

# ASUS | Copilot Starter Pack

This document outlines topics, tools, and general tips and tricks that participants are encouraged to familiarise themselves with before the competition, should they choose to do this bounty.

This document does not describe the bounty. You're going to have to wait till the event for that 😊.

## **Difficulty Level**

Difficulty: Accessible / Beginner–Intermediate

This bounty is designed to be approachable for participants with a basic understanding of programming concepts.

## **Recommended Team Composition**

Teams with at least one member comfortable with programming are recommended. Multidisciplinary collaboration will be especially valuable for system design, user experience, and presentation. A great code solution that solves the wrong business problem will not win.

## **Bounty Prizes**

- 1st place: 3000 aed
- 2nd place: ROG Bag Pack x 4 + Surprise
- 3rd place: ASUS Goodie Bag + 50 aed eshop voucher x 4

## **Topics Encouraged to Look At**

### **1. Programming Foundations**

Participants should be comfortable with:

- Web Development Basics
- Basic syntax and control flow.
- Functions, classes, and objects.
- Error handling and debugging.
- Familiarity with the Microsoft Ecosystem and API Integration may be beneficial.

Example languages include:

- Java  
Power Fx

## 2. Microsoft Ecosystem & Tooling

Pro Tip: This challenge strictly requires Microsoft tools. Familiarity with the specific capabilities of these platforms is essential. Participants are encouraged to explore the catalogue of Microsoft Cloud services to avoid building things from scratch.

Resources:

- [Microsoft Learn: Azure AI Services Documentation](#)
- [Microsoft Power Platform](#)
- [GitHub: Azure Samples for Students](#)

## 3. Presenting & Pitching solutions:

Participants are encouraged to think about how to clearly communicate their solution to both technical and non-technical audiences.

Useful areas to explore include:

- Clearly defining the problem being addressed
- Explaining why the solution matters and who it helps
- Demonstrating system behavior through simple workflows
- Communicating trade-offs, limitations, and future extensions

Strong presentations often focus on clarity and value, rather than implementation details alone.

Resources:

- [Y Combinator: How to Pitch Your Product](#)
- [Guy Kawasaki: The Only 10 Slides You Need in a Pitch](#)

## General Tips and Tricks

- AI-assisted development tools can be useful for ideation, exploration, and implementation.
- If using AI-powered tools that have access to your repository, always use plan mode and reiterate the plan before you let the agent execute the commands.
- If you have a team from beforehand, set up the repository and look into git commands in case you are using version control.

<https://education.github.com/git-cheat-sheet-education.pdf>

- Look into best practices for documenting APIs if you connect to external tools
- Try to solve a real-world problem in an innovative way.

## **Submission Format**

- A working webpage link or demo.
- 3–5 slides of a presentation explaining the problem, solution, features, and AI usage.