Studio 2,3 and 4

Name and Student Id: Kaikun Song 33414564

Self-Evaluation {To be highlighted by Student only}:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Need Help | Work in Progress | Pass | Credit | Distinction | High Distinction |

Task 2

{Answers}

{Screenshots}

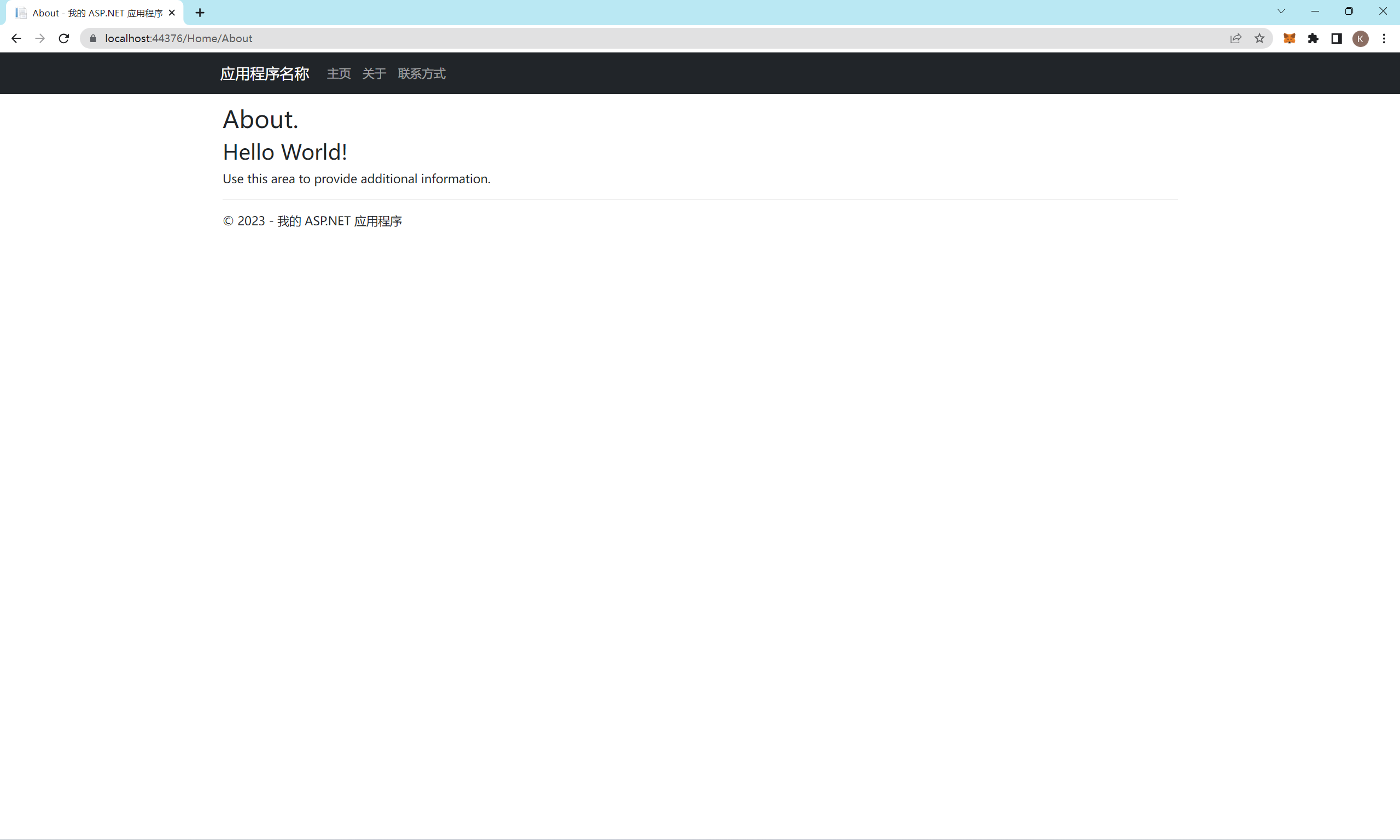


Fig1.HelloWorld screenshot about Step9

{Link to code repository}

[Kyikon/FIT5032\_Week2 at master (github.com)](https://github.com/Kyikon/FIT5032_Week2/tree/master)

{References}

/

{URLs}

/

Task 3

{Answers}

{Screenshots}

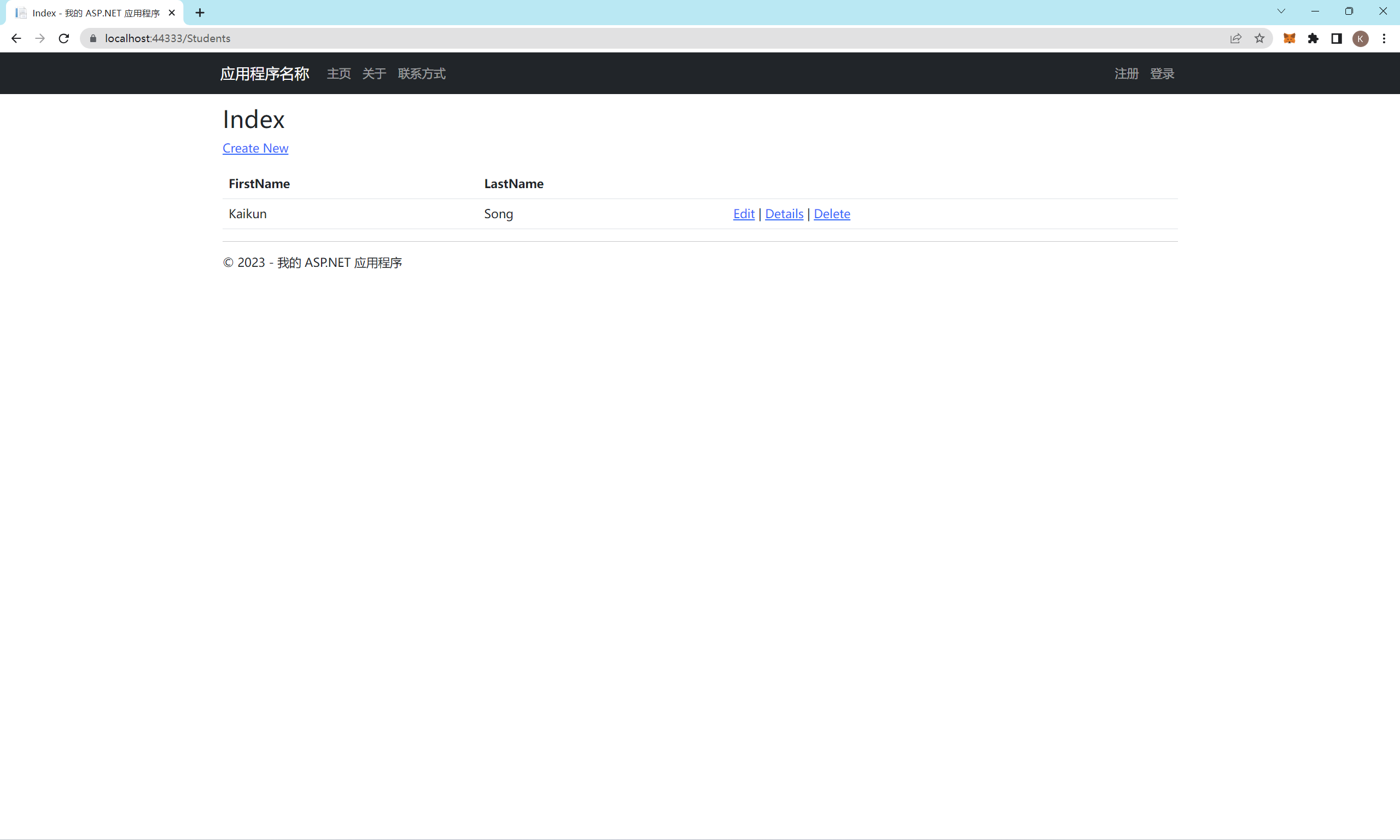


Fig2.Screenshot of running Web Application along with database schema(ModelFirst)

{Link to code repository}

[Kyikon/FIT5032\_Week3 (github.com)](https://github.com/Kyikon/FIT5032_Week3)

{References}

{URLs}

Task 4

{Answers}

The difference between "defer" and "async" lies in how they control the loading and execution order of scripts:

Defer: When a browser parses an HTML document and encounters a script tag with the "defer" attribute, it continues parsing the document and defers the execution of the deferred script until the document has finished parsing. Deferred scripts are executed in the order they appear in the document. The deferred scripts are executed before the DOMContentLoaded event is triggered.

Example scenario for using "defer":

When a script depends on the DOM structure but doesn't need to block the parsing and rendering of the document, you can mark the script as deferred. This can speed up the page load time.

When scripts need to be executed after the page has fully loaded, and the execution order of the scripts is important, you can use deferred scripts to ensure they execute in the desired order.

Async: When a browser parses an HTML document and encounters a script tag with the "async" attribute, it continues parsing the document and asynchronously fetches the script from the specified source. The script is executed as soon as it is available, without waiting for the HTML document to complete parsing. Asynchronous scripts can be executed out of order relative to other scripts or the DOMContentLoaded event.

Example scenario for using "async":

When a script doesn't depend on the DOM structure and can be executed independently, you can mark the script as async. This allows the browser to fetch and execute the script without blocking the parsing of the HTML document.

When multiple scripts are not dependent on each other and can be executed in any order, you can mark them as async to allow for parallel fetching and execution, potentially speeding up the overall loading time of the page.

{Screenshots}

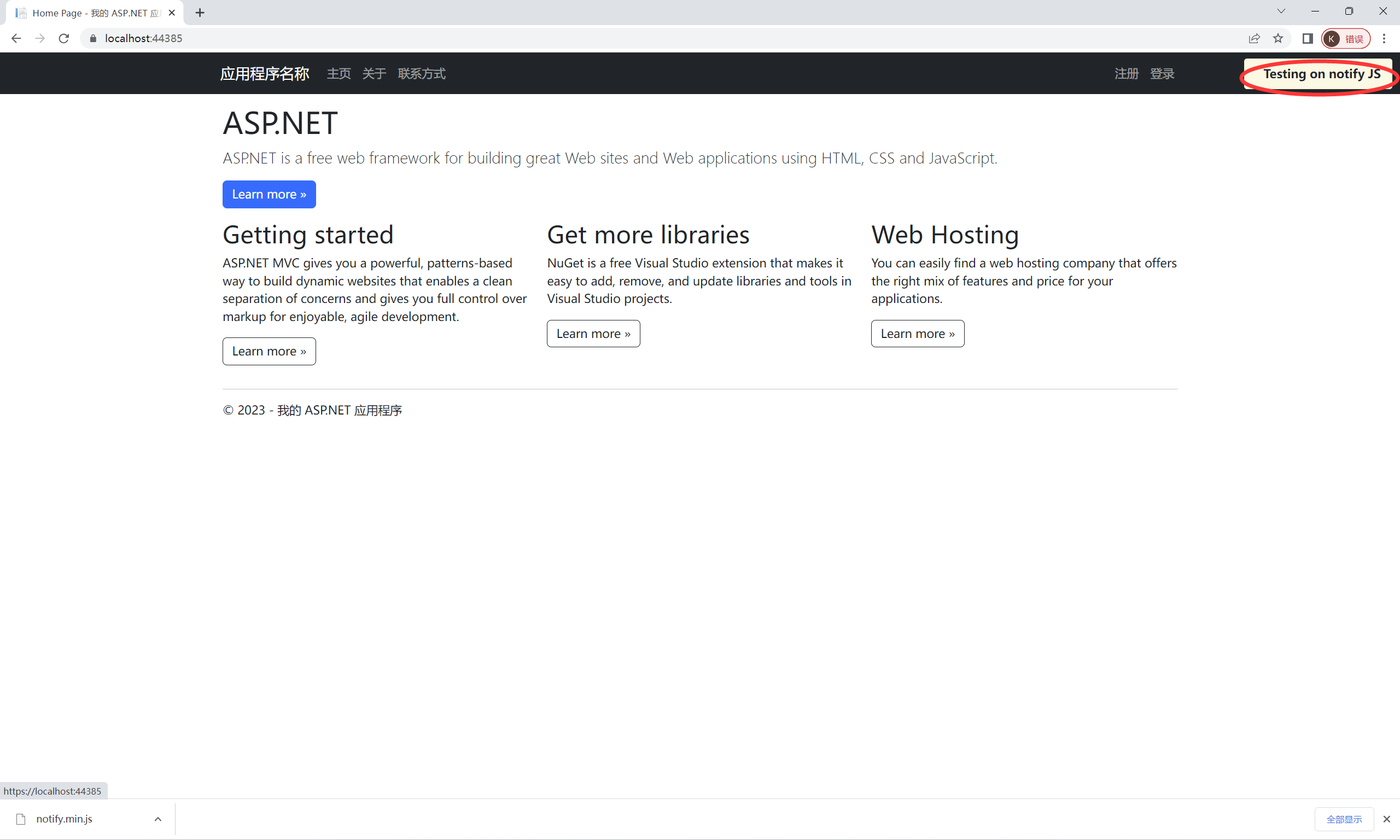


Fig3.Screenshot of website using Notify.js

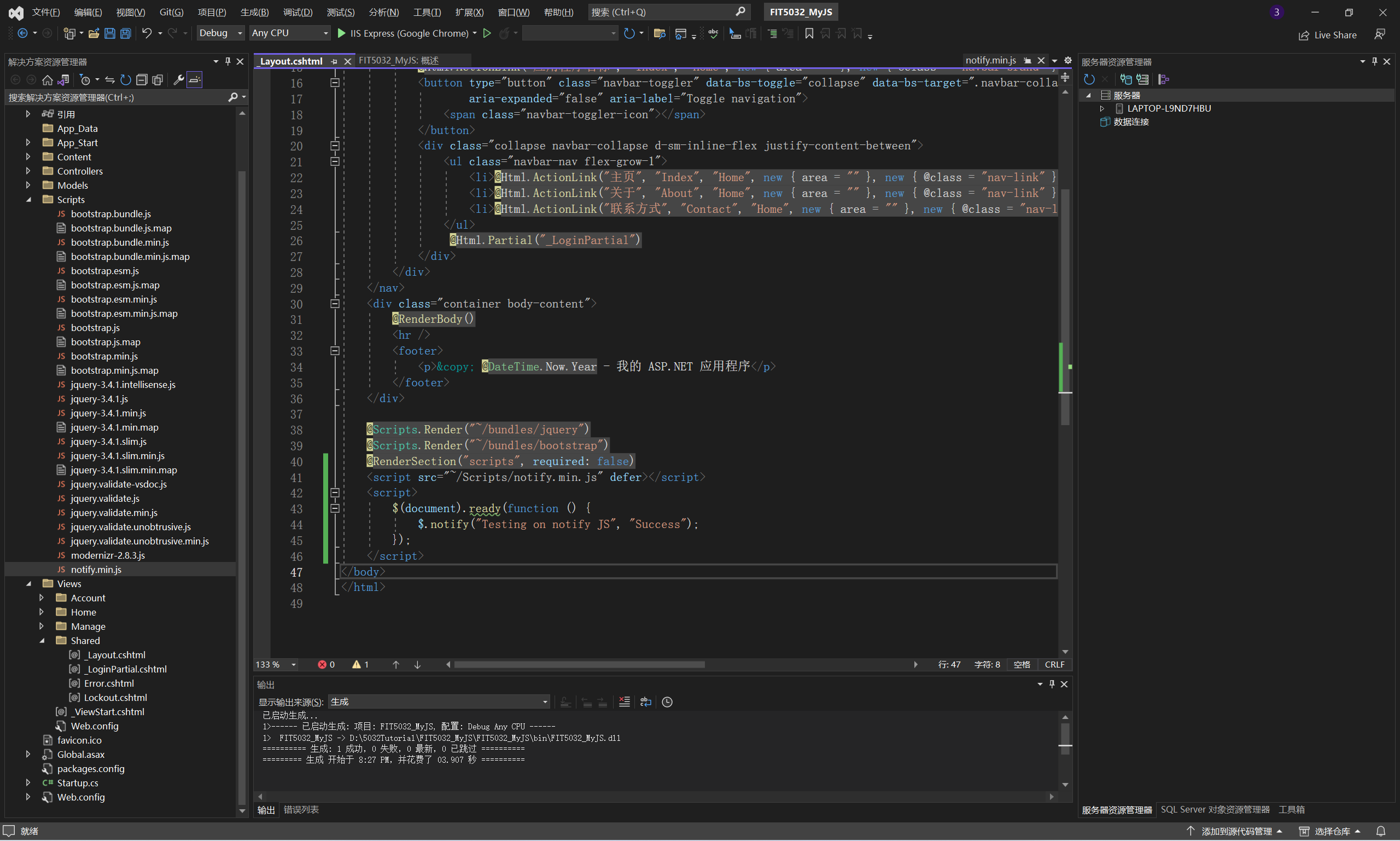


Fig4.Screenshot of code(\_Layout.cshtml)

{Link to code repository}

[Kyikon/FIT5032\_Week4 (github.com)](https://github.com/Kyikon/FIT5032_Week4)

{References}

{URLs}