

# CMPS 3390 Homework 2

*Spring 2026*

## Web Development

- [HTML/CSS/JavaScript](#): Core technologies for building the structure, style, and interactivity of web pages.
- [React.js](#): A popular library for building user interfaces, particularly single-page applications (SPAs).
- [Angular](#): An older, but robust fully featured framework for building dynamic web applications.
- [Vue.js](#): A progressive framework for building user interfaces with a flexible and simple approach.
- [Svelte](#): A compiler that converts declarative components into highly efficient JavaScript code.
- [Bootstrap](#): A CSS/Javascript framework that simplifies responsive web design with prebuilt components and layouts.
- [Sass](#): A mature, stable, and powerful professional grade CSS extension language.
- [Tailwind CSS](#): A utility-first CSS framework that allows you to build custom designs without writing CSS..

## Mobile Development

### 1. Mobile Native Development

- [Swift, SwiftUI](#) (iOS): The primary programming language for developing iOS applications.
- [Objective-C](#) (iOS): The older language for iOS development, still used in many legacy applications.
- [Kotlin](#) (Android): The preferred programming language for Android development.
- [Java](#) (Android): The original programming language for Android development, still widely used.

### 2. Cross-Platform Development

- [Flutter](#) (Dart): A UI toolkit for building natively compiled applications for mobile, web, and desktop from a single codebase.
- [React Native](#) (JavaScript): A framework for building native mobile apps using React.
- [Ionic](#) (JavaScript): A UI framework for building cross-platform mobile apps using web technologies.
- [.Net Maui](#) (C#): A framework for building cross-platform mobile applications using the .NET ecosystem.
- [NativeScript](#) (Javascript): A framework that allows you to build Android and iOS mobile apps using JavaScript by simplifying access to native mobile APIs
- [Kivy](#) / [BeeWare](#) (Python): open-source frameworks that allow developers to build cross-platform applications using Python

## Desktop UI Development

- [Electron](#) (JavaScript): A framework for packaging cross-platform desktop apps with web technologies like HTML, CSS, and JavaScript.
- [Qt](#) (Various): A framework for developing cross-platform applications and UIs.
- [JavaFX](#) (Java): A platform for building rich desktop applications with Java. (Gluon Scene Builder)
- [Swing](#) (Java): Java native library for developing desktop applications with Java. (IntelliJ Builder)
- [GTK](#) (C/various): A toolkit for creating graphical user interfaces, primarily used with Linux applications.
- [WPF \(Windows Presentation Foundation\)](#) (C#): A UI framework for building visually stunning Windows desktop applications.

## Game Development

- [Unity](#) (C#): A widely-used game engine for developing 2D and 3D games across various platforms.
- [Unreal Engine](#) (C++/Blueprints): A powerful game engine with tools for creating high-quality 3D games.
- [Godot](#) (GDScript/C++/C#): An open-source game engine for 2D and 3D game development.
- [Pygame](#) (Python): A set of Python modules designed for writing video games, ideal for beginners.
- [LibGDX](#) (Java): A framework for building cross-platform games in Java.
- [LÖVE](#) (Lua): A free, open-source, cross-platform framework you can use to make 2D games in LUA.

## Additional Resources

- [Native App Development](#)
- [Cross-Platform vs Native Development](#)
- [Is Native App Development Dying?](#)