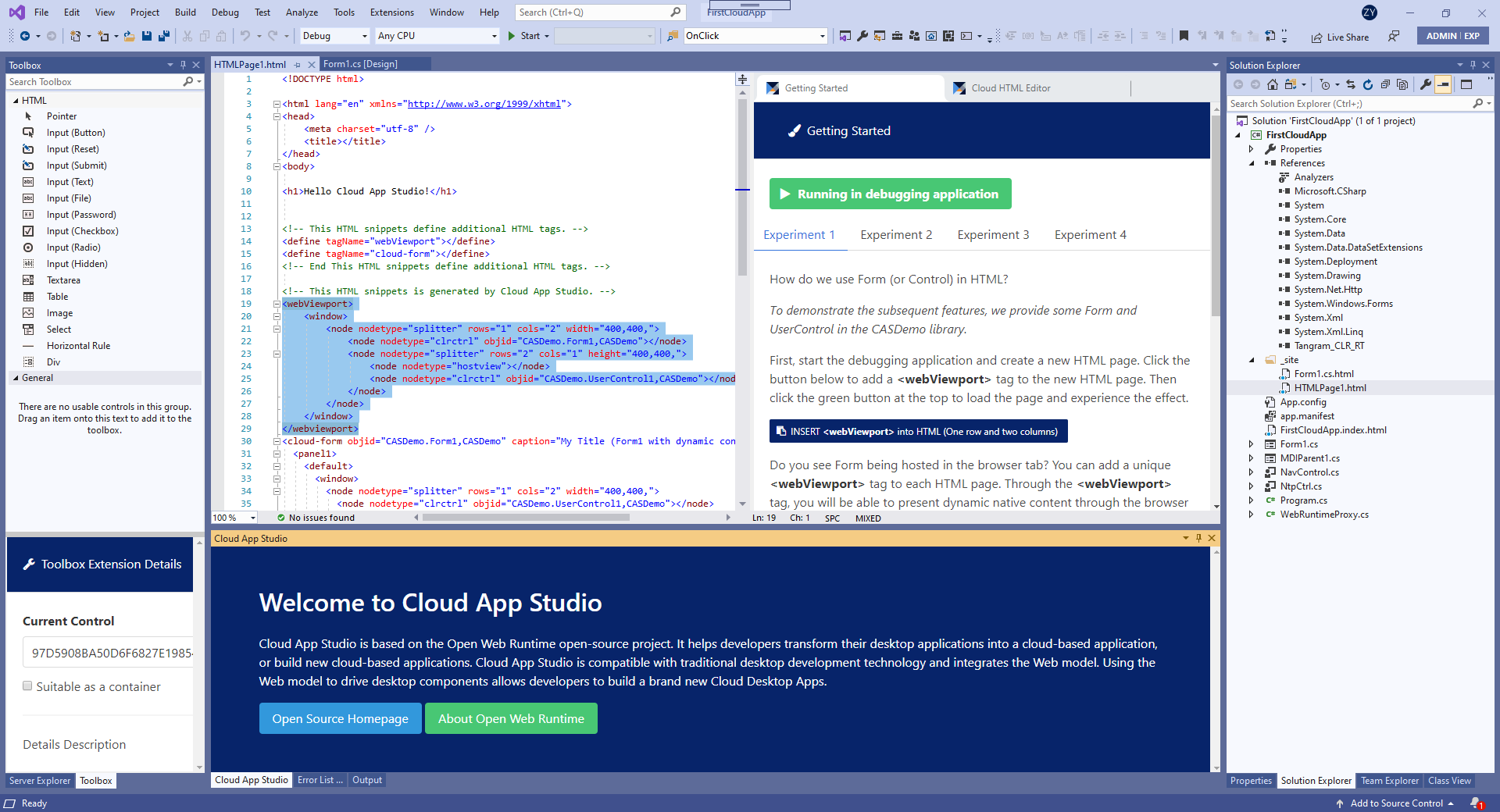
Cloud App Studio: Open the Cloud Era of Desktop Software Development

Tangram Team

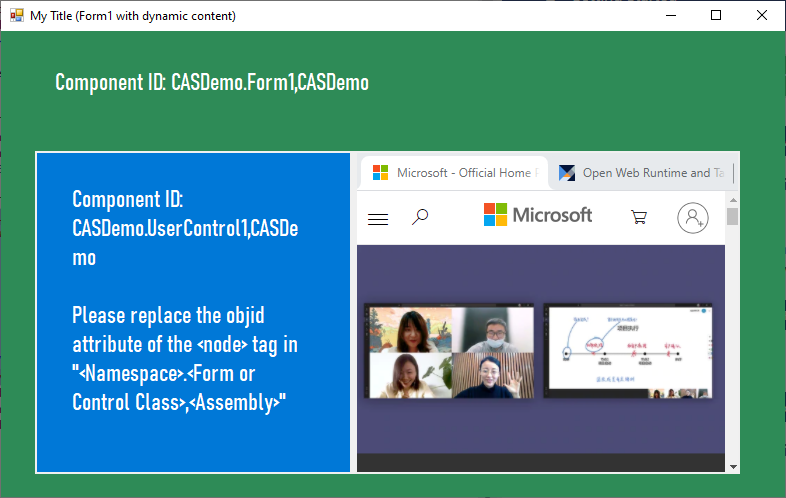
# Cloud App Studio

Cloud App Studio redefines the development model of desktop software with modern Web technology, providing desktop software with a common way to upgrade from "**software functions limited to local computers**" to "**provide software functions based on Internet cloud services**." For .NET developers, this assumption means that desktop software is transformed from software that uses .NET components, such as forms, controls, and WPF pages, to a software organization that is based on **Web pages**.

**Web pages** are the fundamental basis for Cloud App Studio to describe software content. At present, there are two main solutions for desktop software development technology related to Web technology. One is the Electron-based solution, which focuses on Web technology and provides limited support for Native components. The second is the CEF-based solution. The solution is based on controls. It only provides a Web viewport for desktop development and gives the interoperability between Native components and Web technologies. However, it also discards Chrome's "sandbox" isolation mechanism, **which is not suitable for large Scope Internet model**. Cloud App Studio contains the complete Chromium open source project code. Unlike the above two solutions, Cloud App Studio retains all the features of Chromium and provides more powerful Native component support. At the same time, Chromium's Sandbox mechanism is retained to **maintain a balance between Web technology and desktop technology**. Cloud App Studio allows the target software to have its own Web page system. This is an extended DOM system that includes the standard Web page model. In this system, developers can write more flexible and powerful Web pages according to standard Web page technology. As shown in the following figure, the working interface of Cloud App Studio is based on this technology:

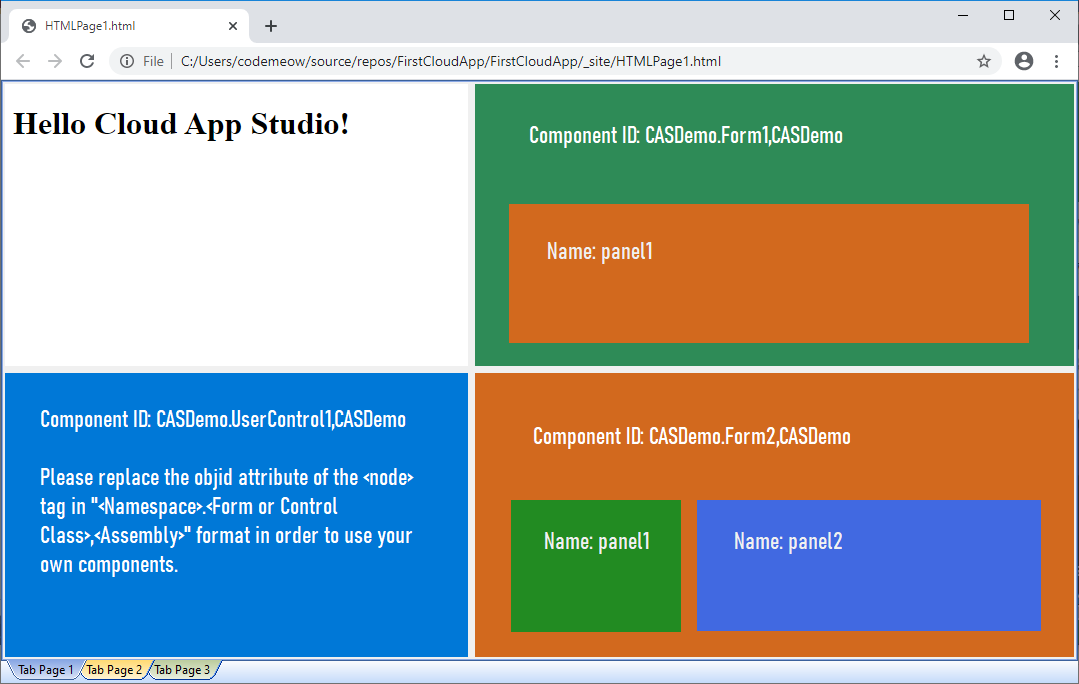


Of course, developers can embed the page into a user form, as shown in the following figure:



(A "multi-tab page" is displayed in a user form)

Developers can also open an application page directly in the application:

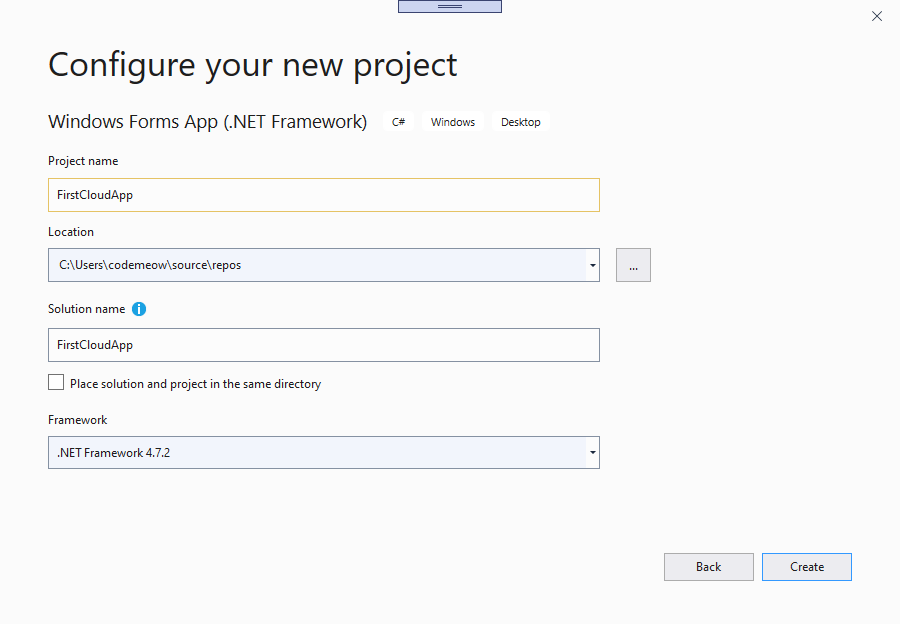


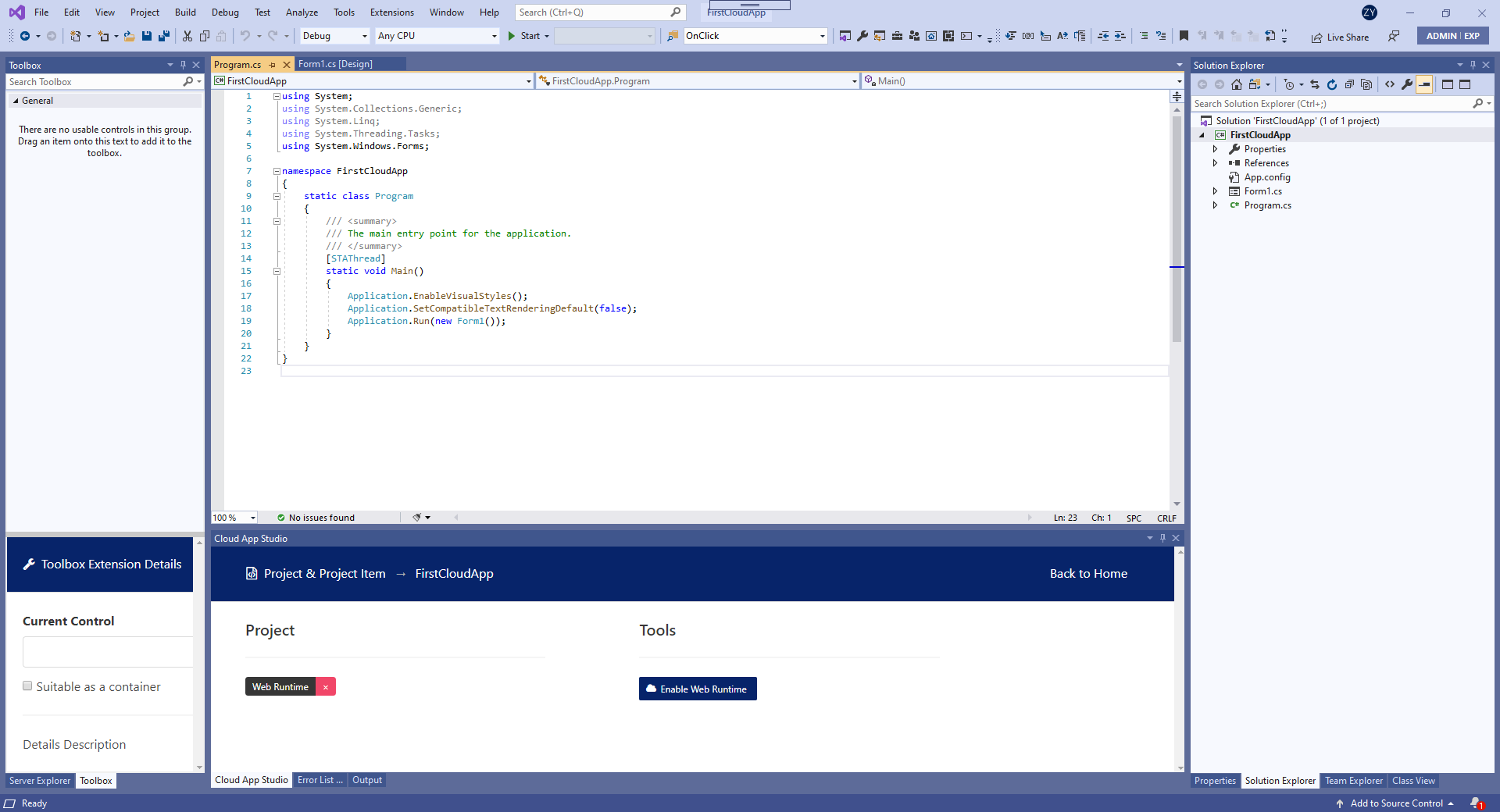
Unlike popular web-based desktop software development models like CEF and Electron, Cloud App Studio allows you to use Native Components naturally and treat them as "**new DOM elements**" in web pages. Because it contains all the Chromium code, **each application software built on the basis of Cloud App Studio is naturally a modern browser.** The difference from the general browser is that such applications include more full application layer support.

# Web page in Cloud App Studio concept

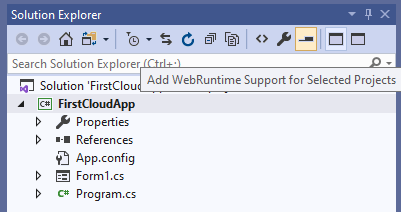
Let's see how web pages work on the Cloud App. To do this, you need to download and install [Cloud App Studio](https://marketplace.visualstudio.com/items?itemName=TangramDev.Web-Runtime-for-Visual-Studio), which is a Visual Studio plug-in. After installing Cloud App Studio correctly, we can start to experience Cloud App Studio.

In the first step, you need to create a Windows Forms App (.NET Framework) Project.





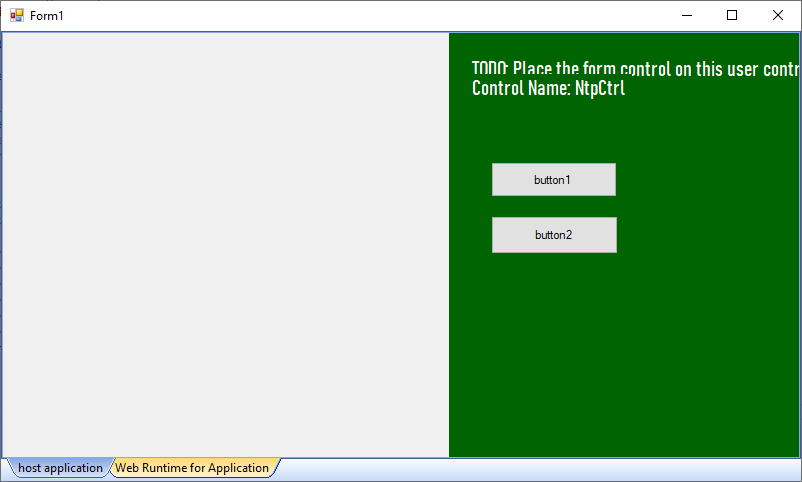
The second step is to find the following buttons on the Solution Explorer window:



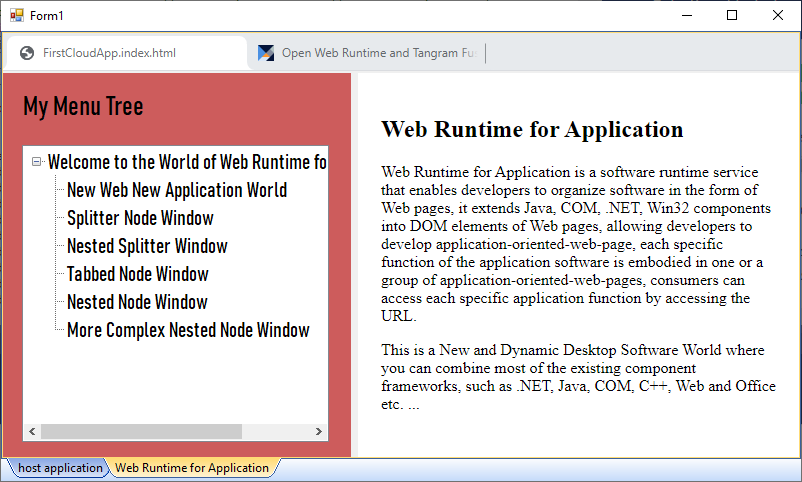
Enable WebRuntime to Selected Projects

Click this button, Cloud App Studio will add the necessary code and reference to your newly created project, after this step, your project will have a complete Cloud application foundation.

The third step is to press F5 to start debugging, a new experience journey begins... When the program starts, we first see the following form:



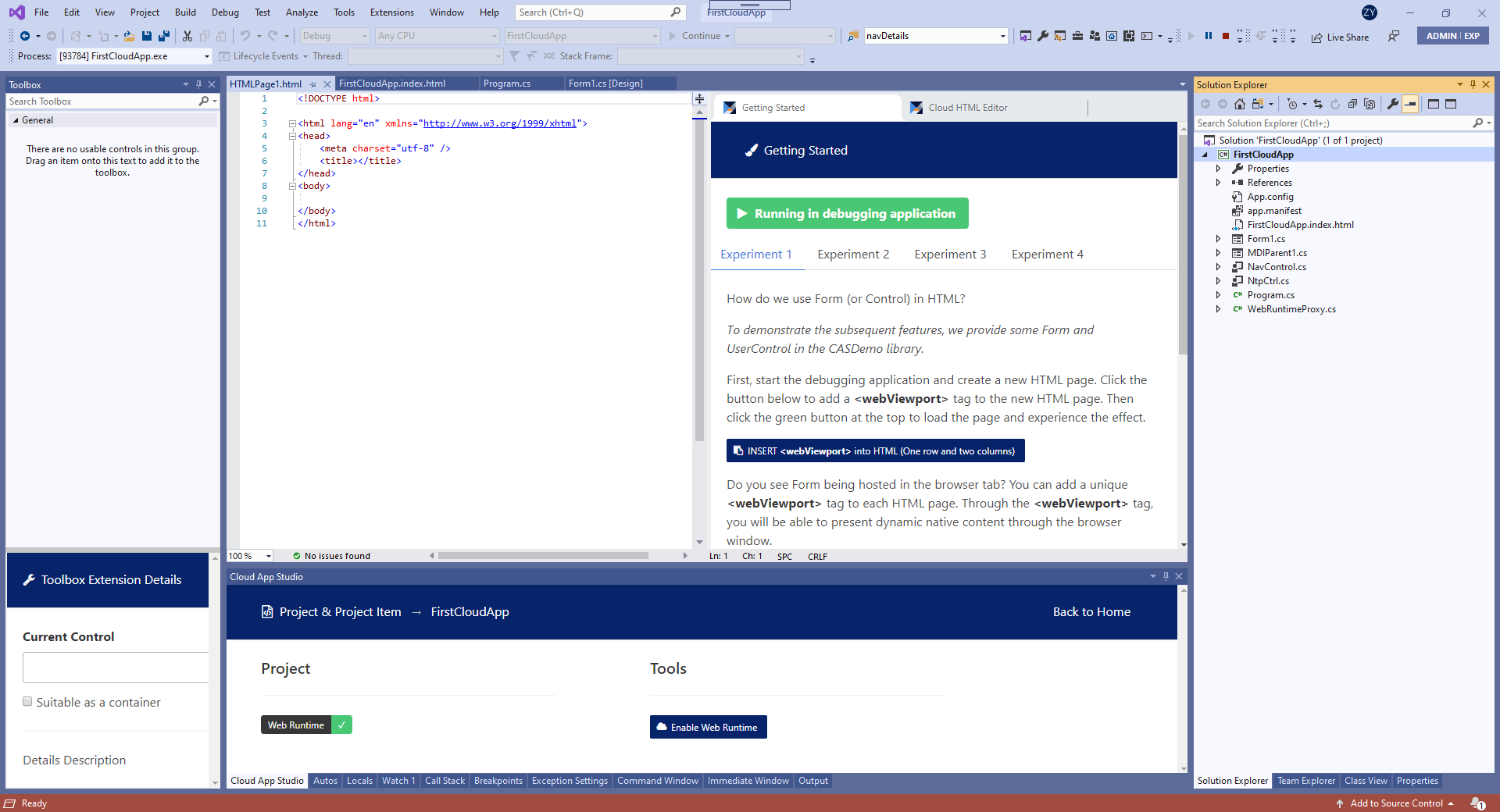
Switch the tab below, we see the following screen:



We see some familiar objects in it. Yes, it is the familiar Chrome browser structure. The difference is that Chrome is a component here, and its internal pages contain new elements that are completely different from standard HTML elements. The page you see is an interface composed of .NET components and HTML. If you use the mouse to operate the tree control on the left, you will see more performances that make you different. **All of this you can hardly see the participation of C# code. Where are they? What dominates all this?**

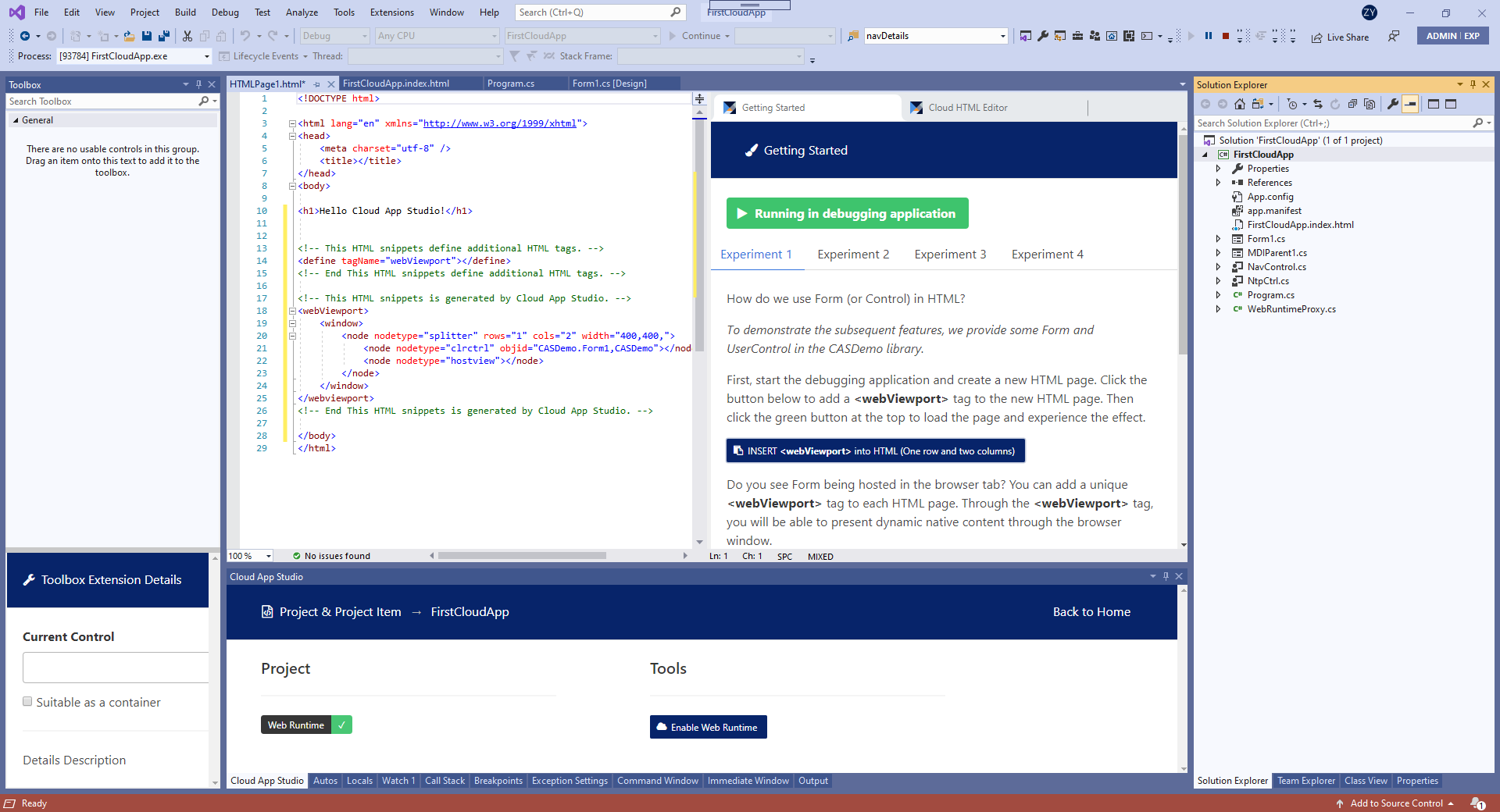
Don't close the running program, you can see an HTML page in the project: FirstCloudApp.index.html, which is the page that controls everything you just saw. **Next, let's see how a Web page organizes various Native component elements.**

The fourth step is to create a new html page in the Visual Studio IDE, as shown in the figure:

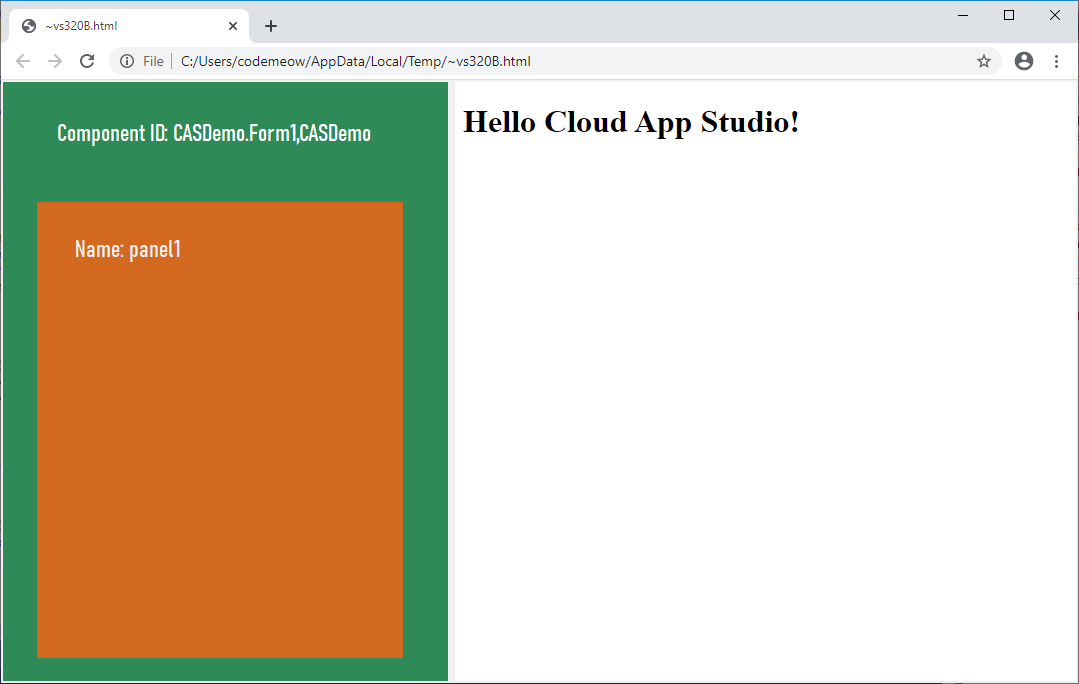


Click here to insert webViewport tag

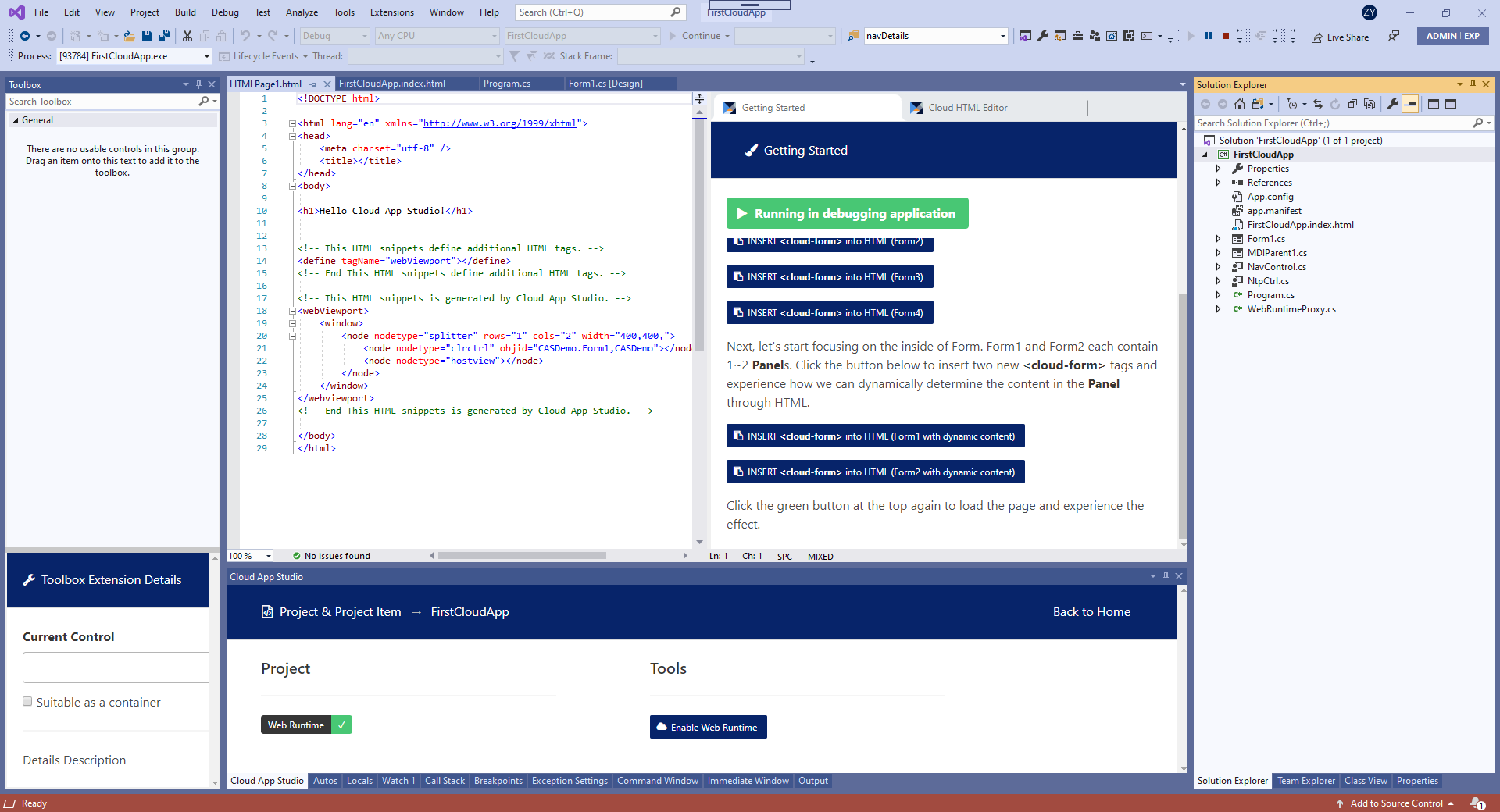
So, how does a Form object become a page element? To do this, you need to insert a "webViewport" tag in the page. Press the button pointed to in the figure and we see the following changes on the page:



Click to Run

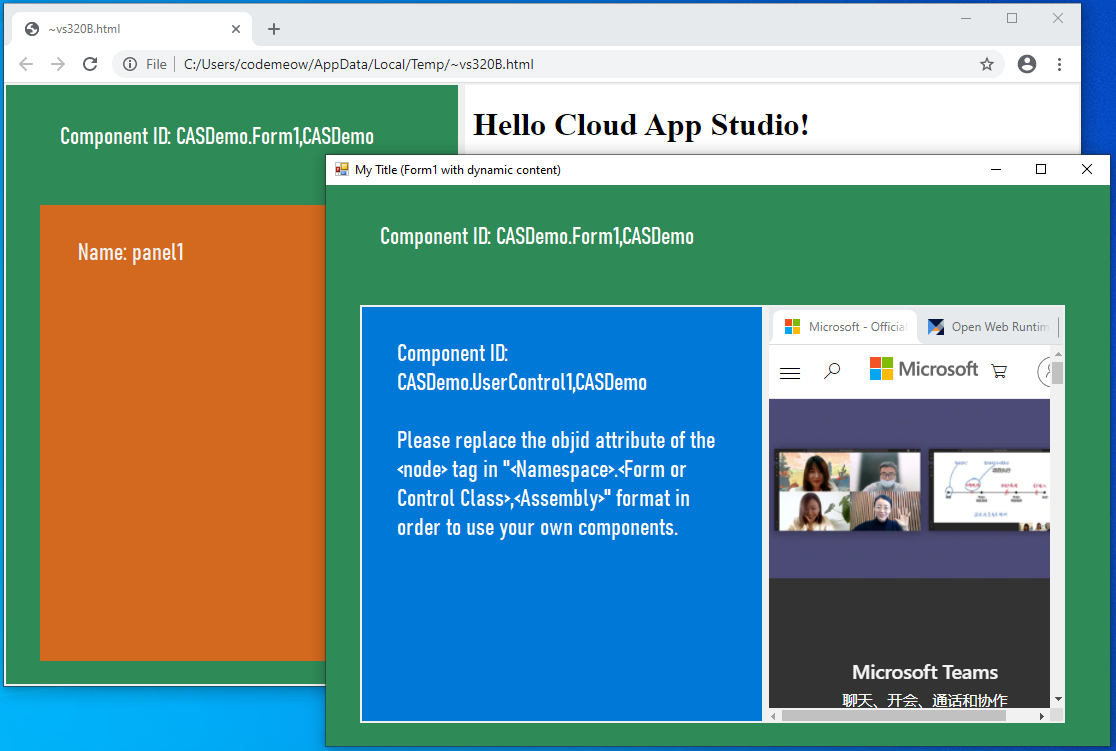
Click the green button to run, the following picture is the effect when the page is running:  


We provide developers with various experience experiments, which are included in the auxiliary panel of the HTML editor of Visual Studio. For example, the effect of experiment 2 is as follows:



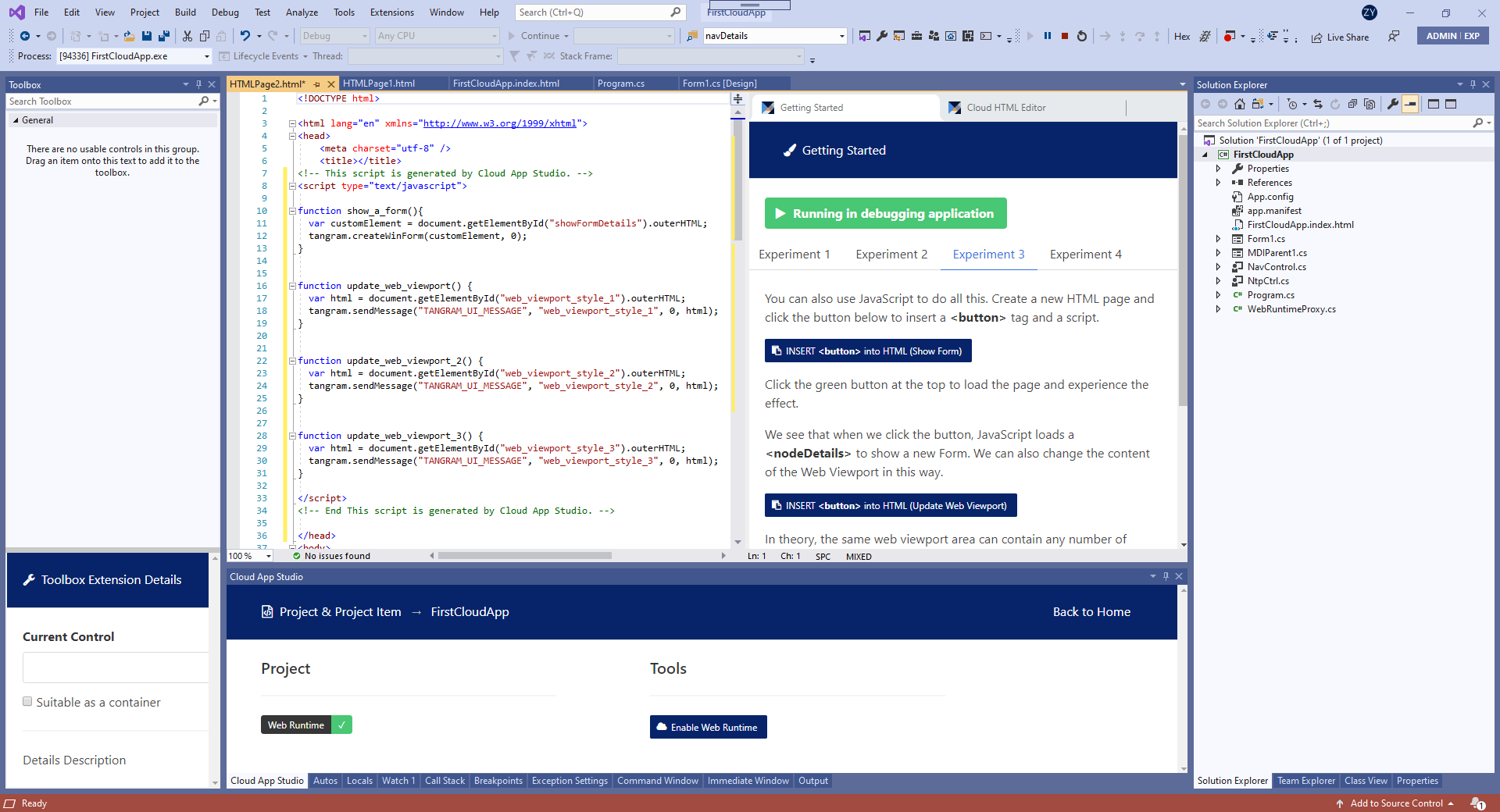
Click here to insert more complex structure in the page

The following is a runtime screenshot of this page:

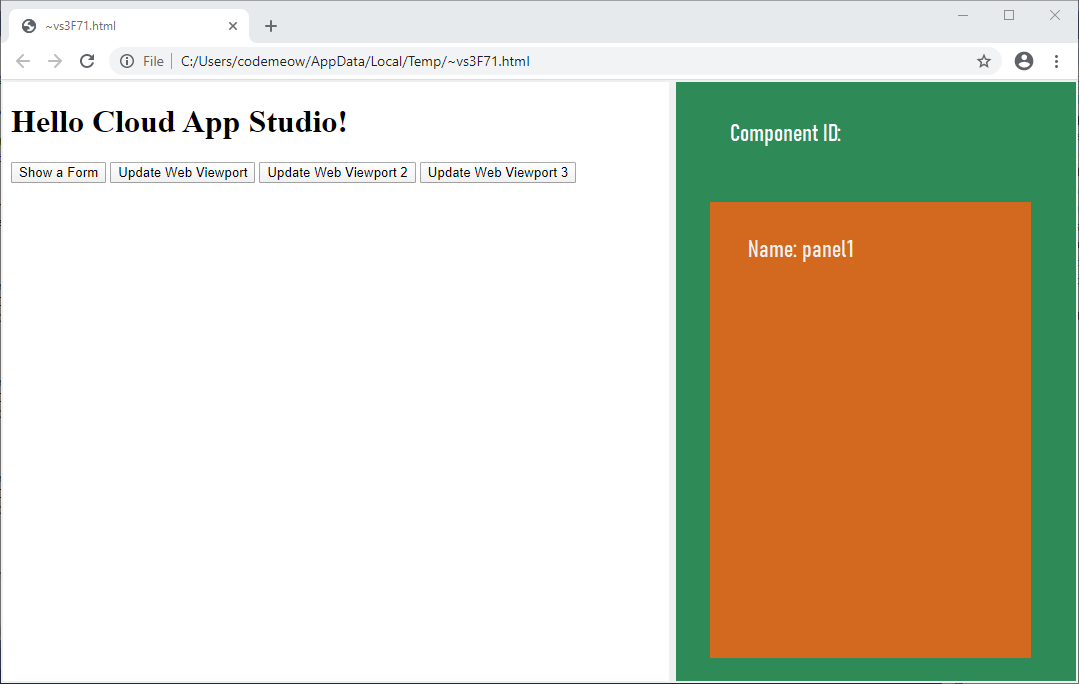


As shown in the figure, this page show a Form, which contains a set of pages.

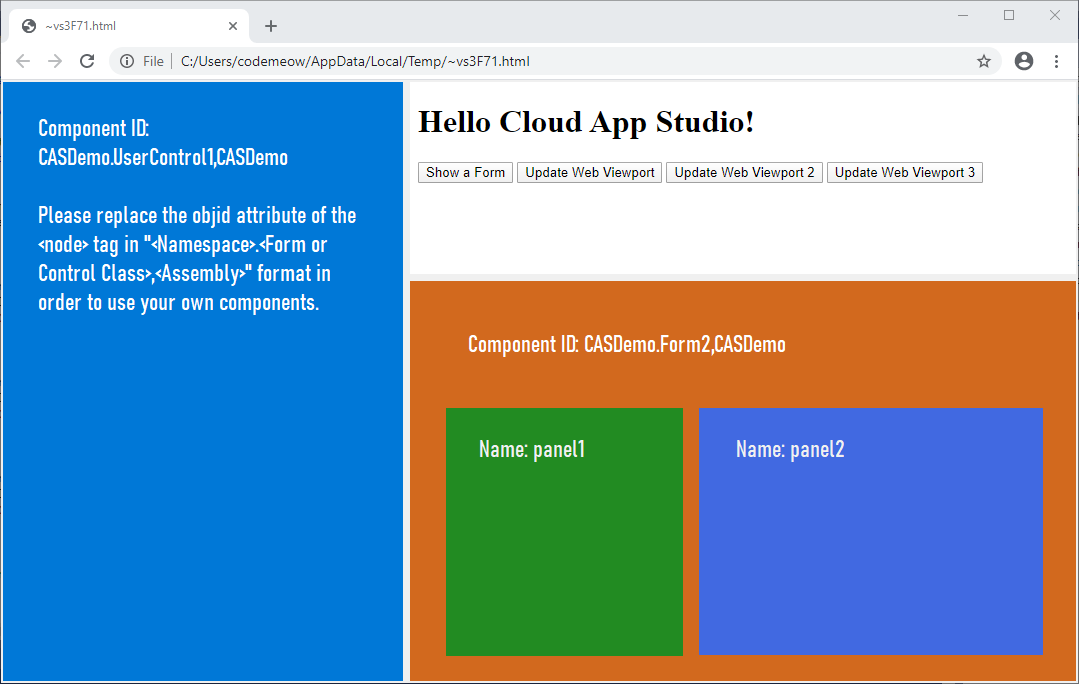
For more page experiments, you need to have an in-depth experience through Cloud App Studio in the Visual Studio IDE. Below, we will introduce a relatively comprehensive page through experiment three. Follow the instructions to insert a set of html buttons, the page structure is as follows:



The runtime effect of this page is as follows:



(Press the second button)



(Press the third button)

By clicking different buttons, different webViewport "layers" will appear, and the switching of the multi-layer structure is used to meet the application of high-complexity structures.

# When everything is a Web page element, the application world will be a brand new Web world

Needless to say, Cloud App Studio will create a brand new web world. In this world, there is almost no difference between Native components and standard DOM elements. If there are differences, it is that Native components run in a sandbox isolated from standard Web pages. This ensures the security of the page. The involvement of the Native component in Web development will have a great impact on the existing Web model. Cloud App Studio ensures that the two can interoperate well without giving up the inherent advantages of the two.

**电脑屏幕截图

描述已自动生成**

(When more programmable elements become web page elements, a whole new web page world will follow)

Cloud App Studio enables developers to re-examine the structure of application software and the way of serving users from a Web point of view, which is the key to adapting desktop software to the development of the Internet. Because of the emergence of this type of technology, compilation-led solutions must transition to Web-led solutions.

# Cloud App Studio version

Cloud App Studio has an open source version and a commercial version. The difference between the two is that the commercial version contains some commercial components that are not suitable for open source and the necessary commercial services that the open source version does not have. The core code is the same.

We are committed to making Cloud App Studio a development framework. Since the project involves a very wide range and our team is very small, people from all walks of life are welcome to contribute to this project. For more details, please refer to our open source homepage.

Visual Studio Package download link:

<https://marketplace.visualstudio.com/items?itemName=TangramDev.Web-Runtime-for-Visual-Studio>

# Contact Us

Official Website：<https://www.tangram.dev>

Open Source Homepage：<https://github.com/TangramDev>

Email：[welcome@tangram.dev](mailto:welcome@tangram.dev)

WeChat：