NodeJS:

Node.js is a scalable network application builder that uses an asynchronous event-driven JavaScript engine. Many connections can be handled at the same time in the "hello world" example below. The callback is invoked with each connection, but if there is no work to be done, Node.js will sleep.

HTML:

HTML (Hypertext Markup Language) is the most widely used markup language for creating web pages and other data that can be seen in any web browser. Inside the web page content, HTML is written as HTML components with labels contained in edge sections (like html>). Most HTML labels are two by two, such as h1> and /h1>. When all other factors are equal, HTML components create the structure squares. HTML allows you to incorporate photos and items and can be used to create intuitive structures. It enables the creation of well-organized reports by interpreting auxiliary semantics for text, such as headers, paragraphs, lists, quotes, links, and other elements. It can include text written in dialects such as JavaScript and Cascading Style Sheets, which have an impact on the behavior of HTML pages. The first advantage of HTML is that it is widely used. The HTML language is supported by each program. It's easy to pick up and use. It is, of course, included in each window, so there is no need to purchase additional software.

RestAPI:

Restful web services are online resources that can be used to obtain specific data. These administrations and services essentially demonstrate how the REST API functions. REST, or Representational State Transfer, is a web service design style or set of guidelines.

REST can be used to modify or view assets on the server without having to perform any server-side operations. The customer requests an asset from the server, and the server responds (if there are no mistakes). The reaction is a representation of the resource that is now available on the server. It could be a JSON, XML, PDF, or DOC file, for example.

Use Case:

A use case depicts how a client uses a framework to accomplish a certain goal. A use case chart is made up of the system, associated use cases, and actors, all of which are linked together to illustrate what is being depicted. Who is using the framework (System)?

What would the actors hope to achieve if they were actors? (Use cases), as a result, assist ensure that the correct framework is produced by capturing the requirements from the client's perspective.

Entity Relationship Diagram (ERD):

The Entity Relationship Diagram (ERD) or Entity Relationship Diagram (ERD) is a type of structural diagram used in database schema. The Entity Relation model is based on the concept of major actual elements inside the system scope and their relationships. In my instance, I have two key entities: proxy sites (providers) and IP addresses (that I get from providers)

There are number of six tables in my database, For the admin, teacher and pupil login, User table store the username and password. For the authentication of API AUTH’s table hold the API key before accessing any API function. Configuration table keep the several value settings like the value for discarding the proxy’s after 7 days. Proxy providers table hold the list of all providers websites details where from I am fetching data and the primary key of proxy provider table is using As Foreign key another table proxy’s that storing the IP;s against each proxy site, That show on front end.