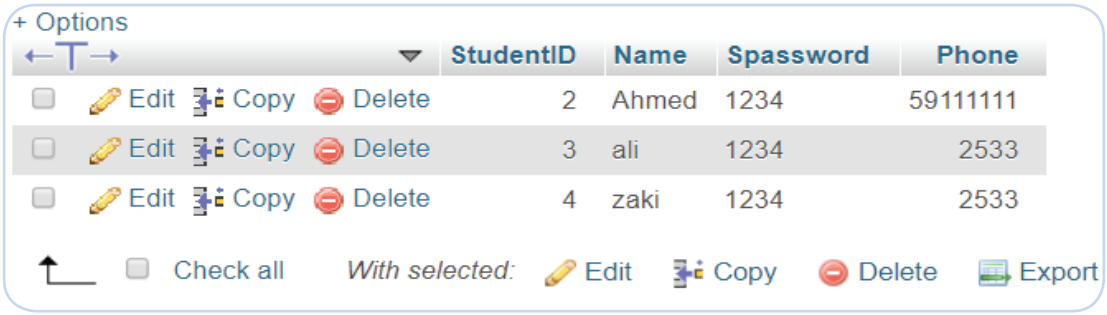
```
UPDATE student SET
 email = 'star@hotmail.com' ,
 age = 18
 WHERE SID = 3 ;
```

حذف البيانات DELETE:

تتمّ عمليّة الـ DELETE بطريقتين:

أوّلاً- طريقة الواجهات:

بالإمكان حذف الحقول، وذلك بالضغط على أيقونة Delete الموضّحة في الشكل الآتي:
عند الضغط عليها يتمّ حذف الحقل المحدّد، وبالإمكان حذف حقول عدّة.



ثانياً- طريقة الأكواد البرمجيّة:

الصيغة العامّة لحذف سجلّات من جدول:

```
DELETE FROM table_name WHERE condition;
```

- DELETE: كلمة محجوزة تعني حذف.
- Table_name: اسم الجدول المخزّن في قاعدة البيانات.
- Where: كلمة محجوزة تعني حيث (تستخدم لإدراج شرط).
- Condition: هي نصّ الشرط الذي سيتم الحذف بناءً عليه.

مثال (8):

أحذف بيانات الطالب الذي رقمه 7

```
DELETE FROM student WHERE SID = 7;
```

JOIN: تعليمة مهمّة، لا غنى عنها، من تعليمات لغة SQL تُتيح ربط بيانات جدولين أو أكثر أثناء الاستعلام، وبطبيعة الحال يجب أن تكون الجداول مُرتبطة بينها بمفاتيح خارجيّة .

الصيغة العامّة لربط الجداول:

```
SELECT column_name(s)
FROM table1
JOIN table2
ON table1.column_name=table2.column_name;
```

- JOIN: كلمة محجوزة وتعني الربط.
- Table1: الجدول الأوّل.
- Table2: الجدول الثاني.
- ON: تُستخدم لإدراج الشرط (يمكن استخدام where).

نشاط (6) عمليّ:

طوّرَ مركز تدريب تطبيق ويب يستخدم قاعدة بيانات تُسمّى TCenterDB، تحتوي على جدول الدورات (Courses) التي يتمّ التدريب عليها.

في ضوء الجدول الآتي أوظّف لغة SQL لكتابة التعليمات البرمجيّة التالية لكلّ ممّا يأتي:

Courses			
Id	Name	Hours	Price
1	تطوير مواقع إنترنت	40	500
2	رخصة قيادة الحاسوب	50	400
3	التخطيط الاستراتيجيّ	45	350

- إنشاء جدول الدورات مع تحديد القيود الآتية: (id: مفتاح رئيسيّ، Name: حقل فريد، Hours: لا تقبل null).
- تخزين البيانات كما هو موضّح في الجدول.
- تعديل سعر دورة رخصة قيادة الحاسوب إلى 250.
- استرجاع بيانات الدورات التي عدد ساعاتها أكبر من 42، مرتّبة بحسب اسم الدورة.
- حذف دورة «التخطيط الاستراتيجيّ" .
- حذف قاعدة البيانات TCenterDB .

الاتّصال بقاعدة البيانات:

لاستثمار قاعدة البيانات بكلّ إمكانيّاتها لا بدّ من إنشاء وفتح اتّصال بين قاعدة البيانات وصفحات الموقع، وذلك قبل إجراء أيّ من العمليّات، إنشاء صفحة php تحتوي على كود يحقّق الاتّصال بقاعدة البيانات، يتمّ استدعاؤها في بداية كلّ صفحة ويب تستخدم قاعد البيانات.

هناك أربعة متطلّبات أساسيّة للاتّصال بقاعدة البيانات:

1- servername: عنوان السيرفر المحليّ، وهو localhost أو 127.0.0.1.

2- username: اسم المستخدم الأساسيّ لقاعدة البيانات.

3- password: كلمة المرور للمستخدم المتّصل بقاعدة البيانات قد تكون فارغة.

4- dbname: اسم قاعدة البيانات المراد الاتّصال بها.

مثال (9):

أنشئ صفحة php لربط قواعد البيانات:

```php
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname="school";

// انشاء الاتصال
$conn = mysqli_connect($servername, $username, $password,$dbname);

// التحقق من الاتصال
if (!$conn) {
    die("فشل الاتصال: " . mysqli_connect_error());
}
else {echo "تم نجاح الاتصال";}
?>
```

1- أنشئ ملفّاً جديداً وليكن config.php مثلاً، أتّصل من خلاله على قاعدة البيانات.

2- أعرّف المتغيّرات الأربعة التي تمّ شرحها سابقاً.

3- أعرف المتغيّر conn الذي يخزّن بيانات الاتّصال.

4- أبني الدالّة mysqli_connect المسؤولة عن الاتّصال وتأخذ 4 قيم.

5- التحقّق من الاتّصال باستخدام أداة الشرط (if).

mysqli:

عبارة عن مجموعة دوال mysql محسّنة ومطوّرة لتصبح ذات أداء وفعاليّة أكبر، والاسم mysqli ينطبق على هذا المعنى (mysql improved).

دوال PHP للتحكُّم في قواعد البيانات:

الدالّة	استخدامها
mysqli_connect()	الاتّصال بقاعدة البيانات.
mysqli_connect_error ()	التحقُّق من وجود خطأ في الاتّصال، وتطبع الخطأ أيضاً.
mysqli_connect_errno()	عمل الدالّة السابقة نفسها، لكنّ الفرق أنّ هذه تطبع رقم الخطأ.
mysqli_query()	تنفّذ أوامر داخل قاعدة البيانات (إرسال، الاستعلام).
mysqli_fetch_assoc()	تحوّل النتائج المستخرجة من قاعدة البيانات إلى مصفوفة.
mysqli_num_rows()	حساب عدد أسطر أو عدد البيانات المضافة في جدول معيّن.
mysqli_close()	إغلاق قاعدة البيانات.
mysqli_num_fields()	حساب عدد حقول سطر معيّن في الجدول.
mysqli_select_db()	اختيار قاعدة البيانات.
mysqli_free_result()	تحرير الاتّصال، وتُستخدم عند تنفيذ الأمر SELECT فقط.

بعد الاتّصال بقواعد البيانات لا بدّ من إجراء العمليّات عليها، من إدخال للبيانات، وعرضها، وتحديثها، وحذفها، باستخدام الأوامر الآتية:

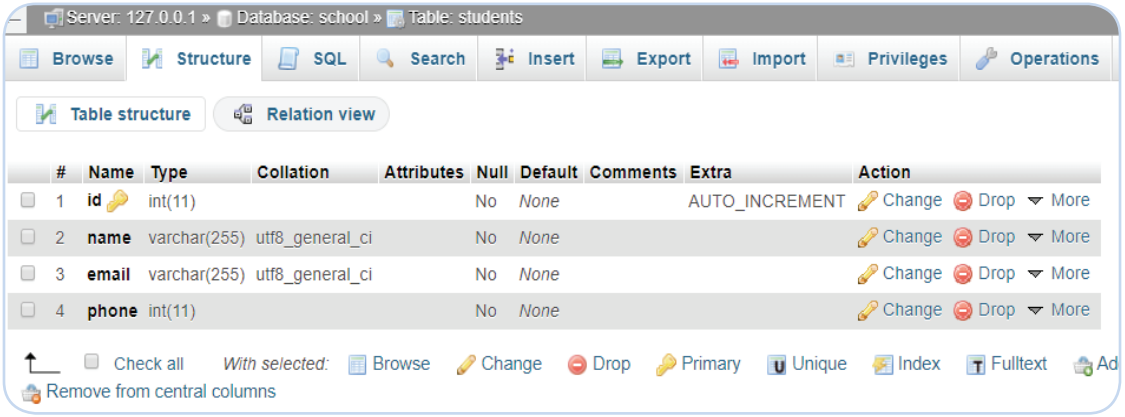
1- INSERT: إدخال البيانات في القاعدة.
2- SELECT:اختيار البيانات من القاعدة.
3- UPDATE: تحديث بيانات القاعدة.
4- DELETE: حذف البيانات من القاعدة.

أوّلاً- إدخال البيانات:

لتخزين البيانات داخل الجدول سوف تستخدم INSERT INTO التي تمّ شرحها مسبقاً في قواعد البيانات.

نشاط (7) عمليّ: إدخال البيانات من خلال صفحة ويب على قاعدة البيانات school:

أوّلاً- قاعدة بيانات:



ثانياً- أنشئ مجلّداً جديداً باسم أختاره.

ثالثاً- أنشئ صفحة PHP بداخلها FORM الإدخال، كما يأتي:

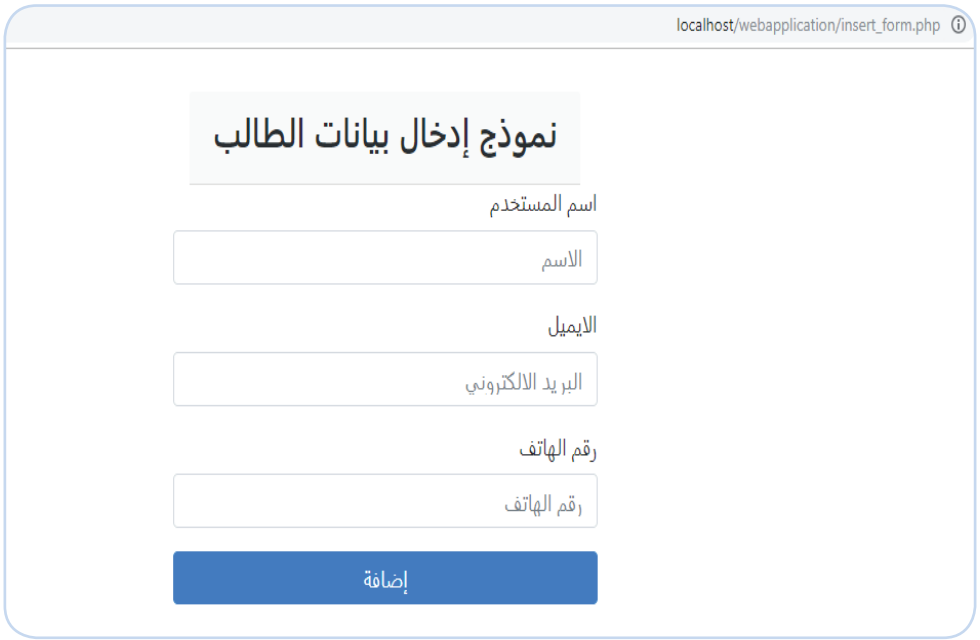
```
<form action="insert.php" method="POST" style="position:absolute;
right:450px;top:30px" ><div >
<div class="card-header"> <h3 >نموذج إدخال بيانات الطالب </h3> </div>
  <div class="form-group row">
    <label >اسم المستخدم</label>
   <input type="text" name="name" class="form-control" placeholder="الاسم">
  </div>
  <div class="form-group row">
    <label >     الايميل</label>
  <input type="text" name="email" class="form-control" placeholder="الايميل ">
  </div>
  <div class="form-group row">
    <label> رقم الهاتف</label>
   <input type="text" name="phone" class="form-control" placeholder="رقم الهاتف">
  </div>
  <div class="form-group row">
  <button type="submit" name="submit" class="btn btn-primary btn-block">إضافة </button>
  </div>
      </div>
</form>
```

1- أسمّي الصفحة (insert_form.php).

2- أنشئ ثلاثة حقول (الاسم، الإيميل، رقم الهاتف)، ولا أضيف حقلاً إلى رقم الطالب؛ لأنّه يأخذ قيماً تلقائيّة Auto_Increament.

3- أحدّد (action) وطريقة انتقال البيانات للصفحة المخصّصة بعد الضغط على زر (submit).

4- أختبر الصفحة كالآتي:



رابعاً- أُنشئ صفحة أخرى لاستقبال البيانات والاستعلام لإدخال البيانات إلى قاعدة البيانات:

1- أستدعي ملف الاتّصال config.php باستخدام دالّة include() المنشأ في المثال السابق.

2- أستقبل البيانات إلى متغيّرات باستخدام دالّة $_POST.

3- وضع الاستعلام داخل متغيّر $sql، أو أيّ متغيّر آخر.

4- أضيف دالّة mysqli_query لتنفيذ الاستعلام على قاعدة البيانات المتّصل عليها.

5- أتحقّق من نجاح الاستعلام باستخدام تعليمة if().

6- أنهي الاتّصال mysqli_close().

- دالّة isset(): تتحقّق من أيّ متغيّر تمّ إنشاؤه .

• الكود:

```php
<?php
include('config.php');
if(isset($_POST['submit'])){
$name=$_POST['name'];
$email=$_POST['email'];
$phone=$_POST['phone'];

$sql="insert into students(name,email,phone) values
('$name','$email','$phone')";
$result=mysqli_query($conn,$sql);
if($result){
    echo "<br>"." تمت عملية الإضافة بنجاح ";
}else{
    echo "فشل في إضافة البيانات".mysqli_error($conn);
}
}
mysqli_close($conn);
?>
```

خامساً- أضيف بيانات، ثمّ أنتقل إلى قاعدة البيانات للتأكُّد من إضافتها.

SELECT (تحديد البيانات من القاعدة):

لتحديد البيانات من القاعدة يلزم وجود جدول على الأقلّ فيه بيانات، واستخدام جملة الاستعلام select بأشكالها كافّة؛ لاسترجاع البيانات إلى صفحة php.

نشاط (8) عمليّ: تحديد واسترجاع البيانات من قاعدة البيانات للنشاط السابق:

أُنشئ ملفّاً جديداً بعنوان select.php داخل المجلّد المُنشأ، وأكتب الكود الآتي:

```php
<body>
    <table border="1px solid" align="center" class="table-sm">
    <tr bgcolor="#ccccbb">
        <td>id</td>
        <td>name</td>
        <td> email</td>
        <td>phone</td>
    </tr>
<?php
include('config.php');
$sql="select * from students";
$result=mysqli_query($conn,$sql);
if(mysqli_num_rows($result)>0){

while($row=mysqli_fetch_assoc($result)){
echo "<tr><td>".$row['id']."</td>
<td>".$row['name']."</td>
<td>".$row['email']."</td>
<td>".$row['phone']."</td></tr>";
}
}else{
    echo "لا توجد نتائج";
}
mysqli_close($conn);
?>
</table>
</body>
```

عرض البيانات داخل جدول

هنا باقي الصفوف للجدول وبداخلها الأعمدة ومحتوياتها

1- أتّصل بقاعدة البيانات من خلال تضمين الملفّ config.php

2- أكتب جملة الاستعلام select لاسترجاع جميع الأعمدة من قاعدة البيانات.

3- أعرّف متغيّر $result وبداخله دالّة الاستعلام mysqli_query.

4- اكتب شرط while للمرور على جميع الصفوف التي تحوي البيانات، أعرّف بداخله المتغيّر $row والذي يحوي بداخله الدالّة mysqli_fetch_assoc() التي تنفّذ الاستعلام، والاحتفاظ بالبيانات المسترجعة داخل مصفوفة .

5- أستخدم المتغيّر $row الذي يحوي بيانات المصفوفة لإخراج بيانات الأعمدة التي أريد بهذا الشكل: $row['name']؛ لجلب بيانات من عمود (name) وهكذا... .

localhost/webapplication/select.php

phone	email	name	id
222557	ah@hotmail.com	ahmed	1
258877	al@hotmail.com	ali	2
539877	mj4@hotmail.com	moh	3
266877	rm@hotmail.com	rami	4
253557	hass@hotmail.com	hassan	5

6- أفتح الصفحة، وأختبر النتيجة:

أناقش بالإمكان استرجاع البيانات وفق الشرط الذي أريد، وبالشكل الذي أريد.

تعديل البيانات وحذفها (DELETE & UPDATE)

نشاط (9) عمليّ: بالاعتماد على النشاط السابق أجري عمليتَي التعديل والحذف على بيانات الجدول:

أوّلاً: تعديل البيانات

1- أنشئ ملفّاً جديداً باسم edit.php، ثم أُضمّن ملفّ الاتّصال config.php.

```
27  while($row=mysqli_fetch_assoc($result)){
28  echo "<tr><td>".$row['id']."</td>
29  <td>".$row['name']."</td>
30  <td>".$row['email']."</td>
31  <td>".$row['phone']."</td>
32  <td>"."<a href='edit.php?id=".$row['id']."'>edit</a>"."</td>
33  <td>"."<a href='delete.php?id=".$row['id']."'>delete</a>"."</td>
34  </tr>";
35  }
```

2- بالعودة إلى ملفّ select.php السابق أُنشئ خليتين جديدتين في الصفّ: واحدة تكون رابطاً لعمليّة التعديل، والأخرى رابطاً للحذف .

- بما أنّني أريد التعديل على صفّ معيّن فلا بدّ أن أحدّد الصفّ المراد التعديل عليه، وأفضل وسيلة لذلك هي استخدام رقم الصفّ (id)، حيث إنّه مفتاح أساسيّ لا يتكرّر.
- قمت بعمل رابط لكلّ من الصفحتين، واستخدمت علامة ? ؛ وذلك لتمرير رقم الصفّ إلى الصفحة المراد الانتقال إليها في رابط الصفحة الذي يحمله المتغيّر id.

localhost/webapplication/select.php

delete	update	phone	email	name	id
delete	edit	222557	ah@hotmail.com	ahmed	1
delete	edit	258877	al@hotmail.com	ali	2
delete	edit	539877	mj4@hotmail.com	moh	3
delete	edit	54545	H@hot	zaki	4
delete	edit	253557	hass@hotmail.com	hassan	5

3- أتوجّه لصفحة edit، وأكتب الكود الآتي :

```
33    <div class="form-group row">
34        <button type="submit" name="submit" class="btn btn-primary
35        btn-block">تحديث </button>
36    </div>
37        </div>
38  </form>                                  نهاية الفورم

39
40  <?php
41  if(isset($_POST['submit'])){
42      $name=$_POST['name'];
43      $email=$_POST['email'];
44      $phone=$_POST['phone'];
45
46
47  $sql="update students set name='$name',email='$email',phone='$phone'
48   where id=".$_GET['id'];          استقبال id من الرابط
49  $res=mysqli_query($conn,$sql);
50
51  header('location:select.php');  الانتقال الى صفحة select
52  }
53  ?>
54  </body>
```

- أنشئ فورماً خاصّاً بالتعديل شبيهاً بفورم الإضافة، وأحدّد طريقة الإرسال(POST)، وأحدّد الـ action الذي سيُترك فارغاً؛ لأنّ كود update في الصفحة نفسها.
- عند الضغط على edit من صفحة select يتوجّه للصفحة الآتية:

- أكتب جمل ودوال الاستعلام الخاصّة بتعديل قيم الحقول السابقة للطالب الذي مرّرنا رقمه في رابط الصفحة .
- استخدمت دالّة ()isset؛أيّ لن يتمّ التعديل على البيانات ما لم يتمّ الضغط على تحديث وملء الحقول .
- جلب رقم id المرسل في الرابط باستخدام $_GET['id']
- أُنشئ الدالّة (header) للانتقال إلى صفحة select بعد تحديث البيانات.

ثانياً- حذف البيانات (DELETE):

عند الضغط على رابط الحذف في صفحة select.php تنتقل مباشرة لصفحة delete.php التي سأنشئها، كما يأتي:

1- تضمين ملفّ الاتّصال config.php في البداية.
2- أكتب جملة الاستعلام لحذف الطالب المخزّن رقمه داخل المتغيّر id الذي تمّ تمريره في الرابط عند الضغط على رابط الصفحة delete في الجدول .
3- أُنشئ الدالّة (header) للانتقال إلى صفحة select بعد حذف البيانات.

```php
<?php
include('config.php');

$sql="delete from students where id=".$_GET['id'];
$res=mysqli_query($conn,$sql);

header('location:select.php');
?>
```

الأسئلة

1. أذكر ثلاثة من أنواع البيانات في قواعد البيانات.
2. أُعدّد أنواع القيود في قواعد البيانات.
3. أذكر أشهر الدوال المستخدمة في SQL.
4. ما المقصود بِـ mysqli؟