Internship Report – Day 16

I learned about RESTful APIs and how to run a server using the Express framework in VS Code. A RESTful API (Application Programming Interface) helps different applications talk to each other using the internet. It follows a set of rules that allow a client (like a web browser or app) to request data from a server and get a response. Express is a popular JavaScript framework that makes it easy to create and manage APIs.

To test my API, I used Insomnia, which is a tool that allows me to send different types of requests to the server and see the responses. This helped me understand how the server processes these requests. There are four main operations I learned:

GET: This is used to fetch data from the server. For example, if I request a list of users, the server sends the data back in response.

POST: This is used to send new data to the server. For example, when I create a new user, the API stores the data in the database.

PUT: This is used to update existing data on the server. If I want to change a user's details, I send a PUT request with the updated information.

DELETE: This is used to remove data from the server. If I want to delete a user, I send a DELETE request, and the server removes that data.

I also learned about request (req) and response (res) objects in Express. When a client sends a request to the API, Express handles it using the req object, which contains the request details. The res object is used to send a response back to the client. This made it clear how the communication between the client and server works.

I also used ToJS code, which helps in handling JavaScript objects and makes working with APIs easier. While running the server in VS Code, I saw how real-world applications handle data. By making different API requests and getting responses, I understood how websites and mobile apps communicate with servers.

One of the most interesting things I learned was how APIs make web development more efficient. Instead of storing everything on a user's device, apps can store and fetch data from a server, making them faster and more scalable. I also learned about error handling, which ensures that if something goes wrong, the API provides a proper response instead of crashing.

Overall, learning about RESTful APIs, Express framework, and Insomnia helped me understand backend development better. Now, I know how data flows between a client and a server, how to create and test APIs, and how different applications communicate. This knowledge will help me build more powerful and efficient web applications in the future.

I learned how to create and use classes in JavaScript. A class is like a blueprint that helps us make objects with the same structure but different values. I now understand that the class keyword is used to create a class, and the constructor method is a special function inside the class that sets up object properties. In the Dog class example, the constructor takes name and age as inputs and assigns them to the object using the this keyword. This means every dog object will have its own name and age.

I also learned about methods inside a class. A method is a function that belongs to a class and describes what an object can do. In the Dog class, the bark() method prints "Woof!" when called. To create objects from the class, I used the new keyword, like this: let dog1 = new Dog("Tommy", 3);. This created an object named dog1 with its own name and age. Another object, dog2, was created with let dog2 = new Dog("Rocky", 5);, showing that I can create many objects from the same class.

To call a function inside a class, I used dog1.bark();. This made dog1 use its bark() method and print "Woof!". The same happened when I called dog2.bark();. This showed me that all objects made from the class can use the same method. I also understood that while methods are shared among objects, their properties like name and age are unique to each object.

This lesson helped me understand how JavaScript classes and objects help organize code. Instead of writing separate code for each object, I can use a class to reuse the structure and save time. I also learned how constructors set up object properties and how methods define what an object can do. Calling functions inside a class showed me how objects can store data and perform actions.

Overall, I now understand how to create classes, use constructors, define methods, and create multiple objects from the same class. This makes programming in JavaScript more organized, efficient, and easy to manage.

I learned about MongoDB, which is a NoSQL database used to store data in a flexible and scalable way. Unlike traditional relational databases, MongoDB stores data in JSON-like documents instead of tables. This makes it easier to handle large amounts of unstructured or semi-structured data.

MongoDB allows operations like creating, reading, updating, and deleting (CRUD) data efficiently. It works well with JavaScript and is commonly used with Node.js and Express in web applications.

With MongoDB, data is stored in collections instead of tables, and each entry is called a document. This structure helps developers build fast and scalable applications easily