

## أسئلة متكررة بخصوص أفكار المشاريع:

- يجب أن يكون هناك طريقة واضحة لإظهار نتائج المشروع، ولإدخال البيانات للalgorithm .. ولذلك قد يتطلب المشروع Basic GUI .. في هذه الحالة يجب أن يكون هناك Basic GUI لإدخال الإختيارات وإظهار النتائج، وليس على GUI أي درجات، ولكن عدم وجوده سبب في خصم درجات.
- الـ development platform هي الـ tools والـ programming languages والـ libraries المستخدمة في المشروع.
- الـ diagrams المطلوبة هي الـ block diagrams و الـ flowcharts توضح الـ algorithms المستخدمة، إلى جانب الـ plots توضح النتائج، إلى جانب أي UML Diagrams تجدونها ضرورية (مثلاً، الـ use-case diagram توضح الـ functions/features من وجهة نظر المستخدم).
- أفكار المشاريع التي بها الـ puzzles كالـ Sudoku والـ N-Queens والـ N-Puzzles، المشروع هو للعثور على حل للعبة بشكل عام (بداية الـ puzzle يجب أن تختلف كل مرة)، وليس لحالة واحدة خاصة فقط.
- أفكار المشاريع التي بها الـ Two-Player Games، المشروع هو لعمل لاعب ذكي واحد فقط لهذه اللعبة، بينما الطرف الآخر من اللعبة هو إنسان وليس آلة. ويجب على اللاعب الذكي استخدام الـ approach المحدد لتقرير كل حركة/خطوة يقوم بها من اللعبة (كرد على كل حركة يقوم بها الطرف الآخر).
- أفكار المشاريع التي بها الـ (heuristic functions)، يجب حل المشكلة باستخدام اثنين الـ heuristic functions على الأقل، ويجب حل المشكلة باستخدام كل الـ heuristic function بشكل منفصل، ثم المقارنة بين تأثير كل منها على سرعة الوصول للحل وجودة الحل.
- أفكار المشاريع التي بها الـ Two-Player Games، يجب تطبيق الـ Minimax والـ Alpha-Beta Pruning واستخدام الـ Heuristics بشكل منفصل (أي تطوير إمكانية تحديد كل حركة في اللعبة، مرة باستخدام الـ Minimax فقط دون أي Heuristics "إن كان الـ search state space بسيط كالـ Tic-Tac-Toe والـ 8-Puzzle، ثم مرة أخرى ولكن مع تطبيق الـ Alpha-Beta Pruning، ثم Minimax باستخدام إحدى الـ Heuristics، ثم Minimax باستخدام الـ Heuristic Function الثانية).
- في الـ Tic-Tac-Toe، يجب بالإضافة لما سبق تطبيق الـ Symmetry Reduction والـ Heuristic Reduction، كلاهما بشكل منفصل.
- أفكار المشاريع التي بها الـ Two-Player Games أو الـ puzzles، إذا لم يكن الـ board size موضح في الـ project description فيجب الالتزام بالـ standard board size (مثلاً الـ Intelligent Go Player يجب تطبيقه على board 19 في 19).
- أفكار المشاريع التي تعتمد على مدخلات مختلفة (Job Scheduling Problem، والـ Faculty's Timetable Scheduling Problem، والـ Vehicle Routing Problem، والـ Knapsack Problem)، يجب على المستخدم أن يدخل كل أو معظم المدخلات المطلوبة (وليس أن يقوم الـ algorithm كل مرة بحل المشكلة باستخدام نفس الـ data).
- أفكار المشاريع التي بها الـ Two-Player Games أو الـ puzzles لا تحتاج إلى Dataset لأنه لا يوجد بها الـ training phase لأنها تصنف ضمن الـ AI ولكن ليس بها ML كما أوضحت سابقاً.
- بالنسبة لأفكار المشاريع التي تحتاج إلى Dataset، بإمكانكم استخدام أي dataset أخرى بدلاً من الـ dataset المقترحة في الـ projects' description.
- جميع الأفكار التي تحتاج إلى Dataset، عند مناقشة وتجربة المشروع، يتم ذلك باستخدام الـ testing instances من الـ dataset المستخدمة، والطلاب غير مطالبين باستخدام الـ data خارجية.
- جميع الأفكار التي تحتاج إلى evolution أو training، أو إلى testing، يجب عمل الـ plots توضح الـ evolution أو الـ training، وجدول يوضح نتائج الـ testing (ويفضل عن طريق الـ N Fold Cross Validation كما أوضحت بالمحاضرات).

- أفكار المشاريع التي بها تطبيق طريقتين أو أكثر (Genetic Algorithm AND the Backtracking Algorithm)، يجب حل المشكلة باستخدام كل طريقة منهما بشكل منفصل، ثم المقارنة بين كفاءة كل منهما في سرعة الوصول للحل وجودة الحل.

- أفكار المشاريع التي بها تطبيق Decision Trees & Random Forests، يجب حل المشكلة أولاً باستخدام Decision Tree واحدة فقط، ثم باستخدام Random Forest، فكل طريقة منهما تطبق بشكل منفصل، ثم يتم المقارنة بين كفاءة كل منهما في سرعة الوصول للحل وجودة الحل.

- أفكار المشاريع التي بها Object Detection، المطلوب تصنيف صور بها Object واحد فقط بكل صورة، وليس تصنيف أكثر من Object من صورة واحدة.

- أفكار المشاريع التي بها تطبيق an Artificial Neural Network، يجب حل المشكلة باستخدام ANN تقليدية، ثم إذا رغب الفريق في تطبيق CNN كحل إضافي فبإمكانه عمل ذلك بعد تطبيق المطلوب.

### أسئلة بخصوص يوم المناقشة:

- يجب طباعة report وتسليمه كـ Hard Copy عليه أسماء كل أعضاء الفريق، ورقم كل طالب، والمستوى الدراسي والقسم/الشعبة لكل عضو.
- يجب في report وضع رابط لـ shared folder عليه كل code والـ report (بـ Shortened URL) .. ويجب أن يكون folder متاح حتى موعد ظهور النتائج لنتمكن من الرجوع إليه في أي وقت.

**N.B.; A documentation (report) should accompany the project. Provide clear and concise (yet, comprehensive) documentation explaining the functionality, algorithms used, and instructions for users to interact with the application. The report should explain the AI algorithms implemented, any design choices, the design rationale behind heuristic functions (if utilized), the rationale behind the values of any parameters/hyper-parameters in the employed algorithms. The documentation should serve as a guide for users and developers to understand the project's inner workings.**

نموذج يمكن إتباعه عند كتابة report:

#### Introduction and Overview:

- Project idea and overview.
- Applications (*desktop, web, or mobile*) similar to the one you're developing, and what are the functionalities/features, and how they work (*if that information is available*).
- A Literature Review of Academic publications (*papers/books/articles*) relevant to the problem you're trying to solve and the approach you're trying to implement (*at least 5 resources*). You may find them by searching using Google Scholar.

#### Proposed Solution & Dataset:

- Main functionalities/features (*from the users' perspective*) in your proposed software/solution (*can be explained using a use-case diagram*).
- If applicable to your project, the Dataset employed (*preferably a publicly available dataset*).

### Applied Algorithms:

- All the details of the AI/Machine-Learning algorithm(s)/approach(es) used to develop your project (*can be explained using block diagrams*).

### Experiments & Results:

- The Experiments, testing, and the results (*including plots of the evolution or training if applicable*) and samples of the output (*and how did you test the solution*).

### Analysis, Discussion, and Future Work:

- Analysis of the results, what are the insights?
- What are the advantages / disadvantages?
- Why did the algorithm behave in such a way? What might be the future modifications you'd like to try when solving this problem?