Microsoft Sponsored Hack

Build an awesome Windows app!

What's the prize?

Surface Go for each member of the winning team (up to 4)

Overview:

Microsoft has been revamping Windows to make it more attractive to developers. Especially with the new release of Windows 11, we have revealed some very exciting news both for Windows users and developers! Windows is now THE open platform for developers. This means everyone has open access to run the widest range of apps on Windows, enabling all developers to build the apps and experiences that empower users to create, learn, and play.

We are embracing all your apps and are working to make all apps feel right at home on Windows. Here are a few examples of the type of apps you can build and leverage the fastest developer OS.

- If you are a member of the ever-growing web developer community, you can now build a PWA from your web app in minutes with the new PWABuilder3. This will allow you to build a cross platform app based on web standards and capabilities. You can leverage any web framework of your choice! Not sure where to get started? Check out pwa-starter repo today to build your first PWA! With the new Chromium based Edge, PWABuilder and the new Microsoft Store in Windows 11 you can now build Progressive Web Apps that not only run in the web browser on any device but can also be shipped to the Microsoft Store!
- Wanting to make your app cross platform utilizing frameworks like React Native, Electron, Flutter, etc.? They are awesome if you want to support multiple platforms. You write once, and with the magic of a command line, build your code to run on multiple platforms, like Android, iOS and MacOS and Windows.
- Have you heard of Project Reunion? We now offer the Windows App SDK that will make it easier for you to integrate Windows native features into your apps while still enabling you to reach more than 1B users on Windows. There are also other types of native Windows apps including UWP, Win32, WPF, WinForms, etc., which all target only Windows and provide a unified look and feel.
- If you participated in HackOHI/O 2020, you may remember our corporate challenge on best app that utilizes Microsoft Graph, which is the gateway to data and intelligence in Microsoft 365. We now offer open-source tools for working with authentication and Microsoft Graph via the Windows Community Toolkit, to help you integrate rich productivity data into your native applications.

Commented [JW1]: I would also love to mention the starter here pwa-builder/pwa-starter: Welcome to the PWABuilder pwa-starter! Looking to build a new Progressive Web App and not sure where to get started? This is what you are looking for! (github.com) along with the new Edge. If users just go to the PWABuilder site they are not going to know where to get started building a PWA. Maybe we could change this paragraph to something like `Are you a web developer and want to build a cross-platform app that also runs natively on Windows? With the new Chromium based Edge, PWABuilder and the new Microsoft Store in Windows 11 you can now build Progressive Web Apps that not only run in the web browser on any device but can also be shipped to the Microsoft Store!

Commented [BP2R1]: I updated the paragraph. Let me know what you think.

How will projects be judged:

- **Functionality:** Participating projects will build a functioning solution that runs natively on Windows platform. This means the solution could be in one of many forms, and written with various languages and frameworks: a Universal Windows app, a Progressive Web app, an app build with cross platform frameworks, such as a Flutter app, a React Native app, a Xamarin app, etc.
- **Originality:** Solutions should be authentic that address emerging problems with creative and original functionalities and designs.
- **Technicality:** Includes how well the idea was executed by the team and developers, including the user experience, the complexity of the scenarios, ability to run on Windows platform natively, and whether it performs as expected.
- **Teamwork:** Projects indicate students solve the problems as a team and share the required work equally.
- **Bonus**: Those solutions that incorporate the use of Microsoft platforms (such as Microsoft Teams, Visual Studio Code, or GitHub) or Microsoft services (Azure services, Microsoft Graph APIs, etc.) will get extra credit in judging!

How to get started?

****Note: You can find all the samples, links to blogs, and links to video tutorials mentioned in this section here: https://aka.ms/win-dev/student/samples**

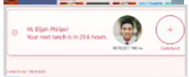
1. Decide what type of app you would like to build
    a. Join us on October 20th, Wednesday, to learn the types of apps you can build on and for Windows; what are the benefits of each type of app; what are the tools that provide acceleration to your app development; how to set up your development environment, and more.
    b. Here is a recording you can checkout!
2. Set up your development environment and get started
    a. UWP:
        i. Step by step blog: https://aka.ms/ win-dev/student/osu/uwp/get-started-blog
        ii. https://docs.microsoft.com/visualstudio/get-started/csharp/tutorial-uwp
        iii. Install tools for Windows app development - Windows apps | Microsoft Docs
    b. PWA
        i. PWA Starter wiki: Home · pwa-builder/pwa-starter Wiki (github.com)
        ii. Video tutorial: https://aka.ms/ win-dev/student/osu/pwa/get-started-video
        iii. Step by step blog: https://aka.ms/ win-dev/student/osu/pwa/get-started-blog
    c. Flutter
        i. Video tutorial: https://aka.ms/win-dev/student/osu/flutter/get-started-video
        ii. Step by step: https://aka.ms/ win-dev/student/osu/flutter/get-started-blog
3. Add your functionalities that make your app unique!
    a. Add WinML through Intelligent APIs to your app
        i. Video tutorial: https://aka.ms/win-dev/student/osu/winml/get-started-video
        ii. Step by step blog: https://aka.ms/win-dev/student/osu/winml/get-started-blog
    b. Add Windows Community Toolkit to make your app look like a native Windows application!
        i. Video tutorial: https://aka.ms/ win-dev/student/osu/uwp-wct/get-started-video
    c. Connect to Microsoft Graph and utilize the rich productivity data and intelligence.
        i. If you are building a UWP application, follow instructions in this article on how to easily incorporate authentication to reach Microsoft Graph data.
        ii. If you are building a PWA, checkout Microsoft Graph Toolkit to add Microsoft Graph authentication and reach Microsoft Graph data.

Commented [MH3]: Should we leave this to the articles below on each tech? As Web is going to go VS Code path and the UWP/WinUI 3 would use Visual Studio?

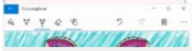
Commented [BP4R3]: I am tempted to link to Alex's WinUI 3 blog for setup. I think for UWP we can just link to the tutorial-uwp link?

Sample app gallery

BuildCast Video Player UWP Sample: Showcases the Microsoft Fluent Design System and UWP capabilities, including ink notes, bookmarks and remote playback.



Lunch Scheduler: See integration with services like Facebook, Microsoft Graph, Bing maps, and Yelp.

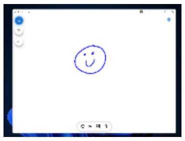


Coloring Book: Demonstrates the versatility of Windows Ink and radial controller features.

UWP Customer Order Database sample: Demonstrates a working database hosted on Azure, implemented in C# and XAML.

Ambie: A modern Windows app that plays soothing tones to help you be healthier, happier, and more relaxed.

Commented [MH5]: Ambie would be a good one too: jenius-apps/ambie: The best white noise app on Windows (github.com)



Webboard: A cross-platform Whiteboarding application (Progressive Web App) that ties in with the latest Edge and Windows 11 features to enable low-latency inking and other advanced capabilities that are now possible for web applications. Webboard runs on any device with a modern web browser and can be installed from the Microsoft Store and, for Android devices, the Google Play Store.