



INTRO TO .NET MAUI

CROSS- PLATFORM DEVELOPMENT WITH .NET



TEAM INTRODUCTION

BSEF21M005 HAIQA AROOB

BSEF21M018 BUSHRA ARSHAD

BSEF21M052 ANEESA NAZIR



WHAT IS MAUI?

.NET MAUI: (MULTI-PLATFORM APP UI) IS A CROSS-PLATFORM FRAMEWORK FOR BUILDING NATIVE MOBILE AND DESKTOP APPLICATIONS USING C# AND XAML.

EVOLUTION: SUCCESSOR TO XAMARIN.FORMS, PROVIDING A MORE STREAMLINED DEVELOPMENT EXPERIENCE.

PURPOSE: WRITE ONCE, RUN ANYWHERE - TARGET ANDROID, IOS, WINDOWS, MACOS FROM A SINGLE CODEBASE.

FEATURES OF MAUI

SINGLE PROJECT STRUCTURE: UNIFIED PROJECT FOR MULTIPLE PLATFORMS.

-HOT RELOAD: INSTANT UI UPDATES WHILE CODING.

-NATIVE PERFORMANCE: ACCESS TO PLATFORM-SPECIFIC APIs.

-RICH UI CONTROLS: BUILT-IN CONTROLS AND LAYOUTS (E.G., BUTTON, LABEL, GRID).

-MVVM SUPPORT: BUILT-IN SUPPORT FOR MODEL-VIEW-VIEWMODEL PATTERN.



.NET MAUI ARCHITECTURE

Single Project Structure:

.NET MAUI INTRODUCES A SINGLE PROJECT STRUCTURE WHICH CONSOLIDATES ALL THE PLATFORM-SPECIFIC PROJECTS (LIKE ANDROID, IOS, WINDOWS, MACOS) INTO A SINGLE .NET MAUI PROJECT. INSTEAD OF MANAGING SEPARATE PROJECTS FOR EACH PLATFORM, DEVELOPERS WORK WITHIN A UNIFIED PROJECT STRUCTURE.

Benefits:

- **SIMPLIFIES PROJECT SETUP AND MAINTENANCE.**
- **REDUCES THE COMPLEXITY OF MANAGING MULTIPLE PROJECTS FOR DIFFERENT PLATAFORMS.**
- **PROVIDES A CENTRALIZED LOCATION FOR MANAGING CODE, RESOURCES, AND DEPENDENCIES.**

.NET MAUI ARCHITECTURE

.NET 6 and C# Support:

.NET MAUI IS BUILT ON TOP OF .NET 6, THE LATEST LONG-TERM SUPPORT (LTS) VERSION OF THE .NET PLATFORM, AND SUPPORTS THE LATEST FEATURES OF THE C# PROGRAMMING LANGUAGE.

Shared UI and Resources:

.NET MAUI ALLOWS DEVELOPERS TO SHARE UI COMPONENTS, RESOURCES (LIKE IMAGES, FONTS, STYLES), AND CODE ACROSS MULTIPLE PLATFORMS. THIS SHARED CODE IS TYPICALLY WRITTEN IN XAML FOR UI AND C# FOR THE BUSINESS LOGIC.

.NET MAUI ARCHITECTURE

Platform-Specific Code and Customization:

WHILE MUCH OF THE CODE CAN BE SHARED ACROSS PLATFORMS, .NET MAUI ALSO PROVIDES MECHANISMS TO INCLUDE PLATFORM-SPECIFIC CODE AND CUSTOMIZATIONS WHEN NECESSARY.

Example:

ACCESSING THE DEVICE'S LOCATION, HOWEVER, REQUIRES PLATFORM-SPECIFIC IMPLEMENTATION BECAUSE THE APIs TO GET LOCATION DATA ARE DIFFERENT FOR ANDROID AND IOS.

.NET MAUI ARCHITECTURE

MVU (Model-View-Update) Architecture Support:

IN ADDITION TO SUPPORTING THE TRADITIONAL MVVM (MODEL-VIEW-VIEWMODEL) ARCHITECTURE, .NET MAUI ALSO SUPPORTS THE MVU (MODEL-VIEW-UPDATE) ARCHITECTURE, WHICH IS INSPIRED BY FUNCTIONAL PROGRAMMING PARADIGMS.

MVU PATTERN:

- **MODEL:** REPRESENTS THE STATE OF THE APPLICATION.
- **VIEW:** THE UI THAT IS RENDERED BASED ON THE CURRENT STATE (MODEL).
- **UPDATE:** A FUNCTION THAT TAKES THE CURRENT STATE AND A MESSAGE (USUALLY A USER ACTION) AND RETURNS A NEW STATE.

INSTALL MAUI

1 INSTALL

Visual Studio 2022 with .NET MAUI workload.

2 CREATE A NEW PROJECT

Use the .NET MAUI App template.

3 PROJECT STRUCTURE

- *MauiProgram.cs: App configuration.*
- *MainPage.xaml: UI layout.*
- *Platforms: Platform-specific code.*



BASIC UI EXAMPLE (C# CODE)

LET'S GO TO VISUAL STUDIO!



MVVM IN .NET MAUI

SEPARATES UI LOGIC FROM BUSINESS LOGIC.

ADVANTAGES

- EASIER TESTING AND MAINTENANCE
- CLEAN CODE SEPARATION.

- Example Components:
- Model: Holds data.
- ViewModel: Binds data to view.
- View: UI elements defined in XAML

PLATFORM-SPECIFIC FEATURES

ACCESS DEVICE CAPABILITIES: CAMERA, GPS, SENSORS, ETC.

-PLATFORM-SPECIFIC CODE EXAMPLE: VIBRATION ON
BUTTON CLICK

CODE (C#)

```
PRIVATE VOID ONBUTTONCLICKED(OBJECT SENDER, EVENTARGS E)
{
    VIBRATION.DEFAULT.VIBRATE(TIMESPAN.FROMSECONDS(1));
}
```



1

DEPLOYMENT AND TESTING

Testing: Use emulators/simulators and physical devices.

-Deploying: Publish to Google Play, Apple App Store, Microsoft Store.

-Tools: Visual Studio 2022, .NET CLI.

2

ADVANTAGES OF .NET MAUI

Cross-Platform Development: Reduce time and cost.

-Rich Ecosystem: Leverage existing .NET libraries.

-Consistent Performance: Native-like performance and appearance.



CONCLUSION AND FUTURE OUTLOOK

.NET MAUI

(Multi-platform App UI) is a cross-platform framework for building mobile applications using C# and XAML. A powerful framework for cross-platform development.

FUTURE ENHANCEMENT

Continuous updates, improved performance, and broader API support.

COMMUNITY

Strong Microsoft support and community-driven enhancements.



THANK YOU !

