# شرح مناقشة مشروع Wesal

## توزيع الشرح على سبع طالبات

هذا الشرح مخصص للمناقشة، ومكتوب بطريقة بسيطة وقريبة من الكلام الطبيعي. كل طالبة تشرح الجزء المحدد لها، وبعد ما تخلص تنتقل للطالبة اللي بعدها.

---

# الطالبة الأولى: مقدمة الفصل الرابع

## ماذا تقول؟

"أنا راح أشرح مقدمة الفصل الرابع. هذا الفصل يتكلم عن تصميم نظام Wesal، يعني يوضح كيف رتبنا مكونات النظام قبل التنفيذ وكيف ترتبط مع بعض. شرحنا في الفصل التصميم العام للنظام، والكلاسات والعلاقات بينها، وطريقة تفاعل المستخدمين مع النظام، وسير عملية التدريب من التقديم إلى إكمال التدريب. بعد ذلك وضحنا تصميم قاعدة البيانات والجداول والعلاقات بينها، وأخيرًا تكلمنا عن تصميم الواجهات حسب نوع المستخدم."

"اعتمدنا في التصميم على المستخدمين الأساسيين في النظام، وكل مستخدم له صلاحيات ووظائف مختلفة حسب دوره. الهدف من الفصل هو تحويل المتطلبات اللي ذكرناها في الفصل الثالث إلى تصميم واضح نقدر نعتمد عليه في التنفيذ."

## جملة الانتقال

"بعد ما وضحنا هدف الفصل، زميلتي راح تشرح البنية العامة للنظام من خلال System Architecture."

---

# الطالبة الثانية: System Architecture

## ماذا تقول؟

"هذا المخطط يوضح البنية العامة لنظام Wesal وكيف تتواصل الأجزاء الرئيسية مع بعض. المستخدم يبدأ من واجهة الويب، والواجهة هي الجزء اللي يتعامل معه المستخدم بشكل مباشر. استخدمنا فيها HTML وCSS وJavaScript وBootstrap لبناء الصفحات وتنسيقها."

"بعد الواجهة يجي دور تطبيق PHP، وهو المسؤول عن معالجة طلبات المستخدمين وتنفيذ العمليات الأساسية، مثل تسجيل الدخول، إدارة الصلاحيات، إنشاء الفرص، استقبال الطلبات، متابعة التدريب، والتعامل مع التقارير والتقييمات. بعد ذلك يتصل PHP بقاعدة بيانات MySQL عشان يحفظ البيانات أو يسترجعها عند الحاجة."

"فالمخطط يوضح المسار بشكل مبسط: المستخدم يتعامل مع الواجهة، والواجهة تتواصل مع PHP، وPHP يتعامل مع قاعدة البيانات."

## إذا سألت الدكتورة عن فائدة تقسيم النظام

"التقسيم يساعدنا نفهم النظام بشكل أوضح، بحيث تكون الواجهة منفصلة عن منطق النظام وعن تخزين البيانات، وهذا يسهل عملية التنظيم والتعديل أثناء التنفيذ."

## جملة الانتقال

"بعد ما وضحنا البنية العامة، زميلتي راح تشرح الكلاسات الرئيسية والعلاقات بينها في Class Diagram."

---

# الطالبة الثالثة: Class Diagram

## ماذا تقول؟

"مخطط Class Diagram يوضح أهم الكلاسات الموجودة في النظام والعلاقات بينها. بدأنا بكلاس User لأنه يمثل حسابات المستخدمين والأدوار الموجودة في النظام. بعده عندنا StudentProfile لبيانات الطالب، وOrganizationProfile لبيانات جهة التدريب."

"عندنا بعد ذلك TrainingOpportunity ويمثل فرصة التدريب اللي تنشرها جهة التدريب، وApplication ويمثل طلب الطالب على الفرصة. إذا تم قبول الطلب، يتكون Training ويمثل سجل التدريب. أثناء التدريب يرفع الطالب Report، وبعد نهاية التدريب تسجل جهة التدريب Evaluation للطالب."

"أضفنا Skill لتمثيل مهارات الطالب، ويستخدم مع بيانات الفرص في عملية التوصيات. أما RecommendationService فهو المسؤول عن حساب مدى توافق مهارات وتخصص الطالب مع متطلبات الفرص التدريبية."

"العلاقات في المخطط توضح مثلًا أن جهة التدريب تقدر تنشر أكثر من فرصة، والطالب يقدر يقدم على أكثر من فرصة، وكل طلب يكون مرتبطًا بطالب وفرصة معينة. كذلك التدريب الواحد ممكن يحتوي على أكثر من تقرير، ويكون له تقييم من جهة التدريب."

## إذا سألت الدكتورة عن العلاقة بين Application وTraining

"الطلب يتحول إلى سجل تدريب بعد ما يتم قبوله من جهة التدريب، لذلك كل طلب مقبول ممكن ينتج عنه سجل تدريب واحد."

## جملة الانتقال

"بعد ما وضحنا الكلاسات والعلاقات الثابتة، زميلتي راح تشرح مخططات Sequence Diagram اللي توضح ترتيب الخطوات أثناء تنفيذ العمليات."

# الطالبة الرابعة: Sequence Diagrams

## مقدمة قصيرة

"مخططات Sequence Diagram توضح ترتيب الخطوات أثناء تنفيذ عملية معينة، وتبين مين يبدأ العملية وكيف تنتقل البيانات بين المستخدم وواجهة النظام وتطبيق PHP وقاعدة البيانات."

## أولًا: تقديم الطلب

"في مخطط تقديم الطلب يبدأ الطالب بعرض الفرص التدريبية، وبعدها يختار فرصة ويشوف تفاصيلها. لما يضغط على التقديم، النظام يتحقق من أن الطالب ما سبق وقدم على نفس الفرصة. إذا ما كان فيه طلب سابق، يحفظ النظام الطلب بحالة Pending ويظهر للطالب أن الطلب تم إرساله. وإذا كان فيه طلب سابق، يمنع النظام التقديم المكرر."

## ثانيًا: مراجعة الطلب

"في مخطط مراجعة الطلب تبدأ جهة التدريب بفتح الطلبات الخاصة بالفرص التابعة لها. بعدها تختار طلبًا وتشوف بيانات الطالب، ثم تختار قبول أو رفض. إذا تم الرفض تتغير حالة الطلب إلى Rejected، وإذا تم القبول تتغير إلى Accepted ويتم إنشاء سجل تدريب للطالب."

## ثالثًا: رفع التقرير ومراجعته

"في مخطط التقرير يفتح الطالب التدريب النشط ويحدد نوع التقرير، سواء تقرير منتصف التدريب أو التقرير النهائي، ثم يرفعه للنظام. يحفظ النظام بيانات التقرير وتكون حالته Submitted. بعد ذلك يفتح المشرف الجامعي التقرير ويضيف ملاحظته ويحدث حالته، وتظهر الحالة الجديدة للطالب."

## إذا سألت الدكتورة عن فائدة المخطط

"فائدته أنه يوضح ترتيب التفاعل خطوة بخطوة، ويساعدنا نتأكد أن كل عملية لها بداية ونتيجة واضحة."

## جملة الانتقال

"بعد ما شرحنا ترتيب العمليات، زميلتي راح تشرح سير العمل كاملًا من خلال Activity Diagram."

---

# الطالبة الخامسة: Activity Diagram

## ماذا تقول؟

"مخطط Activity Diagram يوضح سير عملية التدريب التعاوني من البداية إلى النهاية. تبدأ العملية بتسجيل الطالب ودخوله للنظام، وبعدها يكمل ملفه ويرفع السيرة الذاتية والمهارات. إذا كان الملف غير مكتمل يرجع الطالب ويكمله، وإذا كان مكتمل ينتقل إلى تصفح الفرص التدريبية."

"بعد اختيار الفرصة يرسل الطالب الطلب، وتقوم جهة التدريب بمراجعته. إذا تم رفض الطلب، يقدر الطالب يرجع ويبحث عن فرصة ثانية. أما إذا تم قبول الطلب، ينشئ النظام سجل تدريب ويبدأ الطالب التدريب."

"بعد ذلك يرفع الطالب التقارير، ويقوم المشرف الجامعي بمراجعتها. إذا احتاج التقرير تعديلًا يرجع للطالب، وإذا تمت الموافقة عليه يكمل الطالب بقية التدريب. وفي نهاية التدريب تقوم جهة التدريب بتقييم الطالب، ثم يتم تغيير حالة التدريب إلى مكتمل."

"المخطط يركز على سير العملية ونقاط القرار الرئيسية، مثل اكتمال الملف، قبول الطلب، والموافقة على التقرير."

## إذا سألت الدكتورة عن تقديم الطالب على أكثر من فرصة

"المخطط يسمح للطالب بالرجوع والبحث عن فرصة ثانية إذا تم رفض طلبه، والنظام يمنع فقط تكرار التقديم على الفرصة نفسها."

## جملة الانتقال

"بعد ما وضحنا سير العملية، زميلتي راح تشرح تصميم قاعدة البيانات والعلاقات من خلال ERD."

---

# الطالبة السادسة: Database Design وERD

## ماذا تقول؟

"قاعدة البيانات مصممة باستخدام MySQL، وهدفها تخزين البيانات بطريقة مرتبة وربط الجداول ببعض. عندنا جدول users للحسابات الأساسية والأدوار، وجدول student_profiles لبيانات الطلاب، وجدول organization_profiles لبيانات جهات التدريب."

"عندنا أيضًا جدول training_opportunities للفرص التدريبية، وجدول applications لطلبات الطلاب، وجدول trainings لسجلات التدريب بعد قبول الطلب. أما reports فيخزن التقارير التي يرفعها الطالب، وevaluations يخزن تقييم جهة التدريب للطالب."

"بالنسبة للمهارات، عندنا جدول skills وجدول student_skills، لأن الطالب ممكن تكون عنده أكثر من مهارة، والمهارة نفسها ممكن تكون موجودة عند أكثر من طالب."

"العلاقات الأساسية هي أن جهة التدريب تنشر عدة فرص، والطالب يرسل عدة طلبات، والطلب المقبول ينتج عنه تدريب. التدريب ممكن يحتوي على أكثر من تقرير ويكون له تقييم واحد من جهة التدريب."

"استخدمنا المفاتيح الأساسية للتعرف على كل سجل، والمفاتيح الأجنبية لربط الجداول. كذلك يمنع النظام الطالب من التقديم على نفس الفرصة أكثر من مرة."

## إذا سألت الدكتورة عن سبب استخدام جدول student_skills

"استخدمناه لأن العلاقة بين الطلاب والمهارات many-to-many؛ الطالب عنده أكثر من مهارة، والمهارة ممكن تكون موجودة عند أكثر من طالب."

## إذا سألت عن دور قاعدة البيانات

"قاعدة البيانات تحفظ المعلومات بشكل منظم، وتسمح للنظام باسترجاعها وتحديثها حسب صلاحية كل مستخدم."

## جملة الانتقال

"بعد شرح تصميم قاعدة البيانات، زميلتي راح تشرح التقنيات المستخدمة والجزء الخاص بالواجهات في التقرير."

---

# الطالبة السابعة: الفصل الخامس والتقنيات والواجهات

## أولًا: مقدمة الفصل الخامس

"الفصل الخامس يتكلم عن الأدوات والتقنيات المستخدمة في تطوير النظام، وبعدها يشرح التنفيذ والاختبارات والنتائج. حاليًا جهزنا أول قسمين من الفصل، أما قسم التنفيذ راح نكمله بعد ما ننتهي من تنفيذ النظام بشكل فعلي."

## ثانيًا: التقنيات المستخدمة

"استخدمنا PHP للجزء الخلفي ومعالجة العمليات، مثل تسجيل الدخول والتعامل مع الطلبات والصلاحيات. استخدمنا MySQL لتخزين بيانات المستخدمين والفرص والطلبات والتدريب والتقارير والتقييمات."

"استخدمنا HTML لبناء هيكل الصفحات، وCSS لتنسيق الشكل والألوان والمسافات. استخدمنا JavaScript لإضافة التفاعل والتحقق البسيط من البيانات. كذلك استخدمنا Bootstrap عشان يساعدنا في تصميم واجهات متجاوبة وتشتغل على أحجام شاشات مختلفة."

"استخدمنا XAMPP كبيئة تشغيل محلية، لأنه يوفر Apache لتشغيل ملفات PHP وMySQL لتشغيل قاعدة البيانات. واستخدمنا phpMyAdmin لإدارة الجداول والبيانات. أما Mermaid وdraw.io فاستخدمناها في تصميم المخططات الموجودة في التقرير."

## ثالثًا: شرح قسم الواجهات

"في قسم User Interface Design وضحنا مبدأ تصميم الواجهات والوظائف الأساسية لكل مستخدم. أما صور الواجهات النهائية فراح نوثقها بعد ما نبدأ التنفيذ العملي ونخلص تصميم الصفحات، عشان تكون الصور مأخوذة من النظام الفعلي وتوضح الشكل النهائي بشكل صحيح."

"بعد الانتهاء من الواجهات، راح نضيف صور الصفحات مثل لوحة الطالب، فرص التدريب، لوحة جهة التدريب، صفحة الطلبات، ولوحة المشرف، ونوثقها في المكان المناسب في التقرير."

## جملة ختامية

"بهذا نكون وضحنا تصميم النظام والتقنيات المستخدمة، وبعد اكتمال التنفيذ راح نكمل توثيق الواجهات والتنفيذ والاختبارات والنتائج في الفصل الخامس."

---

# إجابات قصيرة لأسئلة محتملة

## لماذا اخترتم PHP؟

"اخترنا PHP لأنها مناسبة لتطوير تطبيقات الويب، وتسمح لنا بمعالجة البيانات والتعامل مع النماذج وقاعدة البيانات، كما أنها مناسبة لمستوى المشروع والتقنيات المطلوبة في التخصص."

## لماذا اخترتم MySQL؟

"استخدمنا MySQL لأنها قاعدة بيانات علائقية مناسبة لتخزين البيانات المرتبطة بالنظام، مثل المستخدمين والفرص والطلبات والتقارير."

## لماذا استخدمتم Bootstrap؟

"استخدمناه لتسهيل تصميم واجهات مرتبة ومتجاوبة مع الشاشات المختلفة، ولتوحيد شكل الأزرار والجداول والنماذج."

## كيف تعمل التوصيات؟

"التوصيات تعتمد على مطابقة تخصص الطالب ومهاراته مع التخصص والمهارات المطلوبة في الفرص. بعدها يحسب النظام درجة توافق ويعرض الفرص المناسبة للطالب."

## لماذا التقييم من جهة التدريب؟

"لأن جهة التدريب هي الطرف اللي يتابع أداء الطالب العملي، وعنده معرفة مباشرة بالتزام الطالب وجودة عمله وتعاونه أثناء التدريب."

## لماذا يستطيع الطالب التقديم على أكثر من فرصة؟

"لأن هذا يعطي الطالب أكثر من خيار، خصوصًا إذا ما تم قبول طلبه الأول. لكن النظام يمنع التقديم على نفس الفرصة أكثر من مرة."

## ماذا يحدث بعد قبول الطلب؟

"تتغير حالة الطلب إلى Accepted، وبعدها ينشئ النظام سجل تدريب مرتبط بالطالب والفرصة."

## ما الفرق بين Class Diagram وERD؟

فيوضح ERD يوضح الكلاسات والعمليات والعلاقات على مستوى تصميم النظام، أما "Class Diagram جداول قاعدة البيانات والمفاتيح والعلاقات بينها."

## ما الفرق بين Sequence Diagram وActivity Diagram؟

Activity يوضح ترتيب التفاعل بين المستخدم وأجزاء النظام أثناء عملية معينة، أما "Sequence Diagram فيوضح سير العملية كاملًا ونقاط القرار فيها Diagram."

## لماذا لم تضعوا صور الواجهات النهائية الآن؟

"لأن صور الواجهات لازم تكون من النظام المنفذ فعليًا. لذلك راح نبدأ بتنفيذ الواجهات، وبعد الانتهاء منها نوثق الصور ونضيفها في القسم المناسب من التقرير."

## هل الواجهات مهملة في المشروع؟

"لا، الواجهات موجودة ضمن التصميم والنطاق، لكن توثيق صورها مرتبط بمرحلة التنفيذ العملي، وراح تتم إضافتها بعد الانتهاء من تصميم الصفحات."

---

# ملاحظات سريعة للطالبات أثناء المناقشة

- كل طالبة تشرح الجزء الخاص بها فقط وباختصار.
- لا تدخلون في تفاصيل ما سألت عنها الدكتورة.
- إذا سألت عن شيء غير منفذ، قولوا إن توثيقه النهائي بيكون بعد اكتمال التنفيذ العملي.
- استخدموا أسماء المخططات والعمليات الموجودة في التقرير كما هي.
- لا تحفظون الكلام حرفيًا؛ افهموا الفكرة وقولوا الكلام بأسلوبكم.
- ركزوا على وظيفة كل مخطط، وليس على شرح كل سهم أو كل كلمة داخله.

- إذا نسيت الطالبة نقطة، ترجع للفكرة الأساسية: المستخدم، PHP، وقاعدة البيانات.