Week-5 Web application Assignment

1.Explain Model, View and Controller in brief?

MVC stands for model-view-controller. Here's what each of those components mean:

- **Model**: The backend that contains all the data logic
- **View**: The frontend or graphical user interface (GUI)
- **Controller**: The brains of the application that controls how data is displayed

The concept of MVCs was first introduced by Trygve Reenskaug, who proposed it as a way to develop desktop application GUIs.

Today the MVC pattern is used for modern web applications because it allows the application to be scalable, maintainable, and easy to expand.

Why Should You Use MVC?

Three words: **separation of concerns**, or SoC for short.
The MVC pattern helps you break up the frontend and backend code into separate components. This way, it's much easier to manage and make changes to either side without them interfering with each other.

But this is easier said than done, especially when several developers need to update, modify, or debug a full-blown application simultaneously.

How to Use MVC

To better illustrate the MVC pattern, I've included a web application that shows how these concepts all work.

My Car Clicker application is a variation of a well-known Cat Clicker app.

Here are some of the major differences in my app:

1. No cats, **only** muscle cars images (sorry cat lovers!)
2. Multiple car models are listed
3. There are multiple click counters
4. It only displays the selected car

2.What is the purpose of render method?

Purpose of render():

- React renders HTML to the web page by using a function called render().
- The purpose of the function is to display the specified HTML code inside the specified HTML element.
- In the render() method, we can read props and state and return our JSX code to the root component of our app.
- In the render() method, we cannot change the state, and we cannot cause side effects(such as making an HTTP request to the webserver).

3.How Do I Render Plain Html?

Express Js is the web application framework based on Node.js web server functionality that helps us to create the application endpoints that

respond based on the HTTP request (POST, GET, etc) method and the requested route.
The res.sendFile() method of the express.js module is used to render a particular HTML file that is present in the local machine.

Syntax:

res.sendFile(path,[options],[fn])

Parameters: The path parameter describes the path and the options parameter contains various properties like maxAge, root, etc and fn is the callback function.

Returns: It returns an Object.

Project Setup:

Step 1: Install Node.js if Node.js is not installed in your machine.

Step 2: Create a new folder named public, inside the public folders. Create two files named index.html and products.html inside the public folder.

Step 3: Now, initialize a new Node.js project with default configurations using the following command on the command line.

npm init -y

Step 5: Now install express inside your project using the following command on the command line.

npm install express

Project Structure: After following the steps your project structure will look like.

4. Develop Node.js is a platform for building fast and scalable server applications using JavaScript.

Node.js is a server-side platform built on Google Chrome's JavaScript Engine (V8 Engine). Node.js was developed by Ryan Dahl in 2009 and its latest version is v0.10.36. The definition of Node.js as supplied by its official documentation is as follows −

Node.js is a platform built on Chrome's JavaScript runtime for easily building fast and scalable network applications. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices.

Node.js is an open source, cross-platform runtime environment for developing server-side and networking applications. Node.js applications are written in JavaScript, and can be run within the Node.js runtime on OS X, Microsoft Windows, and Linux.

Node.js also provides a rich library of various JavaScript modules which simplifies the development of web applications using Node.js to a great extent.

Node.js = Runtime Environment + JavaScript Library

Features of Node.js

Following are some of the important features that make Node.js the first choice of software architects.

- **Asynchronous and Event Driven** − All APIs of Node.js library are asynchronous, that is, non-blocking. It essentially means a Node.js based server never waits for an API to return data. The server moves to the next API after calling it and a notification mechanism of Events of Node.js helps the server to get a response from the previous API call.
- **Very Fast** − Being built on Google Chrome's V8 JavaScript Engine, Node.js library is very fast in code execution.
- **Single Threaded but Highly Scalable** − Node.js uses a single threaded model with event looping. Event mechanism helps the server to respond in a non-blocking way and makes the server highly scalable as opposed to traditional servers which create limited threads to handle requests. Node.js uses a single threaded program and the same program can provide service to a much larger number of requests than traditional servers like Apache HTTP Server.
- **No Buffering** − Node.js applications never buffer any data. These applications simply output the data in chunks.

- **License** − Node.js is released under the MIT license.

Who Uses Node.js?

Following is the link on github wiki containing an exhaustive list of projects, application and companies which are using Node.js. This list includes eBay, General Electric, GoDaddy, Microsoft, PayPal, Uber, Wikipins, Yahoo!, and Yammer to name a few.

- Projects, Applications, and Companies Using Node

Concepts

The following diagram depicts some important parts of Node.js which we will discuss in detail in the subsequent chapters.