**Final Assignment**

**Summer 2024**

**Course Title: Web Technology**

**Course Code:** **CSE480**

**Submitted by:**

**Name: Shakil Ahmed Shawon**

**ID: 201014081**

**Department of CSE**

**University of Liberal Arts Bangladesh (ULAB)**

**Ans. To The Question No. 01**

1. The main differences between ‘$\_GET’ and ‘$\_POST’ are:
2. ‘$\_GET’ is used to collect data sent through the URL parameters, while ‘$\_POST’ is used to collect data sent through HTTP POST method.
3. ‘$\_GET’ has a limitation on the amount of data that can be sent (usually 2048 characters), while ‘$\_POST’ has a higher limit.
4. ‘$\_GET’ requests are visible in the URL, making them less secure for sensitive data. while ‘$\_POST’ requests are not visible in the URL, making them more secure.

In case of a form submit, using ‘$\_POST’ is more secure because:

1. ‘$\_POST’ requests are not visible in the URL, so sensitive data like passwords is not exposed.
2. ‘$\_POST’ requests are more reliable for sending large amounts of data.
3. Most developers prefer using JSON over XML for data transmission due to the following reasons:
4. JSON is lighter and more compact compared to XML, resulting in faster transmission and parsing
5. JSON uses a simple syntax with key-value pairs and arrays, making it easier to read and write compared to XML.
6. JSON is natively supported by JavaScript, the primary language for web development, allowing for easy parsing and manipulation of data.
7. JSON is less verbose than XML, requiring fewer characters to represent the same data.
8. A loosely typed language, also known as a dynamically typed language, is a programming language that does not require explicit declaration of variable types. Some example of loosely typed language include:-
9. JavaScript: Variables ca hold values of any data type without explicit declaration.
10. Python: Variables are dynamically typed, and their type is determined at runtime.
11. PHP: Variables can be used to store values of different data types without explicit declaration.
12. Synchronous calls are blocking, meaning the program execution is halted until the called ffunction returns a result.

Asynchronous calls are non-blocking, allowing the program to continue executing other tasks while waiting for the called function to return.

Scenarios where synchronous calls fit better:

1. Simple operations that complete as the blocking behavior is not noticeable.
2. When the order of execution is crucial, and the program must wait for a result before proceeding.

Example: User authentication, where the application needs to verify for credentials before allowing access.

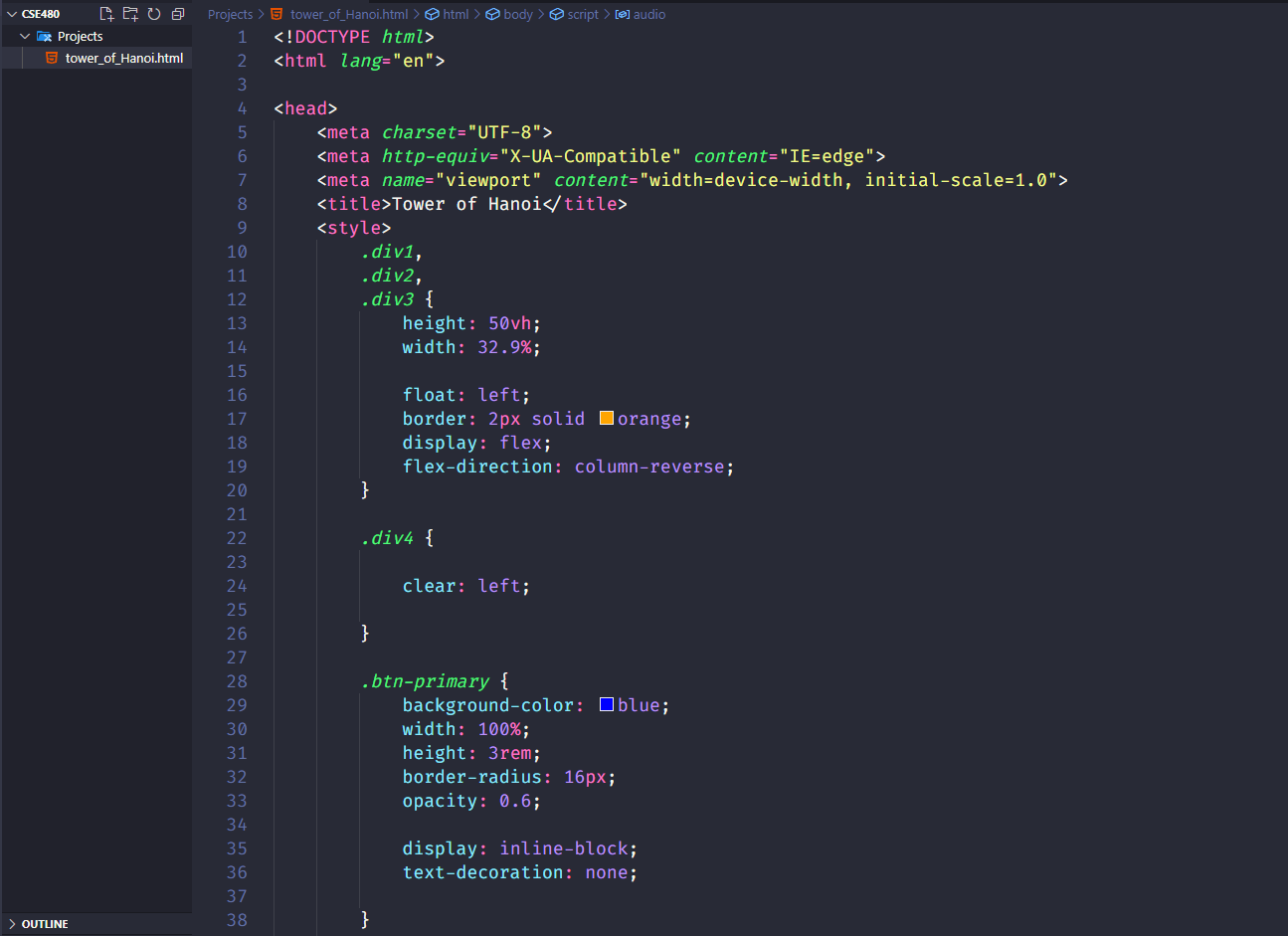
Scenarios where asynchronous calls fir better:

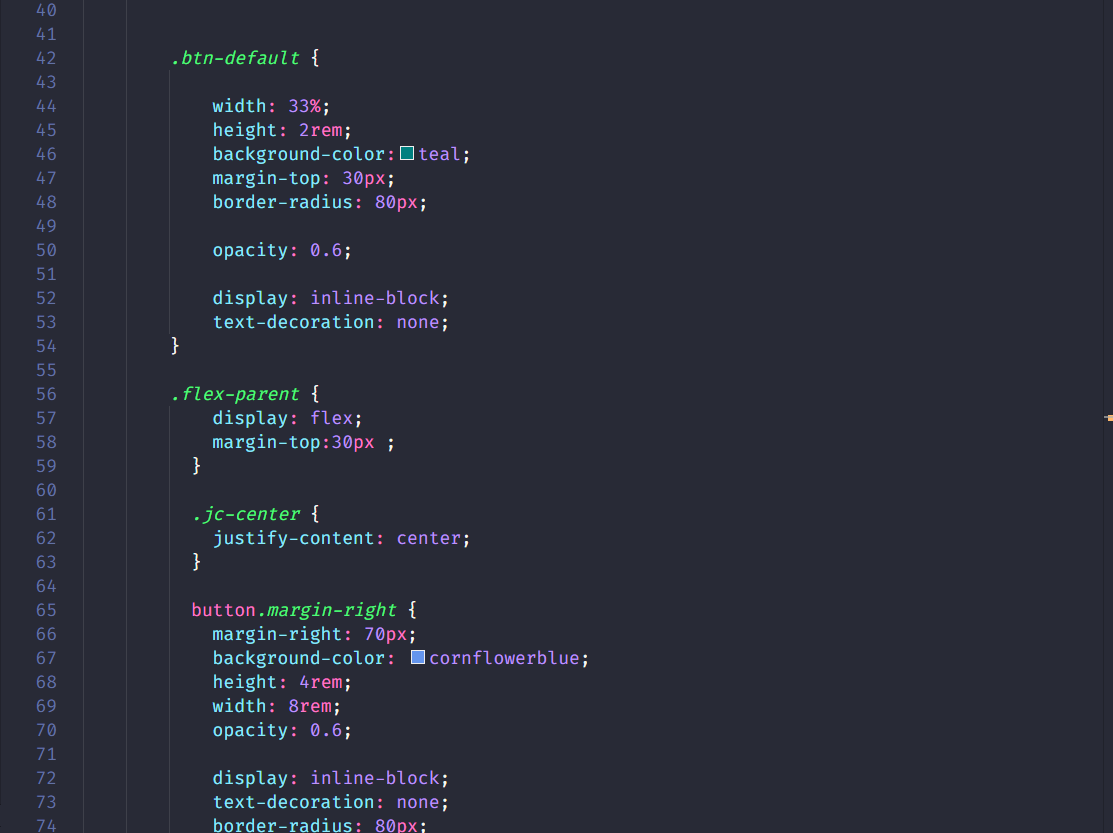
1. Long-running operations like network requests, file I/O, or database queries, to avoid blocking the main thread.
2. When the order of execution is not critical, and the program can continue with other tasks while waiting for the result.
3. In event-driven architectures and user interfaces to provide a responsive experience.

Example: Loading data from a server without blocking the UI.

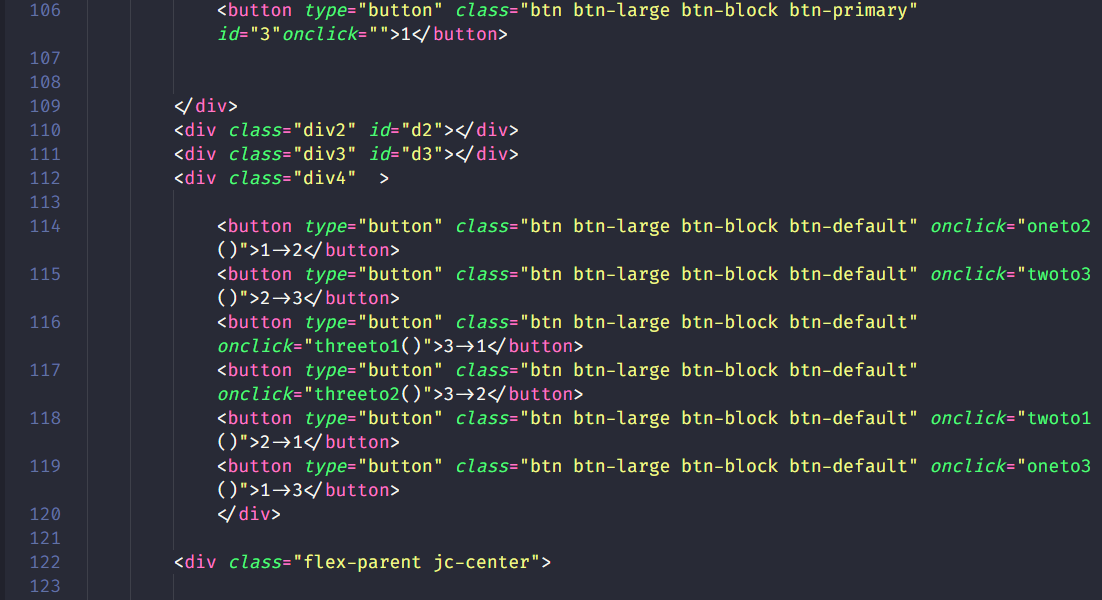
**Ans. To The Question No. 02**

**Screenshots of code:**



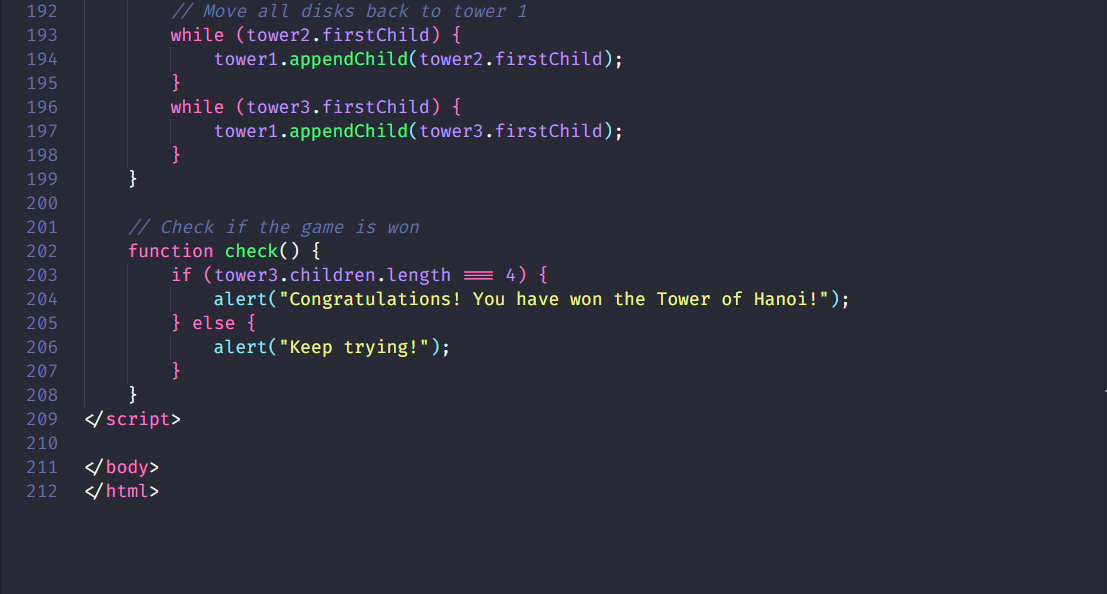












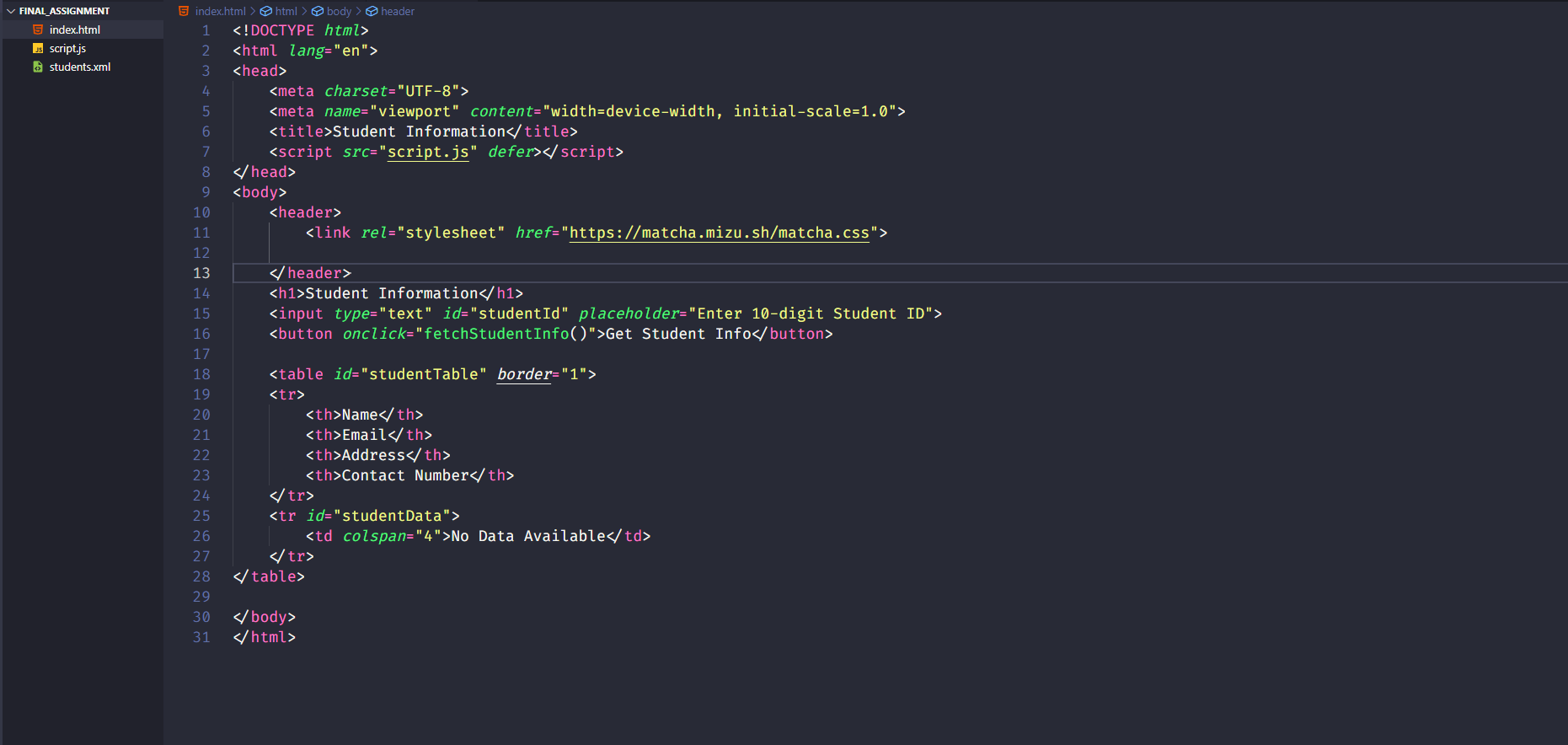
**Implementation Screenshots:**

****

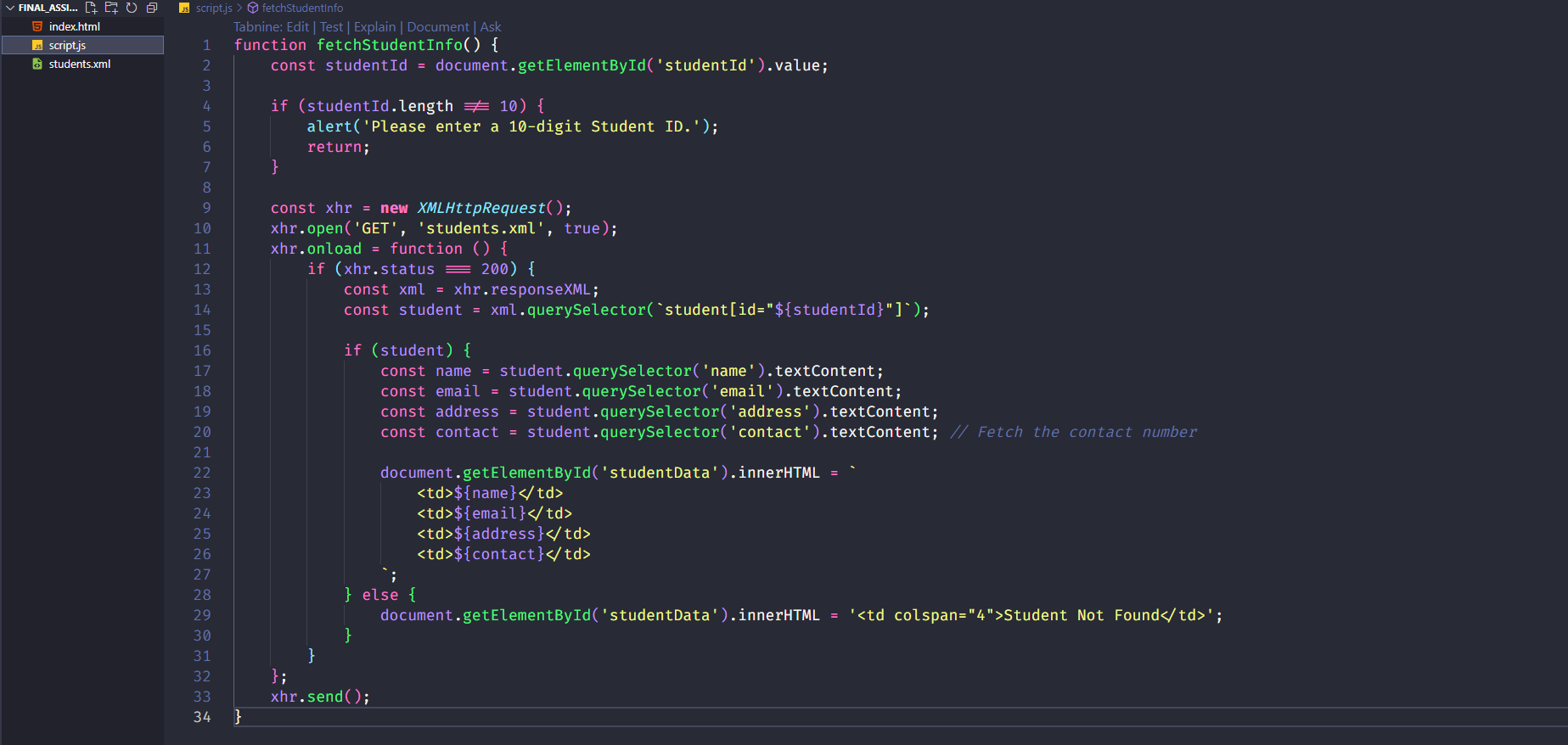
****

**Ans. To The Question No. 03**

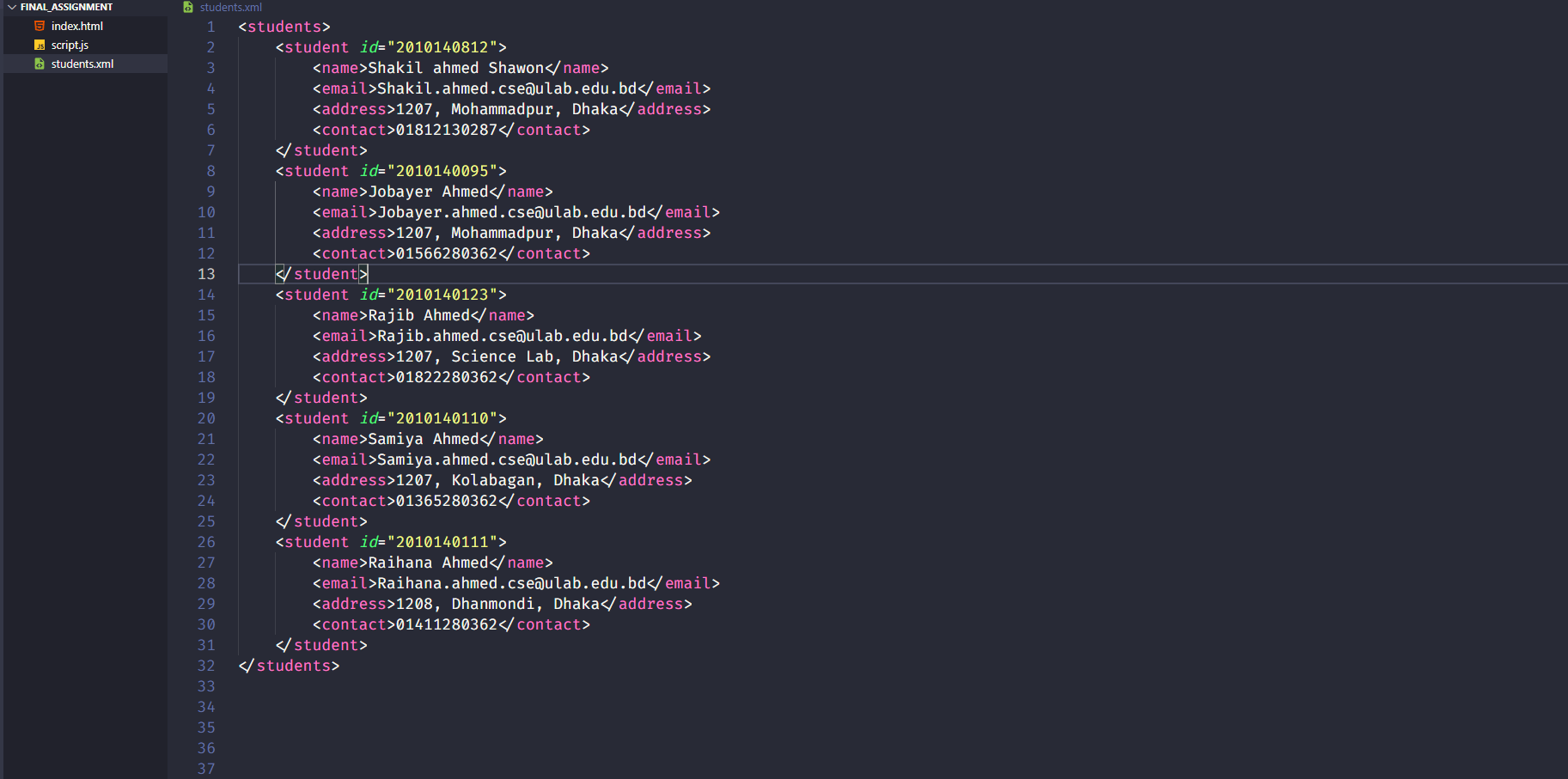
**Index.html code:**

****

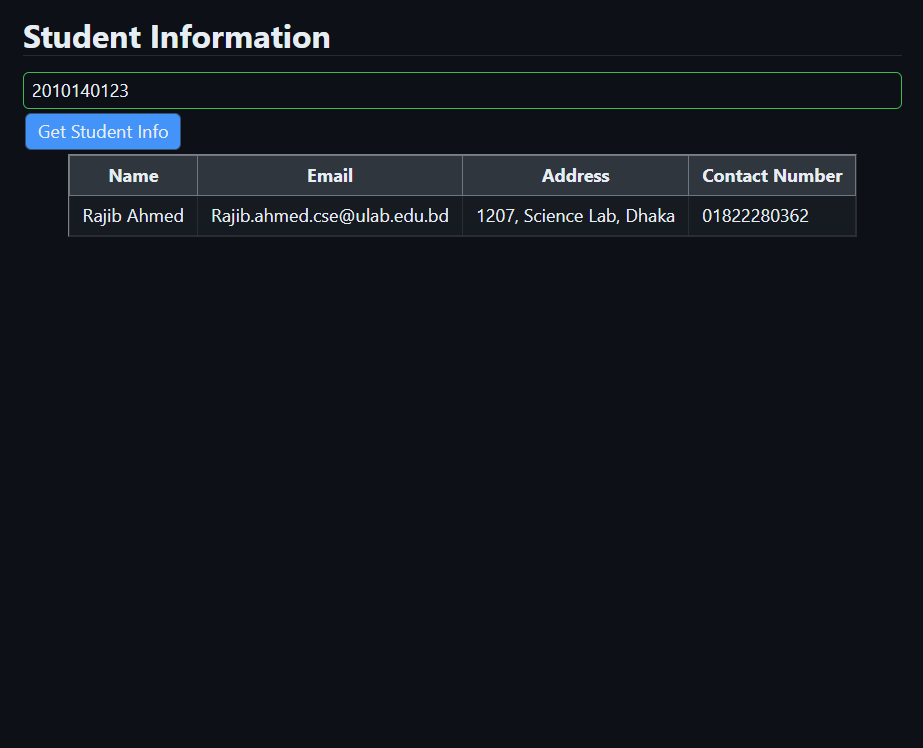
**Script.js code:**

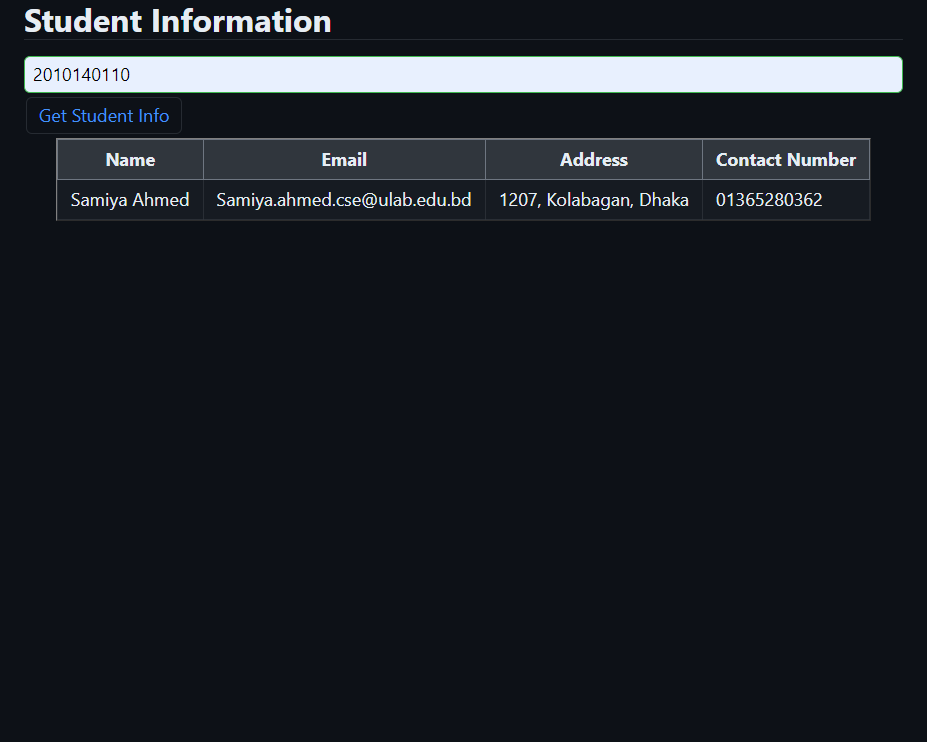
****

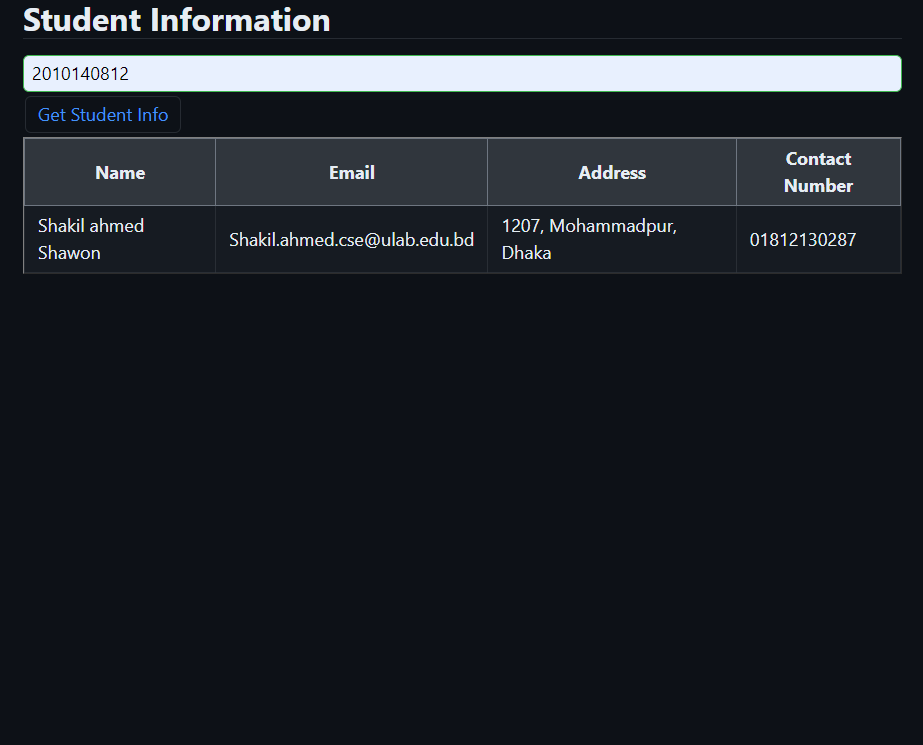
**Students.xml:**

****

**Outputs:**

****

****

****